Міністерство освіти і науки України
Сумський державний університет

В. В. Москаленко, А. С. Довбиш

ВСТУП ДО ІНФОРМАЦІЙНОГО АНАЛІЗУ І СИНТЕЗУ ІНФОКОМУНІКАЦІЙНИХ СИСТЕМ

Навчальний посібник

Рекомендовано вченою радою Сумського державного університету

Суми
Сумський державний університет
2016

УДК 004.75(075.3)
ББК 73я73
М82

Рецензенти:
С. Ф. Теленик – доктор технічних наук, професор (Національний технічний університет України «Київський політехнічний інститут ім. Ігоря Сікорського»);
О. І. Цопа – доктор технічних наук, професор (Харківський національний університет радіоелектроніки)

Рекомендовано до видання
вченою радою Сумського державного університету
як навчальний посібник
(протокол № 5 від 10 листопада 2016 року)

Москаленко В. В.

М82 Вступ до інформаційного аналізу і синтезу інфокомунікаційних систем : навч. посіб. / В. В. Москаленко, А. С. Довбиш. – Суми : Сумський державний університет, 2016. – 226 с.

ISBN 978-966-657-639-5

Метою навчального посібника є ознайомлення студентів-програмістів з основами інформаційного аналізу і синтезу систем керування розподіленим обчислювальним середовищем. При цьому особливу увагу приділено необхідності розв'язання задачі енергозбереження під час забезпечення високої якості обслуговування користувачів інфокомунікаційних сервісів. Посібник дозволяє студентам, аспірантам і науковцям одержати необхідні знання для інформаційного синтезу систем керування та розроблення програмних додатків у галузі інфокомунікацій.

УДК 004.75(075.3)
ББК 73я73

© Москаленко В. В., Довбиш А. С., 2016
ISBN 978-966-657-639-5 © Сумський державний університет, 2016

ЗМІСТ

С.

Перелік умовних позначень 5
Вступ 6

Розділ 1. Керування якістю обслуговування в інфокомунікаційній системі 8

1.1. Оцінювання якості обслуговування в інфокомунікаційній системі 8
1.2. Принципи забезпечення якості обслуговування в інфокомунікаційній системі 23
1.3. Ідентифікація трафіку та анонімність в інфокомунікаційній системі 41
1.4. Виявлення атак та керування безпекою в інфокомунікаційній системі 56
1.5. Контрольні запитання та завдання для самопідготовки 68

Розділ 2. Керування ресурсами та планування завдань в інфокомунікаційній системі 71

2.1. Системи розподілених обчислень 71
2.2. Методи планування завдань та ресурсів 85
2.3. Енергозбереження в інфокомунікаційних системах ..99
2.4. Прогнозування функціонального стану інфокомунікаційної системи 105

2.5. Контрольні запитання та завдання
для самопідготовки 126

**Розділ 3. Оптимізація системи керування
розподіленим обчислювальним середовищем..... 130**

3.1. Інформаційні характеристики системи керування ... 130
3.2. Критерії оцінювання функціональної ефективності
та оптимізації системи керування, що навчається..... 143
3.3. Оптимізація керування інфокомунікаційною
системою за узагальненим критерієм ефективності..... 161
3.4. Методи автоматичної класифікації 169
3.5. Методи оптимізації параметрів функціонування
системи керування розподіленим обчислювальним
середовищем 197
3.6. Контрольні запитання та завдання
для самопідготовки 209

Список літератури 214

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ІЕІ-технологія – інформаційно-екстремальна інтелектуальна технологія.
КФЕ – критерій функціональної ефективності.
СППР – система підтримки прийняття рішень.
ЦОД – центри оброблення даних.
ELA – угода про рівень очікуваної якості послуг (Experience Level Agreements).
FSS – пошук косяком риб (Fish School Search, FSS).
IDS – системи виявлення атак (Intrusion Detection System).
KPI – ключовий показник продуктивності (key performance indicators).
KQI – ключовий показник якості (Key Quality Indicator).
MOS – середнє значення експертних оцінювань (Mean Opinion Score).
PSO – оптимізація роєм частинок (Particle Swarm Optimization).
SLA – угоди про рівень послуг (Service-level agreement).
SVM – метод опорних векторів (Support Vector Machine).
QoE – якість послуги, що сприймається користувачем (Quality of Experience).
QoS – якість обслуговування (Quality of Service).

ВСТУП

Глобалізація світової економіки та формування соціально орієнтованого середовища вимагають значних обчислювальних і телекомунікаційних ресурсів, систем накопичення і збереження великих обсягів даних. З цією метою мережі провайдерів, корпорацій, комунальні та державні ІТ-системи інтегрувалися в єдине розподілене обчислювальне середовище, ІТ-інфраструктура та сервіси якого утворили інфокомунікаційні системи різного призначення. При цьому особливої актуальності набуває задача інформаційного аналізу і синтезу систем керування плануванням завдань та розподілом ресурсів в інфокомунікаційному обчислювальному середовищі з метою забезпечення енергозбереження при високій якості обслуговування користувачів інфокомунікаційних сервісів.

Розуміючи складність проблеми інформаційного аналізу і синтезу інфокомунікаційних систем керування, автори розглядають цей посібник як введення в спеціалізацію «Інформаційно-комунікаційні технології», що викладається студентам спеціальності «Комп'ютерні науки та інформаційні технології» в Сумському державному університеті. Оскільки посібник написано для програмістів, то він містить необхідні відомості про стан і проблематику сучасних телекомунікацій, методи створення систем керування на основі машинного навчання та розпізнавання образів. При цьому увага студентів акцентується на перспективності застосування в задачах інформаційного аналізу і синтезу здатних навчатися (самонавчатися) інфокомунікаційних

систем керування так званої інформаційно-екстремальної інтелектуальної технології аналізу даних, в основу якої покладено максимізацію інформаційної спроможності системи в процесі її машинного навчання.

Навчальний посібник складається з трьох розділів.

У першому розділі розглядаються визначення, принципи і основні методи оцінювання якості обслуговування в інфокомунікаційних сервісах, ідентифікації та маскування трафіку та захисту інформації.

У другому розділі викладені варіанти організації розподілених обчислень, методи і способи енергозбереження в сучасних розподілених середовищах і шляхи зменшення ймовірності перевантажень і відмов при оптимальних енерговитратах.

Третій розділ присвячено оптимізації, здатної навчатися інфокомунікаційної системи керування як за інформаційними, так і узагальненими функціонально-вартісними критеріями. При цьому значну увагу приділено конструюванню критеріїв оптимізації та їх обчислювальному аспекту.

Загальне редагування і підрозділи 3.1 і 3.2 виконано А. С. Довбишем, а В. В. Москаленком написано перший, другий розділи і підрозділи 3.3–3.5.

РОЗДІЛ 1
КЕРУВАННЯ ЯКІСТЮ ОБСЛУГОВУВАННЯ В ІНФОКОМУНІКАЦІЙНІЙ СИСТЕМІ

1.1. Оцінювання якості обслуговування в інфокомунікаційній системі

Розвиток інформаційних та телекомунікаційних технологій призвів до формування уніфікованої інфокомунікаційної архітектури, в якій інформаційні технології не можуть працювати без телекомунікаційних і навпаки. Наприклад, сервери датацентрів використовуються за допомогою каналів зв'язку, а телекомунікаційна мережа керується, адмініструється і надає послуги за допомогою засобів інформаційних технологій. Ядром інформаційно-телекомунікаційного середовища є опорна IP-мережа, що підтримує пакетне передавання даних і забезпечує повну або часткову інтеграцію (конвергенцію) послуг передавання голосу, даних і мультимедіа. При цьому одним із основних аспектів, який повинен братися до уваги при проектуванні таких систем, є забезпечення якості обслуговування.

Визначення 1.1.1. Якість обслуговування (Quality of Service, QoS) характеризує загальний ефект, вироблений послугою, який визначає ступінь задоволеності користувача послугою і може бути виражена набором технічних характеристик інформаційно-телекомунікаційного середовища [1].

У галузі телекомунікацій під якістю обслуговування розуміють здатність мережі забезпечити необхідний сервіс заданому трафіку в певних технологічних рамках.

У рекомендації Y.1540 [2] розглядаються такі характеристики пакетних мереж IP, як найбільш важливі за ступенем їх впливу на наскрізну QoS (від джерела до одержувача):

- смуга пропускання (Bandwidth);
- затримка доставлення пакета (IPTD);
- варіація затримки пакетів (IPDV);
- коефіцієнт втрат пакетів (IPLR);
- коефіцієнт помилок пакетів (IPER).

Визначення 1.1.2. Смуга пропускання описує номінальну пропускну здатність середовища передавання інформації, протоколу або з'єднання і вимірюється кількістю бітів за секунду.

Визначення 1.1.3. Затримка доставлення пакета (IP packet transfer delay, IPTD) визначається як час між двома подіями – введенням пакета у вхідну точку мережі та виведенням пакета з вихідної точки мережі.

Загалом параметр IPTD визначається як час доставляння пакета між джерелом і одержувачем для всіх пакетів – як успішно переданих, так і пошкоджених помилками. Значення середньої затримки залежить від переданого в мережі трафіку і доступних мережевих ресурсів, зокрема від пропускної здатності. Зростання навантаження і зменшення доступних мережевих ресурсів ведуть до зростання черг у вузлах мережі і, як наслідок, до збільшення середніх затримок доставляння пакетів.

Визначення 1.1.4. Варіація затримки пакета (IP packet delay variation, IPDV) описує різницю між абсолютною затримкою пакета та нормованою, що визначається як абсолютне значення затримки доставляння першого пакета IP між вхідною і вихідною точками мережі.

Варіація затримки пакета IP, або джитер, виявляється у тому, що послідовні пакети надходять до одержувача в нерегулярні моменти часу. У системах IP-телефонії це, наприклад, призводить до спотворень звуку і, як наслідок, до нерозбірливості мови.

Визначення 1.1.5. Коефіцієнт втрати пакетів (IP packet loss ratio, IPLR) визначається як відношення сумарної кількості втрачених пакетів до загальної кількості прийнятих в обраному наборі переданих та прийнятих пакетів.

Втрати пакетів у мережах IP виникають у тому разі, коли значення затримок під час передавання перевищує нормоване значення ($T_{\max}$). Якщо пакети губляться, то при передаванні даних можливе їх повторне передавання за запитом сторони, яка приймає. У системах голосового зв'язку (VoIP) пакети, які надійшли до одержувача із затримкою, що перевищує $T_{\max}$, відкидаються, це призводить до провалів у прийнятій мові. Серед причин, що викликають втрати пакетів, необхідно відзначити зростання черг у вузлах мережі, які виникають під час перевантажень.

Визначення 1.1.6. Коефіцієнт помилок пакетів (IP packet error ratio, IPER) визначається як сумарна кількість пакетів, прийнятих із помилками, до суми успішно прийнятих і прийнятих із помилками.

Рекомендація Y.1541 встановлює відповідність між класами якості обслуговування і додатками:

- Клас 0 – додатки реального часу, чутливі до джитера, характеризуються високим рівнем інтерактивності (VoIP, відеоконференції).
- Клас 1 – додатки реального часу, чутливі до джитера, інтерактивні (VoIP, відеоконференції).
- Клас 2 – транзакції даних, що характеризуються високим рівнем інтерактивності (наприклад, сигналізація).
- Клас 3 – транзакції даних, інтерактивні.
- Клас 4 – додатки, що допускають низький рівень втрат (короткі транзакції, масиви даних, потокове відео).
- Клас 5 – традиційні застосування IP-мереж.

Рекомендація Y.1540 визначає числові значення мережевих характеристик, які повинні забезпечуватися в IP-мережах на міжнародних трактах, що з'єднують термінали користувачів. У таблиці 1.1 [2] наведені норми на вищерозглянуті мережеві характеристики, де позначка Н означає ненормовані значення.

Таблиця 1.1 – Норми для характеристик IP-мереж

Параметр	Клас QoS					
	0	1	2	3	4	5
IPTD	100 мс	400 мс	100 мс	400 мс	1с	Н
IPDV	50 мс	50 мс	Н	Н	Н	Н
IPLR	$1x10^{-3}$	$1x10^{-3}$	$1x10^{-3}$	$1x10^{-3}$	$1x10^{-3}$	Н
IPER	$1x10^{-4}$	$1x10^{-4}$	$1x10^{-4}$	$1x10^{-4}$	$1x10^{-4}$	Н

У таблиці 1.1 норми на параметри поділені на різні класи QoS і визначаються залежно від додатків і мережевих механізмів, що застосовуються для забезпечення гарантованої якості обслуговування.

Істотним системним недоліком під час оцінювання якості обслуговування лише на основі аналізу мережевих параметрів є виключення з розгляду можливостей програмного забезпечення прикладного рівня. Сучасні методи кодування і декодування інформації дозволяють, наприклад, змінювати вимоги до пропускної спроможності за рахунок кодування зі змінною швидкістю, зменшувати вимоги диференційованого захисту пакетів із мультимедійною інформацією, компенсувати вплив втрати пакетів і джитера за рахунок адаптивного відтворення [3].

Користувач сприймає якість обслуговування як сукупне оцінювання своїх контактів з оператором – від замовлення послуги до відмови від неї. Для врахування впливу всіх аспектів і учасників надання послуги (користувач, термінал, мережа тощо) згідно з рекоментаціями P.10 / G.100 [4] інтегральним показником є якість послуги, що сприймається користувачем.

Визначення 1.1.7. Якість послуги, що сприймається, (Quality of Experience, QoE) – загальна прийнятність програми або послуги, що суб'єктивно сприймається кінцевим користувачем [4, 5].

QoE враховує як вплив параметрів мережі та прикладного програмного забезпечення, так і очікування користувача. Залежність рівня QoE від покращання параметрів QoS часто має вигляд логістичної кривої (рис. 1.1), однак у

загальному випадку залежність може мати більш складний або неоднозначний характер, оскільки на суб'єктивне оцінювання якості інформаційно-телекомунікаційних послуг впливає безліч факторів.

Рисунок 1.1 – Залежність QoE від параметрів QoS

Методи оцінювання якості послуг, що сприймаються користувачем, згідно з рекомендаціями ITU-T G.1080 [6] класифікують на об'єктивні та суб'єктивні.

Суб'єктивні методи дозволяють одержати найбільш адекватне оцінювання сприйняття якості, оскільки прямо відображають погляд користувачів. Численні вітчизняні та міжнародні стандарти визначають особливості організації та проведення суб'єктивного оцінювання як окремих показників (розбірливість, впізнавання), так й інтегральної якості послуги. Результатом застосування суб'єктивних методів є усереднений погляд групи осіб – експертів на якість наданої інформаційної послуги. У рекомендації ITU-T

P.830 [6] для оцінювання сприйняття якості обслуговування встановлюється п’ятибальна шкала (табл. 1.2) MOS (Mean Opinion Score – середнє значення експертних оцінювань).

Таблиця 1.2 – Шкала MOS

Оцінка MOS	Категорія якості	Задоволеність користувача
5	Найкраща (Excellent)	Задоволення найвищою мірою
4	Висока (Good)	Задоволені
3	Середня (Fair)	Деякі не задоволені
2	Низька (Poor)	Багато не задоволених
1	Погана (Bad)	Майже всі не задоволені

Галузь застосування суб’єктивних тестів обмежується тривалістю процедури тестування (особливо, якщо досліджується якість, залежна від великої кількості показників), а також неможливістю автоматизації та проведення у реальному темпі часу.

Об’єктивні методи оцінювання якості дозволяють виключити людину із процедури оцінювання. Отже, легко автоматизуються. Об’єктивні методи поділяють на активні (інтрузивні) та пасивні (неінтрузивні). В активних методах оцінювання якості здійснюється шляхом порівняння еталонної послідовності (оригіналу) із послідовністю, що була викривлена під час передавання по мережі.

До активних методів належать :

- PESQ (Perceptual Evaluation of Speech Quality) – оцінювання сприйняття якості передавання мовлення (рекомендація ITU-T P.862).
- PEAQ (Perceptual Evaluation of Audio Quality) – оцінювання сприйняття якості передавання аудіо (рекомендація ITU-T BS.1387).
- PEVQ (Perceptual Evaluation of Video Quality) – оцінювання сприйняття якості передавання відео (рекомендація ITU-T J.247).

До пасивних методів належать :

- PSQM (Perceptual Speech Quality Measurement) алгоритм пасивного моніторингу для оцінювання якості мовного зв'язку (рекомендація ITU-T P.563);
- E-model оцінює якість мовного зв'язку за допомогою R-фактора (рекомендація ITU-T G.107);
- оцінювання якості передавання даних (рекомендації ITU-T G.1030 G.1040).

Особливе положення у класифікації методів оцінювання якості сприйняття займає метод PSQA (Pseudo-Subjective Quality Assessment) [6], що дозволяє здійснювати об’єктивне оцінювання із використанням нейронної мережі, навчання якої проводиться із використанням суб’єктивних оцінювань. Проте в умовах незбалансованості та перетину класів розпізнавання, що характерно для задач оцінювання QoE, біонічні підходи до інтелектуального аналізу даних характеризуються невисокою достовірністю і вимагають значних обчислювальних ресурсів [7].

Подальший розвиток методу PSQA здійснюється в напрямі удосконалення алгоритмів машинного навчання та способів формування статистичних вибірок, що характеризують параметри мережі при відомих оцінюваннях сприйняття якості сервісу. Алгоритми машинного навчання формують вирішальні правила, які дозволяють у режимі реального часу прогнозувати рівень QoE. При цьому для формування навчальної матриці здійснюється періодичне опитування користувачів.

У 2014 році Європейский інститут телекомунікаційних стандартів (ETSI) запропонував специфікацію TS 103 294 V1.1.1, де розглядається методологія оцінювання QoE у рамках моделі ARCU (Application-Resource-Context-User) [8]. Весь простір факторів, що впливають на сприйняття якості, було поділено на чотири підпростори : простір ресурсів, простір додатка, простір контексту та простір користувача (рис. 1.2).

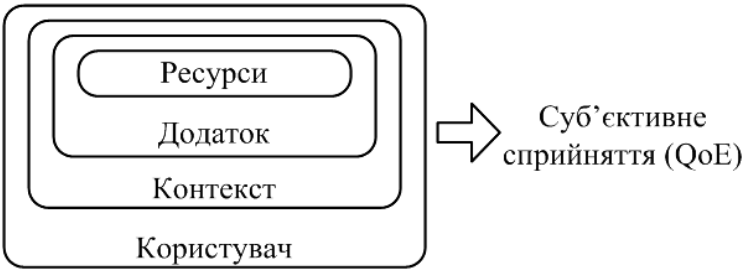


Рисунок 1.2 – ARCU-модель формування сприйняття якості інформаційно-телекомунікаційної послуги

Простір ресурсів подає характеристики та продуктивність технічних систем і мережевих ресурсів, використовуваних для надання інформаційно-телекомунікаційних послуг, наприклад мережеві параметри QoS, системні ресурси серверів та кінцевого пристрою користувача. Простір додатка складають конфігураційні параметри додатка/сервісу, наприклад, медіа-кодек, роздільна здатність відеопотоку, частота кадрів, розмір буфера, різні контентозалежні фактори (2D/3D, глибина кольору, інтерфейс). Простір контексту складають характеристики умов, за яких використовують сервіс або додаток, наприклад зовнішні умови (освітленість, шум), місце розташування користувача, час доби. Крім того, до контексту відносять характеристики мети використання сервісу/додатка та економічні фактори (ціна послуг, умови надання послуг). Простір користувача описує специфіку користувача цього додатка/сервісу, наприклад, демографічні дані, вподобання користувача, вимоги, очікування, апріорні знання, настрій, мотивацій та інші.

У працях [9, 10] фактори впливу на QoE було запропновано поділяти на технологічні, соціальні та економічні. При цьому технологічні фактори пов’язують з якістю оброблення зростаючих обсягів трафіку, а соціальні – з видом контенту (аудіо, відео, інше), галуззю застосування (освіта, мистецтво, медицина, інше), способом надання (streaming, broadcast, файл, інше), напрямком потоків (односпрямовані, двоспрямовані, мультиспрямовані), місцем перебування (в транспорті, вдома, на вулиці, інше) та пристроєм доступу (смартфон, ноутбук, планшет, інше). Еко-

номічні фактори пов'язують з бажанням постачальників збільшити свій прибуток, зберегти існуючих та залучити нових клієнтів шляхом розвитку позитивного чи негативного досвіду користувачів щодо тих чи інших послуг через рекламу, програми лояльності та інші методи впливу.

Моделювання впливу контекстних параметрів на задоволеність користувача та їх відображення на єдину шкалу QoE є складним процесом, що зумовлено наявністю великої кількості об'єктивних та суб'єктивних факторів. Деякі з цих факторів можуть бути виміряні, а інші можуть бути прихованими і впливати непрямим чином. При цьому ці фактори часто розглядають або окремо, або їх відображення здійснюється на різні шкали. Крім того, еволюційний розвиток суспільно-економічних процесів та інформаційних технологій обумовлюють обмеження на період часу актуальності моделей QoE. Питання врахування нестаціонарності потреб та очікувань клієнтів інформаційно-телекомунікаційних сервісів під час оцінювання QoE залишається відкритим.

Різниця поглядів користувача послуг та оператора зв'язку на якість сервісу стала причиною появи угоди про рівень послуг (Service-level agreement, SLA).

Визначення 1.1.8. Угода SLA – це угода між двома сторонами, яка покликана дати єдине розуміння постачальникам послуг та його клієнтам у частині обслуговування, пріоритетів, надійності тощо [11].

SLA може змінюватися від одного провайдера до іншого і, як правило, стосується доступності мережі/послуг та надійності передавання даних. Порушення SLA провайде-

ром послуг може компенсуватися користувачеві під час тарифікації у наступному періоді користування послугою.

Визначення 1.1.9. Доступність мережі – діапазон часу досяжності між вхідною і вихідною точками мережі. Доступність мережі визначається такими показниками :

- кількістю пошкоджень на одну абонентську лінію на рік;
- відсотком пошкоджень, усунених у контрольні терміни;
- час відновлення можливості доступу до мережі та час локалізації погіршення якості.

Визначення 1.1.10. Доступність сервісу – це діапазон часу, впродовж якого цей сервіс доступний між певними вхідною та вихідною точками з параметрами, оговореними в SLA.

Типова SLA про рівень сервісу містить такі елементи :

- детальний опис послуг, що надаються;
- опис рівнів забезпечення конфіденційності;
- детальний опис доступності сервісу (плановий час простою і період доступності);
- опис можливостей масштабування;
- перелік можливостей щодо додавання нових додатків, користувачів, послуг;
- перелік параметрів якості, методів і засобів їх контролю;
- звітність провайдера перед користувачем, періодичність і вид документів, що надаються;
- перелік доступних рівнів сервісу;

- фінансові умови надання послуг.

Розрізняють три типи SLA: внутрішній, партнерський та клієнтський.

Визначення 1.1.11. Внутрішній SLA – угода між функціональними підрозділами всередині оператора (для вибудовування наскрізного процесу «Забезпечення»).

Визначення 1.1.12. Партнерський SLA – частина договору між партнером-постачальником сервісів і оператором для забезпечення наскрізного QoS.

Визначення 1.1.13. Клієнтський SLA – формальний договір між клієнтом та оператором, спрямований на підвищення лояльності клієнта.

Головними елементами клієнтського SLA є розроблені TM Forum ключові показники якості KQI (Key Quality Indicator), які відображають погляд клієнта на якість послуги. В ієрархічній архітектурі інформаційно-телекомунікаційних систем показники KQI є числовими метриками вищого рівня і обчислюються на основі ключових показників продуктивності KPI (key performance indicators), що відображають показники (метрики) функціонування технологій і обладнання на нижчих рівнях системи. У класичному розумінні KPI є фізичними параметрами телекомунікаційних протоколів.

Найважливішою метрикою найвищого, технологічно-нейтрального, рівня є задоволення користувача, яке є оцінюванням налаштованості користувача на продовження обслуговування. Тому активно проводять дослідження і розроблення методів прогнозування показника QoE та механізмів його внесення до договору про рівень обслугову-

вання. Побудова моделей прогнозування QoE з прийнятною точністю дозволить реалізувати цінову диференціацію надання послуг за рівнем очікуваної якості та здійснити перехід від QoS-орієнтованих угод SLA до угод про рівень очікуваної якості послуг.

Визначення 1.1.14. *У*года про рівень очікуваної якості послуг (Experience Level Agreements, ELA) – це угода між двома сторонами, виражена в термінах і показниках QoE, і покликана дати узгоджене розуміння якості послуг їх постачальникам і клієнтам [12].

Договори ELA і SLA можуть співіснувати в інформаційно-телекомунікаційному середовищі і регламентувати умови надання послуг між інтернет-сервіс провайдером, провайдером контенту та користувачами (рис. 1.3).

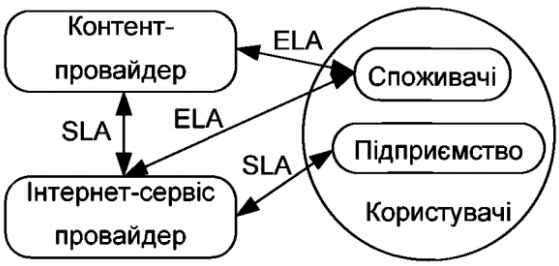


Рисунок 1.3 – Використання договорів SLA та ELA
в інформаційно-телекомунікаційному середовищі

У загальному випадку під час укладання договору ELA окремо для кожного типу інформаційно-телекомунікаційних послуг їх рівень очікуваної якості може бути узго-

дженний на симуляторі шляхом фіксації показників QoE (олімпійська модель, рейтинг зірочок, MOS-оцінювання) [12]. При цьому оперативний контроль і публікація відповідних значень QoS та їх відображення на шкалу QoE доцільно здійснювати третьою стороною для усунення конфлікту інтересів та забезпечення об'єктивності рішень щодо компенсації за порушення умов договору.

Збільшення номенклатури технологій, обладнання, ресурсів і компонентів інфраструктури сучасного інформаційно-телекомунікаційного середовища та зростання кількості користувачів призводять до значного зростання набору контрольованих метрик KPI та KQI. Виникає проблема впорядкування метрик, їх систематизації, зведення (агрегація) часткових метрик в інтегральні. Багаторівневість представлення інформаційних мереж визначає наявність ієрархічних зв'язків між метриками різних рівнів. Метрики вищого рівня обчислюють на основі метрик нижчого рівня, їх менше і вони дають узагальнені значення, що приховують вплив окремих метрик нижнього рівня [13]. Тому чим вищий рівень аналізу мереж операторів, тим більше чинників невизначеності впливає на оцінювання їх показників.

Таким чином, у зв'язку з переміщенням акценту з технічних питань на питання бізнесу і необхідністю врахування домінуючого характеру якісних, невизначених та нечітких факторів при формуванні висновку про рівень якості мережевих сервісів застосування методів інтелектуального аналізу даних є виправданим та доцільним.

1.2. Принципи забезпечення якості обслуговування в інфокомунікаційній системі

Поява принципово нових додатків, які дозволяють дистанційно керувати побутовою та іншою технікою («Інтернет речей»), створення «розумних» мереж (smart grids), широке використанням «хмарних» обчислень – нового втілення архітектури клієнт-сервер, що забезпечує єдине добре масштабоване середовище виконання різноманітного програмного забезпечення, – все це зумовило формування концепції мереж майбутнього [10, 14].

Визначення 1.2.1. Мережа майбутнього (Future Network, FN) – це глобальна інформаційна інфраструктура, яка об'єднує в собі вже існуючі інформаційно-телекомунікаційні мережі з урахуванням компонентів, що тільки плануються до впровадження, з єдиним центром керування інфраструктурою і здатністю надавати повний спектр інформаційно-телекоммунікаційних послуг (у будь-якому географічному місці, з гарантованою якістю, за прийнятною вартістю в будь-який час) на базі інноваційних технологій.

При проектуванні як сучасних, так і майбутніх мереж потрібно враховувати стрімке зростання популярності вебсервісів, технологій, побудованих на концепції «хмарних» обчислень, та активне впровадження концепції «все як послуга» (Everything as a service, EaaS), що обумовлює підвищення уваги до проблем забезпечення необхідного рівня якості обслуговування [13,14]. При цьому якість комплексних інформаційно-телекомунікаційних послуг може бути

забезпечена лише за умов укладання і строгого додержання SLA всіма провайдерами/операторами, що беруть участь у наданні цих послуг.

У сучасних та майбутніх мережах як транспортні можуть бути використані технології MPLS (Multiprotocol Label Switching), Ethernet та інші. IP-мережі, засновані на Ethernet-комутаторах і маршрутизаторах, є простими в проектуванні та експлуатації, легко нарощуються і модернізуються, але їх істотним недоліком є недостатня адаптованість до пропускання різнорідного трафіку [15].

Усю різноманітність відомих методів забезпечення QoS можна поділити на дві основні групи: методи збільшення кількості ресурсів мережі; методи раціонального розподілу обмежених мережевих ресурсів (рис. 1.4).

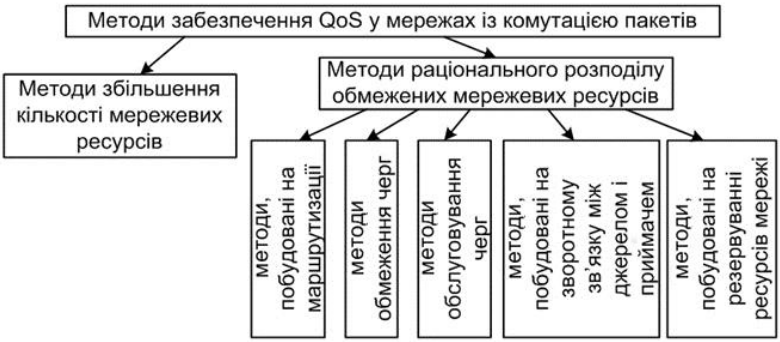


Рисунок 1.4 – Класифікація методів забезпечення QoS

Методи збільшення кількості ресурсів мережі передбачають використання засобів, що збільшують пропускну спроможність фізичних каналів, наприклад, за рахунок за-

стосування високошвидкісного середовища передавання або поліпшення характеристик сигналів. Питанням проектування інформаційно-телекомунікаційних мереж присвячено багато робіт. Наприклад, у працях [14–16] обґрунтовується вибір обсягу мережевого ресурсу та його використання в процесі функціонування мережі. Однак здійснивши обґрунтування вибору обсягу мережевих ресурсів, не завжди можна досягнути максимізації показника ефективності їх використання. Як свідчить досвід, за сучасних умов на практиці дуже складно забезпечити необхідний QoS у мережах тільки за рахунок збільшення їх фізичних ресурсів.

Методи раціонального розподілу обмежених мережевих ресурсів у різних літературних джерелах трактуються як методи управління ресурсами або методи боротьби з перевантаженнями. До основних засобів керування ресурсами традиційно відносять механізми керування чергами (буферний ресурс), засоби розподілу пропускної здатності трактів передавання (канальний ресурс), механізми керування трафіком (інформаційний ресурс).

Забезпечення наскрізної QoS «із кінця в кінець» (end-to-end) у рамках сучасних мультисервісних інформаційно-телекомунікаційних мереж передбачає використання цілого комплексу засобів керування трафіком на вузлах: класифікація, вимірювання, маркування, вирівнювання та профілювання потоку пакетів (рис. 1.5).

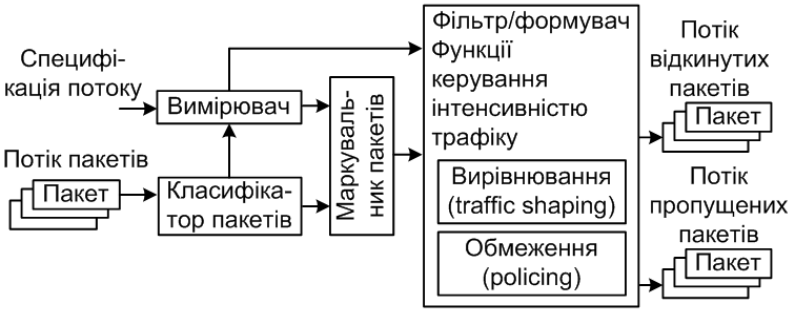


Рисунок 1.5 – Керування вхідним трафіком на мережевому вузлі з комутацією пакетів

Класифікатор пакетів є засобом, який відповідно до заданої політики QoS дозволяє розмістити пакет в одну з черг пріоритетів обслуговування. Класифікатор спочатку аналізує вміст одного або декількох полів заголовка пакета – пріоритет 802.1p, IP-пріоритет або поле DSCP у байті ToS. Якщо на вхідний порт вузла мережі надходить немаркований кадр (заголовок кадру не містить бітів пріоритету), то його класифікація може здійснюватися на основі MAC-адреси, IP-адреси, номера порту TCP / UDP або тегу VLAN. У разі шифрування трафіку або використання додатків, що динамічно змінюють порти, можуть знадобитися більш складні засоби аналізу та класифікації пакетів трафіку.

Мережевий вузол після процесу класифікації може здійснювати маркування пакетів. Маркування пакетів визначає спосіб запису/перезапису значень бітів пріоритету (DSCP, 802.1p або IP Precedence) вхідних пакетів даних. Як

правило, процес маркування здійснюється на граничних пристроях і дозволяє наступним вузлам мережі використовувати нове значення пріоритету пакета для віднесення його до одного з класів обслуговування, які підтримуються в мережі.

Профілювання трафіку на основі правил політики QoS передбачає у разі порушення параметрів профілю (наприклад, перевищення тривалості пульсації чи середньої швидкості) відкидання пакета чи його маркування зі зниженням пріоритету. Відкидання окремих пакетів знижує інтенсивність потоку і приводить його параметри у відповідність до вказаних у профілі. Маркування пакетів без відкидання необхідне для того, щоб пакети обслуговувалися, але вже зі зниженою якістю.

Функція вирівнювання трафіку призначена для набирання трафіком, що пройшов профілювання, необхідної «форми» в часі та реалізується шляхом буферизації пакетів. В основному за допомогою цієї функції намагаються згладити пульсації трафіку, і тим самим скоротити черги на вузлах мережі. Вирівнювання доцільно використовувати для відновлення часових співвідношень трафіку додатків, що працюють з рівномірними потоками, наприклад, додатків передавання голосу.

У разі підвищеного навантаження або тимчасового коливання трафіку, яке викликає перевантаження каналів зв'язку, повне або часткове вирішення проблем продуктивності мережі може здійснюватися методами оптимізації динамічної маршрутизації. Задача оптимізації маршрутизації трафіку полягає в знаходженні такого рішення, яке на

заданих структурі мережі і матриці попиту трафіку призведе до оптимального QoS у мережі. Як міра QoS може розглядатися утилізація каналів, що пояснюється її впливом на затримку та втрату пакетів між маршрутизаторами [17]. При цьому суть оптимізації полягає в «підстроюванні» маршрутизації до поточного навантаження з метою кращої утилізації мережевих ресурсів, що, у свою чергу, підвищує якість мережевих послуг.

Під час маршрутизації, побудованій на пункті призначення пакетів, маршрутизатор визначає вихідний інтерфейс для подальшого пересилання пакетів, виходячи зі значень метрик, які кількісно описують дистанцію до пункту призначення. Як правило, окрема адитивна метрика присвоюється кожному каналу, потім алгоритм визначення найкоротшого шляху використовується для визначення оптимальних маршрутів між усіма вузлами мережі (одноме-трична маршрутизація). Тобто, задаючи відповідні значення метрик каналів, можна побічно впливати на схему маршрутизації та таким чином оптимізувати її.

Поряд із одноме-тричними протоколами існують також схеми маршрутизації, що дозволяють враховувати при визначенні маршруту більше ніж одну метрику каналу (багатометрична маршрутизація). Прикладом є протокол маршрутизації EIGRP, розроблений компанією Cisco, який враховує чотири метрики. Однак тільки дві з них використовують за замовчуванням – адитивна метрика «затримка» і метрика «ємність».

Оптимізація маршрутизації здійснюється в процесі пошуку мінімуму максимального значення утилізації за різ-

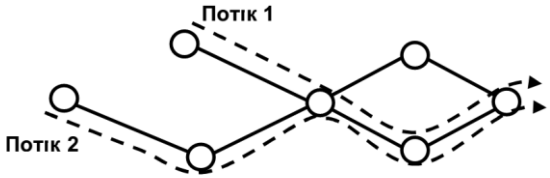
них значень метрики кожного з каналів. Критерій оптимізації має такий вигляд [17] :

$$\text{Критерій} = \left(\frac{1}{\max\limits_{i \in \text{канали}} (\text{утилізація}_i)} \right)^p, p > 0.$$

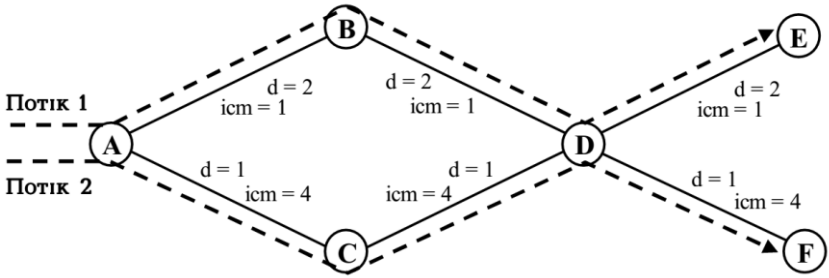
Вектор параметрів в оптимізаційній задачі складається з вагових коефіцієнтів метрик, що є компонентами метрики маршруту кожного з каналів. Процедура пошуку багаторазово застосовує можливі модифікації метрик для відведення трафіку із каналу з найвищою утилізацією. Процедура відведення трафіку повторюється поки вона покращує поточний результат.

Для пошуку оптимальних рішень можна використати будь-який із алгоритмів багатопараметричної оптимізації, який має достатню оперативність, наприклад, такі популяційні алгоритми пошуку, як генетичний або рою частинок. У разі використання однометричної маршрутизації вектор параметрів, що оптимізуються, буде містити вагові коефіцієнти всіх каналів у порядку їх нумерації. У разі двометричної маршрутизації вирішення оптимізаційної задачі матиме вдвічі більшу кількість параметрів.

На рис. 1.6 *а* показано, що при перетині двох потоків з однаковим пунктом призначення вони об’єднуються і далі надсилаються по одному й тому ж інтерфейсу, що може викликати перевантаження одних каналів і недовантаження інших.



а



б

Рисунок 1.6 – Сценарії маршрутизації при перетині двох потоків із однаковими пунктами призначення : *а* – без оптимізації; *б* – з оптимізацією

Аналіз рис. 1.6 *б* показує, що, маніпулюючи в процесі оптимізації адитивними метриками «затримка» та «ємність», можна відвести трафік від найбільш завантаженого каналу і здійснити довантаження альтернативних шляхів передавання.

Технологія маршрутизації MPLS (Multiprotocol Label Switching) дає можливість встановлювати структуру маршрутизації всередині IP-мережі незалежно від використовуваного протоколу маршрутизації і задавати певні маршрути для окремих потоків трафіку. Кожен IP-пакет, призначений для MPLS-маршрутизації, містить спеціальну поз-

начку, якою керуються маршрутизатори під час подальшого пересилання пакета на всьому шляху його проходження. Такий підхід дає високий рівень гнучкості і дозволяє досягти будь-якої бажаної картини маршрутизації. Однак дороговизна обладнання, проектування та обслуговування стримують широке впровадження MPLS. Крім того, для технології характерна складність в підтримці інформаційної безпеки – якщо не працює один протокол, то вся мережа не функціонує.

Основою ядра сучасних пакетних транспортних мереж є технологія IP/MPLS. За її допомогою можливе забезпечення параметрів QoS. Установлення логічних з'єднань виконується на базі роботи протоколів маршрутизації IP-мережі – тобто спочатку за допомогою протоколів динамічної маршрутизації у вузлах мережі будуються таблиці маршрутизації, за допомогою іншого протоколу на базі даних таблиць маршрутизації автоматично прокладаються канали передавання інформації LSP (Label Switching Path) і будується MPLS-мережа. Тому надійність, якість та керованість MPLS-мережі значною мірою визначаються якістю роботи алгоритмів динамічної маршрутизації.

Більшість відомих систем керування телекомунікаційними мережами використовують одну й ту ж саму базову структуру і способи взаємодії, побудовані на моделі взаємодії системи керування мережею (Network Management System, NMS) з об'єктами керування за допомогою мережевих агентів (рис. 1.7) [18].

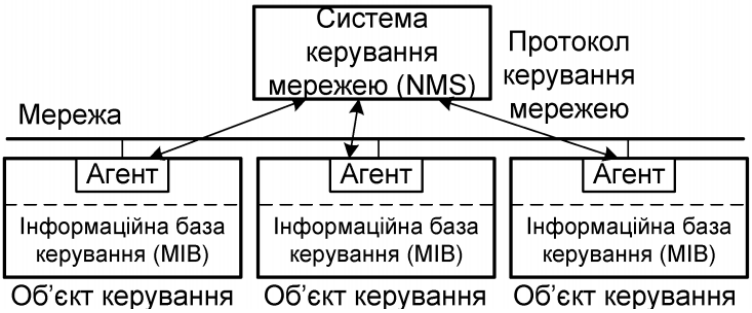


Рисунок 1.7 – Типова архітектура системи керування мережею

Визначення 1.2.2. Агент – це сутність, що сприймає своє середовище за допомогою сенсорів та діє на це середовище за допомогою виконавчого механізму.

Визначення 1.2.3. Мережевий агент – це програма, що є посередником між керованим ресурсом та головною керуючою програмою-менеджером, здатна автономно взаємодіяти з ресурсом, обробляти і подавати у формалізованому вигляді інформацію про його стан та надавати цю інформацію менеджеру.

Агенти безпосередньо взаємодіють з керованими об'єктами і працюють з інформаційною базою керування (Management Information Base, MIB), яка містить списки керованих параметрів та їх стан. Менеджер у будь-який момент може здійснити запит даних про стан об'єкта. Опитування стану реалізуються за ініціативи менеджера регулярно чи епізодично. Агенти можуть реалізовувати і асинхронне інформування менеджера, наприклад повідомленням тривоги.

У комп'ютерних мережах системи мережевого керування використовують один із стандартних протоколів – SNMP (Simple Network Management Protocol) або CMIP (Common Management Information Protocol). CMIP використовується в телекомунікаційних мережах, де необхідні всі доступні можливості керування мережами, у той час як SNMP використовується в локальних і корпоративних мережах, де достатньо мінімуму даних.

Для керування складною телекомунікаційною мережею доцільно використовувати розподілену систему керування, що значною мірою відображає структуру мережі (рис. 1.8) [18].

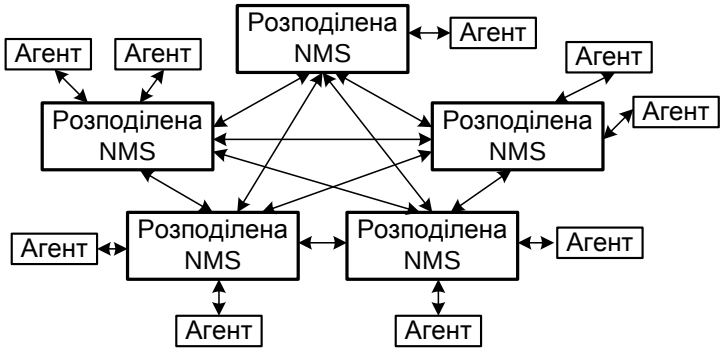


Рисунок 1.8 – Розподілена система керування телекомунікаційною мережею

Для динамічно реконфігурованих телекомунікаційних мереж необхідна більш гнучка, адаптивна система керування. Вирішення даного завдання знаходиться у площині побудови інтелектуальних систем керування, де за-

безпечення заданого рівня якості обслуговування реалізують за допомогою так званих інтелектуальних агентів, тобто агентів, здатних адаптуватися до змін стану мережі.

Визначення 1.2.4. Інтелектуальний агент – агент, дії якого раціональні, тобто спрямовані на досягнення певної мети.

Інтелектуальні агенти можуть мати такі властивості: навчатися і розвиватися в процесі взаємодії з довкіллям, пристосовуватися в режимі реального часу, застосовувати нові методи вирішення проблем, володіти базою прикладів із можливістю її поповнення, мати параметри для моделювання продуктивності, пам'яті, часу, аналізувати себе та результат.

Для керування мережевими компонентами з метою забезпечення надійності та QoS необхідно розв'язати велику кількість задач, серед яких розгортання, конфігурування, моніторинг, налагодження, оброблення помилок, ремонт і так далі. Дослідники в галузі комп'ютерних мереж давно визнали факт необхідності впровадження інтелектуальних компонентів для автоматизації обслуговування мереж, їх самостійної адаптації та вдосконалення в умовах постійно зростаючих розмірів, складності та гетерогенності.

Методи штучного інтелекту забезпечують інтелектуальний та динамічний контроль і керування мережевими системами. Машинне навчання часто використовують для виявлення вторгнень, несправностей мережевих пристроїв та інших відхилень у функціонуванні мережі, для своєчасного відновлення режиму нормального функціонування. Інтелектуальний аналіз передісторії функціонування мере-

жі дозволяє виявити закономірності, що стосуються конкретної ситуації в мережі. Так, якщо раніше певні рішення приводили до успішного результату, то в подальшому це ж рішення буде застосовуватися під час подібних умов без додаткового аналізу, що дозволяє прискорити реакцію системи на зміну зовнішнього середовища.

Незважаючи на те, що керування інфраструктурою та сервісами інформаційно-телекомунікаційного середовища з урахуванням якості обслуговування орієнтоване на ринок, де рівень обслуговування розглядається провайдерами як потенційний прибуток, керування середовищем повинно здійснюватися економічно ефективно з урахуванням витрат на забезпечення SLA та ELA (рис. 1.9) [19].

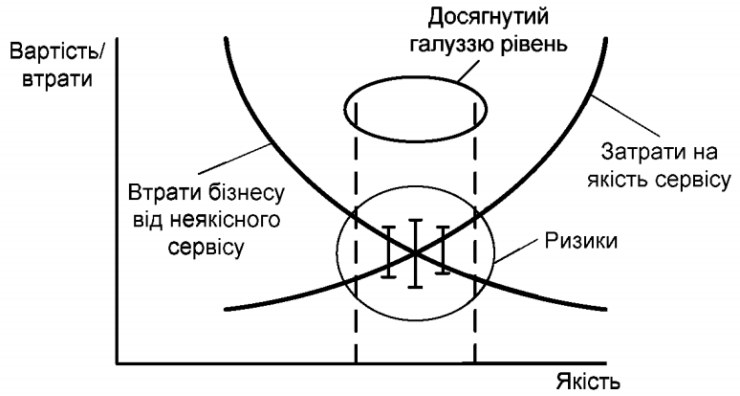


Рисунок 1.9 – Знаходження оптимального відношення рівня якості сервісу та вартості досягнення цієї якості

Вимога максимізації рівня якості послуг і мінімізація пов’язаних із цим витрат знаходяться в природній супе-

речності. Це призводить до необхідності встановлення економічно обгрунтованого обмеження рівня якості для послуг з урахуванням як можливостей компанії, так і досягнутого ІТ-галуззю рівня та очікувань клієнтів.

Мережі провайдерів, корпорацій, комунальні ІТ-системи інтегрувалися, утворивши єдине ІТ-середовище. ІТ-інфраструктура як його основа перетворилися у важливий об'єкт керування. Сегмент ринку систем керування ІТ-інфраструктурою насичений продуктами, створеними на основі IT-Infrastructure Library (ITIL) і концепції IT-Service Management (ITSM). У процесній моделі ITIL виділяють три групи процесів керування : керування інфраструктурою (проектування і планування, розширення, супроводження і технічна підтримка); підтримка обслуговування (керування інцидентами і проблемами, конфігурацією і змінними, релізами); керування наданням послуг (рівнем обслуговування, фінансами, ІТ-послугами, готовністю, неперервністю обслуговування, потужностями). Суттєвими недоліками відомих продуктів є висока вартість, використання закритих фірмових технологій, моделей і алгоритмів керування, відсутність засобів моделювання та керування QoE, засобів зміни оптимізаційних алгоритмів, критеріїв та обмежень.

У працях [19, 20] розроблено і детально описано декомпозиційно-компенсаційний спосіб організації керування рівнем послуг в ІТ-середовищі, заснованого на декомпозиції завдань керування і компенсації негативного впливу окремих чинників, таких як збільшення кількості користувачів, відмови в ІТ-інфраструктурі та інші, за рахунок ви-

ділення додаткових ресурсів критичним застосуванням. Для реалізації даного підходу розробники пропонують використовувати дворівневу модель системи керування з координатором (рис. 1.10).

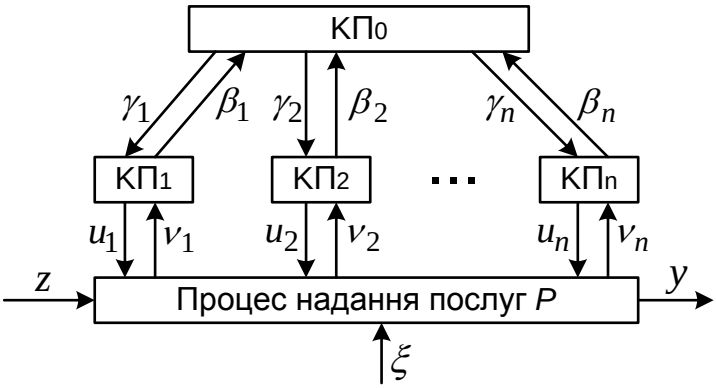


Рисунок 1.10 – Дворівнева система керування IT-інфраструктурою

Показаний на рис. 1.10 блок $КП_0$ виконує функції координатора, а розміщені нижче блоки $КП_1$, ...,$КП_n$ є керуючими підситемами. Р – керований процес, що проходить в IT-інфраструктурі. Координуючі сигнали $\gamma_1, ..., \gamma_n$ впливають на підсистеми керування $КП_1$, ...,$КП_n$, примушуючи їх діяти узгоджено підпорядковуючись єдиній політиці, орієнтованій на досягнення глобальної мети, незважаючи на те, що ця мета може суперечити локальним цілям підсистем. Командні сигнали $u_1, ..., u_n$ від $КП_1$, ...,$КП_n$ до процесу Р є керуючими. Знизу вгору надходять сигнали зворотного зв'язку: від процесу Р до $КП_1$, ...,$КП_n$ – $\nu_1, ..., \nu_n$ і від

керуючих підсистем до координатора $\beta_1, ..., \beta_n$. При цьому на процес P, що являє собою керовану підсистему, крім керуючих сигналів $u_1, ..., u_n$, надходять вхідні сигнали $z \in Z$ (запити користувачів) та сигнали збурювального впливу $\xi \in \Xi$ (несправності в ІТ-інфраструктурі, функціональні відмови, запити інших користувачів, які є перешкодою для даних користувачів та інші впливи, що ускладнюють досягнення цілей керування).

Оскільки система керування функціонує в умовах невизначеності, неповноти і недостовірності інформації, наявності факторів ризику, множини конфліктуючих критеріїв і цілей підсистем системи керування, то від такої системи керування вимагається не досягнення оптимального функціонування ІТ-інфраструктури, що практично неможливо, а покращання якісних характеристик роботи ІТ-інфраструктури. Координатор узгоджує самостійні рішення і дії підсистем системи керування для покращання роботи ІТ-інфраструктури з точки зору якості надання ІТ-послуг. При цьому дії координатора спрямовані на покращання глобальної функції якості надання послуг, а прийняті ним рішення здійснюються в умовах невизначеності.

На початковому етапі даного підходу визначається обґрунтоване значення інтегрального рівня SLA на основі можливих втрат провайдера внаслідок незадовільного рівня обслуговування та його витрат на надання сервісів із визначеним рівнем щодо врахування ризиків. Решта етапів пов’язані з декомпозицією проблеми визначення мінімальної вартості досягнення рівня обслуговування на підпроб-

леми компенсаційного визначення цієї вартості на підмережах, сервісах, ресурсах і технологіях. Задача системи керування полягає в протидії збуренню, реалізуючи ітеративне керування за відхиленням. При цьому визначення сигналу похибки керування здійснюється на основі агрегації метрик, що вимірюються на рівні процесу до метрик, з якими оперує координатор [13, 20].

Використання декомпозиційно-компенсаційного підходу дозволяє створювати ієрархію рішень у керуванні і підтримці узгодженого рівня послуг з урахуванням існуючих ресурсних обмежень і повноважень рівнів у виборі керування, задіюючи можливості верхніх рівнів ієрархії для вибору керування за неможливості реалізації керування на нижніх рівнях. Основними недоліками підходу можна вважати ігнорування індивідуальних користувацьких QoE при обгрунтуванні SLA та агрегації метрик нижчих рівнів, виключення з розгляду надання підсистемам керування прогностичних властивостей.

Оператори та провайдери інформаційно-телекомунікаційних послуг часто мають справу з надмірним резервуванням ресурсів, що призводить до марного споживання енергії та підвищення експлуатаційних витрат. Тому набула поширення практика оверселінгу (overselling), яка полягає в продажі ресурсів у більшому обсязі, ніж є у наявності. При цьому в основу оверселінгу покладено переконання, що більшість клієнтів ніколи повністю не використовують зарезервовані ресурси згідно з тарифним планом, тому невикористані ресурси перепродаються іншим клієнтам. Одночасно з цим розвивається принцип «оплата у мі-

ру використання» (pay-as-you-go), згідно з яким у міру необхідності ресурси можна докупити або відмовитися і сплачувати лише за реально спожиті ресурси. Реалізація цих підходів потребує миттєвої еластичності ресурсів (миттєвого масштабування).

Визначення 1.2.5. Еластичність ресурсів – здатність ресурсів, необхідних для реалізації послуги, виходячи з потреб користувача, швидко надаватися, розширюватися, стискатися і вивільнятися.

Миттєве введення в експлуатацію нових ресурсів часто не можливе, оскільки необхідний певний час для оброблення запиту, перерозподілу ресурсів чи під'єднання фізичних ресурсів до мережі живлення і запуску програмного забезпечення. Такі затримки можуть спричиняти неприйнятно великий час відгуку для критично важливих додатків, до порушення SLA чи зниження QoE, і, як наслідок, втрату доходів провайдера. Вирішити проблему можна шляхом прогнозування потреб додатків у ресурсах, що дозволяє системі керування здійснювати необхідний розподіл або введення до експлуатації додаткових ресурсів із деяким випередженням – до моменту реального запиту на задоволення потреб.

Реалізацію прогнозування потреб додатків у ресурсах можна здійснити на основі технік оброблення та ідентифікації сигналів для виявлення моделей короткострокового попиту ресурсів (сигнатури). Якщо не вдається вилучити сигнатури, то короткострокове прогнозування можна здійснити на основі методів розпізнавання образів. Результат прогнозування можна використати як для випереджуваль-

ного запиту на додаткові ресурси, так і для обмеження ресурсів для некритично важливих додатків до рівня, що не порушує SLA чи не знижує QoE. Зрозуміло, що реалізація прогностичних функцій не повинна призводити до невиправданих витрат, відповідні алгоритми повинні мати невисоку обчислювальну складність.

Таким чином, у загальному випадку для реалізації ефективного керування інформаційно-телекомунікаційною системою передбачається розподіл інтелекту по всіх рівнях системи, надаючи їй властивості самокерування та самоорганізації. Конфігурація та функціональність ІТ-інфраструктури повинні автоматично змінюватися залежно від вимог користувача. При цьому передбачається, що система не тільки реагує на поточні запити користувача, але також аналізує його вподобання і поточне оточення, надаючи системі керування відповідну інформацію. Важливим завданням є скорочення часу відгуку інформаційно-телекомунікаційної системи, що можливо досягти під час надання системі керування здатності прогнозувати потреби та задоволеність користувачів.

1.3. Ідентифікація трафіку та анонімнімність в інфокомунікаційній системі

Сучасні інформаційно-телекомунікаційні мережі характеризуються високими і надвисокими показниками пакетообороту, що зумовлює необхідність пріоритезації трафіку відповідно до вимог рівня обслуговування користувачів та якості мережевих сервісів. Для визначення пріоритету при

формуванні смуги пропускання окремого трафіку необхідно мати інструмент його точної ідентифікації.

Методи ідентифікації трафіку, окрім забезпечення механізмів фільтрації та пріоритезації інформаційних потоків, можуть бути використані як зловмисниками, так і спеціальними державними службами для одержання конфіденційної інформації. Навіть при використанні захищених каналів (наприклад, SSL / TLS або SSH-з'єднання) за допомогою статистичного аналізу пакетованого мережевого трафіку можна розпізнати різні види мережевої активності користувача, наприклад, відрізняти веб-навігацію, віддалене керування робочим столом, передавання файлів, пошук у Google в режимі захищених веб-сесій. Одержана інформація певною мірою може порушувати недоторканість приватного життя.

Методи ідентифікації (класифікації) мережевого трафіку активно досліджуються впродовж останніх років. Було запропоновано багато методів класифікації трафіку, побудованих на аналізі портів призначення, корисного навантаження та характеристик потоку пакетів [21]. Усі ці методи мають свої переваги і недоліки, обмеження в застосуванні (рис. 1.11).

Історично склалося так, що багато додатків використовують «добре відомі» порти на своїх локальних хостах. У цьому разі завдання класифікатора полягає в пошуку TCP SYN-пакетів, щоб визначити серверну сторону нового клієнт-серверного TCP-з'єднання. Потім щоб зробити висновок про додаток, який генерує трафік, розглядається номер порту призначення пакета в списку зареєстрованих портів

IANA [21]. UDP використовує порти так само, але без встановлення з'єднання.

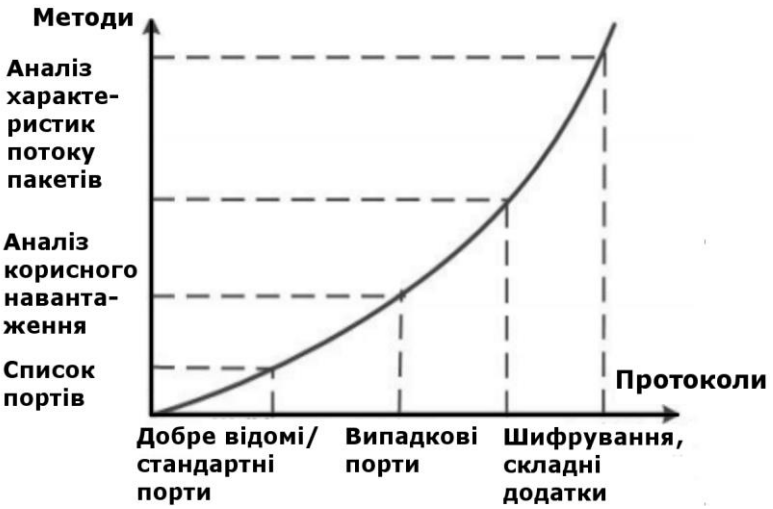


Рисунок 1.11 – Методи класифікації пакетного мережевого трафіку

До безсумнівних переваг методу, побудованого на списку портів, відносять простоту його реалізації і високу швидкість роботи. Однак цей метод має ряд недоліків. По-перше, певні додатки можуть не мати своїх портів, зареєстрованих в IANA, наприклад пірингові додатки. Додатки можуть використовувати відмінні від добре відомих портів, щоб обійти обмеження контролю доступу в операційній системі. Наприклад, непрівілейовані користувачі на UNIX-подібних системах можуть примусово запустити HTTP-сервери на портах, відмінних від 80. Також в окремих ви-

падках порти серверу надаються динамічно у міру необхідності. У деяких випадках шифрування IP-рівня може сплутати TCP- та UDP-заголовки, що робить неможливим визначення фактичного номера порту. У праці [22] показано, що на основі аналізу портів не вдається визначити 30-70 % потоків Інтернет-трафіку.

Щоб уникнути повної залежності від номерів портів і зібрати відомості про використаний протокол, багато сучасних комерційних продуктів використовують відновлення стану сеансу і прикладну інформацію із вмісту кожного пакета. При цьому виділяють чотири різні рівні перевірки.

Перший рівень перевірки полягає в пошуку сигнатур, визначених для відомого протоколу, в корисному навантаженні прикладного рівня. Так, наприклад, HTTP-пакет починається з команди, що йде за URL і версією протоколу, у той час як більшість пакетів у мережі обміну файлами має поля, що містять розмір корисного навантаження.

Другий рівень перевірки – синтаксичний. Він може розглядатися як більш точна версія сигнатурної перевірки, оскільки спрямований на перевірку правильності переданих даних із синтаксичної точки зору (наприклад, передбачається, що корисне навантаження HTTP повинне містити HTTP-заголовки). У цьому разі необхідно декодувати всі поля, що містяться у повідомленні, та оцінити, чи повідомлення правильно сформоване.

Третій рівень контролю пов'язаний з протоколом відповідності. Наприклад, цей рівень контролює, що на HTTP GET-запит від клієнта дійсно надходить відповідь від сервера. Така форма контролю є більш точною, оскільки вона

може перевіряти відповідно до специфікації реальну поведінку протоколу.

Четвертий рівень контролю відносять до семантики даних. Наприклад, алгоритми цього рівня здатні перевірити, чи об'єкт, що передається за протоколом HTTP, є зображенням або якою-небудь іншою формою змісту. Такий контроль дуже корисний для виявлення «тунелів», в яких додаток використовує інший протокол для транспортування даних. На даний час це найбільш неформалізований рівень.

Аналіз корисного навантаження дозволяє класифікувати трафік незалежно від його порту призначення, однак цей аналіз істотно завантажує пристрій ідентифікації трафіку. Такий пристрій повинен мати великі знання про семантику прикладних протоколів, повинен бути досить потужним, щоб виконувати одночасно аналіз потенційно великої кількості потоків. Застосування цього підходу може бути ускладнене або неможливе, коли йде мова про запатентовані протоколи або зашифрований трафік.

Значного прогресу у вирішенні проблеми класифікації трафіку вдалося досягти в рамках підходу, побудованого на аналізі потоку пакетів (flow-based techniques).

Визначення 1.3.1. Потік пакетів – ряд пакетів, що поділяють однаковий кортеж із п’яти елементів: IP-адреса джерела та одержувача, номер портів джерела і одержувача, номер протоколу.

При цьому TCP-потоки, як правило, обмежують тривалістю до 600 с, а UDP-потоки обмежуються максимальною тривалістю між прибуттям пакетів, що становить 64 с.

Аналіз унікальних статистичних характеристик дво-

спрямованого потоку пакетів методами інтелектуального аналізу даних дозволяє одержати вирішальні правила для класифікації трафіку в режимі реального часу (рис. 1.12).

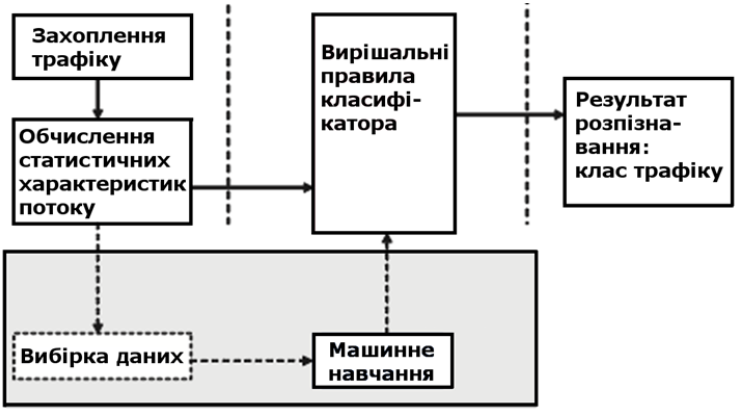


Рисунок 1.12 – Здатний навчатися класифікатор трафіку

Двоспрямований потік пакетів характеризується великою кількістю ознак розпізнавання: кількість пакетів та байтів у прямому/зворотному напрямках потоку; відношення кількості пакетів до кількості байтів корисного навантаження в прямому/зворотному напрямках; середнє значення, мінімальне значення, перша та третя квартилі, медіана та дисперсія розміру корисного навантаження (в байтах) для вхідних/вихідних пакетів двоспрямованого потоку; відношення кількості пакетів малого розміру (до 50 байтів корисного навантаження) до загальної кількості пакетів у прямому/зворотному та в обох напрямках; відношення кількості пакетів великого розміру (більше 1300

байтів корисного навантаження) до загальної кількості пакетів у прямому/зворотному та в обох напрямках; мінімальне, максимальне та середнє значення тривалості часового інтервалу між прибуттям пакетів у прямому/зворотному напрямках; відношення кількості пакетів без корисного навантаження до загальної кількості пакетів у прямому/зворотному та в обох напрямах; кількість прапорців ACK / PSH у потоці прямого/зворотного напрямку та інші.

Навчальні набори даних можуть бути сформовані в процесі трасування трафіку на зеркальних портах (Mirror Port) межових комутаторів/маршрутизаторів утилітою Tcp-Dump із подальшим розщепленням потоків і обчисленням ознак розпізнавання за допомогою утиліти NetMate. Апріорна класифікація зразків трафіку комп'ютерних додатків може бути здійснена за результатами моніторингу відповідних сокетів утилітою CurrPorts (Windows) або Net-ActivityViewer (Linux) [23].

Оскільки шифрування не забезпечує цілковитої конфіденційності внаслідок витоку інформації про активність користувача при спостереженні за ознаками потоку пакетів, то це потребує додаткових заходів, які називаються маскуванням трафіку і застосовуються на кінцях відповідного маршруту, що пролягає в незахищеній мережі (рис. 1.13). Методи маскування трафіку покликані модифікувати потік пакетів таким чином, щоб звести кількість інформації, яка «просочується», до мінімуму.

Рисунок 1.13 –Захищений канал передавання трафіку

Модифікація потоку пакетів здійснюється як за рахунок уставлення (padding) у потік фіктивних повідомлень та затримок, так і шляхом фрагментації (fragmentation) потоку, що призводять до зміни довжини, кількості пакетів та інтервалів часу між їх надходженням. На рисунку 1.14 показана послідовність модифікації трафіку.

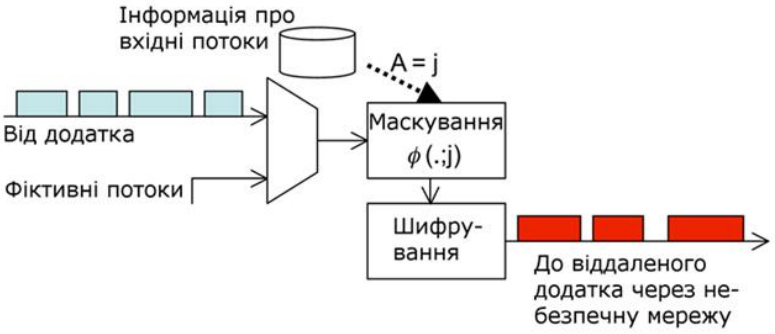


Рисунок 1.14 – Загальна схема модифікації потоку трафіку

Кожний вхідний потік A = j маскується потоком φ(j) та шифрується. Наявність апріорної інформації про статистичні властивості потоків заданого алфавіту класів у вигляді функції розподілу ймовірностей для ознак потоку дозволяє оптимізувати процес маскування. При цьому ознаки результуючого потоку можуть спостерігатися в точці моніторингу. На боці одержувача відбуваються зворотні операції (розшифрування та демаскування).

Найпростіший метод маскування полягає у встановленні незмінного розміру пакетів даних, який дорівнює максимальному розміру MTU (Maximum Transmission Unit), що підтримується протоколом. Проте для багатьох протоколів застосування такого підходу призведе до збільшення обсягу даних, що передаються по каналу. Найбільш відомі такі підходи до модифікації довжини пакетів [24]:

1) лінійна вставка (Linear padding), де всі довжини пакетів збільшені до найближчого кратного 128 або MTU, дивлячись що менше;

2) експоненційна вставка (Exponential padding), де всі довжини пакетів збільшені до найближчого степеня числа 2, або довжини MTU, дивлячись що менше;

3) вставка Миші-Слони (Mice-Elephants padding) – якщо довжина пакета менша або дорівнює 128, то довжина пакета збільшиться на 128, інакше пакет доповнюється до довжини MTU;

4) килимок для MTU (Pad to MTU), де всі довжини пакетів збільшені для досягнення довжини MTU;

5) випадкова вставка до MTU (Packet Random MTU padding) – якщо M є довжиною MTU, а L є довжиною вхід-

ного пакета, то кожний пакет буде збільшено на значення, що обирається випадково з множини {0, 8, 16, ..., M − L}.

Перелічені підходи використовуються в багатьох відомих архітектурах захищеного зв'язку, побудованих на протоколах SSH, TLS та IPSec. Однак у праці [24] на прикладі задачі ідентифікації трафіку завантаження веб-сторінок показано, що застосування цих протоколів не забезпечує надійного маскування. При цьому точність статистичного класифікатора при ідентифікації трафіку становила 98 %.

Оскільки у загальному випадку маскуванню підлягає не лише реальна довжина пакетів, але й інші характеристики потоку пакетів, то завдання маскування може бути сформульоване як задача пошуку шаблону трафіку певного класу і шаблону дій, необхідних для його трансформації у шаблон захищеного трафіку. На рисунку 1.3.5 показано уточнену схему модифікації трафіку.

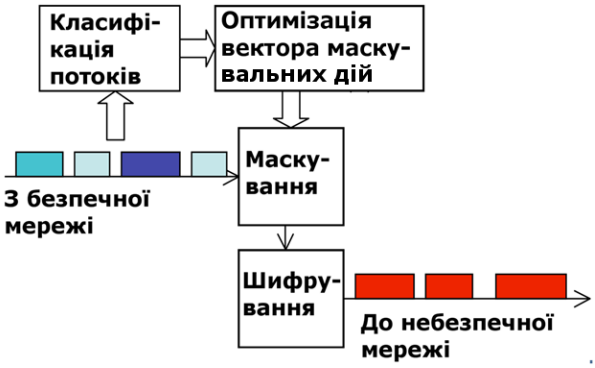


Рисунок 1.15 – Схема маскування трафіку, побудована на шаблонах

Шаблон трафіку можна описати системою обмежень на ймовірність появи значень ознак відповідного потоку пакетів, а шаблон дій – характеристиками фіктивного потоку, що додається до початкового, та параметрами фрагментації. Класифікатор трафіку в схемі (рис. 1.3.5) дозволяє ідентифікувати шаблон трафіку, який відповідає одному з апріорно відомих класів. Відмінності даного шаблону від шаблону ідеального зашифрованого трафіку є вхідною інформацією для алгоритму оптимізації шаблону дій щодо модифікації початкового потоку трафіку. Під час оптимізації, крім наближення шаблону трафіку до ідеально зашифрованого, повинна розв'язуватися задача мінімізації накладних витрат, пов'язаних як із процесом маскування, так і з процесом транспортування трафіку надлишкового обсягу. При цьому типовий модуль маскування передбачає такі процедури:

1) формування черги пакетів перед надсиланням;

2) вибір пакетів із черги згідно з процедурою таймінгу (для регулювання інтервалів часу між надходженням пакетів);

3) керування довжиною пакетів відповідно до методу вставлення фіктивних пакетів;

4) генерація фіктивних пакетів;

5) формування результуючого захищеного трафіку.

Підвищення ефективності маскування та продуктивності основних протоколів є актуальними для реалізації анонімних комунікацій, наприклад під час розроблення сервісів миттєвого обміну повідомленнями в соціальних мережах. У соціальних мережах маскування трафіку може при-

ховати від зловмисника інформацію про те, хто з ким спілкується.

Застосування різних методів маскування трафіку дозволяє покращити підтримку конфіденційності інформації, проте такі заходи можуть ввести в оману механізми пріоритезації та фільтрації трафіку, що може призвести до зниження QoS для високопріоритетних додатків.

З метою приховування контенту і типу трафіку, IP-адреси та реквізитів користувача часто використовуються такі централізовані засоби технічної анонімізації, як проксі-сервери та сервіси віртуальних приватних мереж.

Визначення 1.3.2. Проксі-сервер (proxy server) – сервер (комплекс програм) в комп'ютерних мережах, що дозволяє клієнтам здійснювати непрямі запити до інших мережевих служб [25].

Визначення 1.3.3. Віртуальні приватні мережі (Virtual Private Network, VPN) – мережі, організовані у вигляді зашифрованого тунелю, що проходить поверх (або всередині) вже існуючих мереж [25].

Централізовані засоби технічної анонімізації характеризуються невисокою надійністю, що зумовлено наявністю єдиного центру контролю та єдиної точки відмови в обслуговувані. Тому останнім часом дістає поширення використання засобів децентралізованої технічної анонімізації, які характеризуються підвищеною відмовостійкістю сервісів анонімізації навіть за великої кількості користувачів та вищим ступенем захищеності від цензури. До найбільш популярних децентралізованих сервісів анонімізації відносять I2P- та Tor-мережі.

Визначення 1.3.4. I2P (Invisible Internet Project) – це анонімна оверлейну (поверх Інтернету), самоорганізована розподілена мережа, в якій неможлива цензура і відстеження користувача.

У I2P-мережах використовується хешування та шифрування IP-адрес вузлів мережі. Під час пересилання повідомлення використовуються чотири рівні шифрування (наскрізне, часникове, тунельне, а також шифрування транспортного рівня). Перед шифруванням кожен мережевий пакет автоматично доповнюється невеликою випадковою кількістю випадкових байтів (вставлення фіктивних даних). Мережа надає додаткам простого транспортного механізму для анонімного і захищеного пересилання повідомлень один одному. Основним недоліком є низька швидкість передавання даних та неможливість звернення до будь-яких ресурсів Інтернет. Працюючи з I2P, можна звертатися лише до I2P-ресурсів (I2P-сайтів, пошти, трекерів тощо).

Визначення 1.3.5. Tor (The Onion Router) – це система проксі-серверів, в якій клієнт з'єднується з Інтернетом через ланцюжок вузлів [26].

У мережі Tor, як правило, ланцюжок складається з трьох вузлів, кожному з яких невідомі адреси клієнта і ресурсу одночасно. Крім того, Tor шифрує повідомлення окремо для кожного вузла, а відкритий трафік видно тільки вихідному вузлу. Tor складається з десятків керуючих (вхідних) вузлів, близько десятка тисяч вузлів-посередників та декількох тисяч вихідних вузлів.

Трафік у зворотному напрямку надходить у відкритому

вигляді, шифрується на вихідному вузлі за тимчасовими симетричними ключами і передається далі по ланцюжку вузлів.

На рисунку 1.16 показано спрощену схему взаємодії клієнта з сервером через Tor-мережу [26].

Рисунок 1.16 – Спрощена схема взаємодії клієнта з сервером через Tor-мережу

Застосування Tor лишає можливості зловмиснику, який відстежує інтернет-з'єднання клієнта, визначити, які сайти були відвідані, а відвідані сайти лишає можливості дізнатися про фізичне перебування клієнта. Сервіс Tor працює з додатками, що використовують TCP-протокол (веб-браузери, клієнтські програми служб миттєвого обміну повідомленнями, програми віддаленого входу в систему та ін.).

Трафік вхідних та вихідних вузлів Tor може бути перехоплений і піддаватися інформаційним атакам шляхом аналізу довжини пакетів, інтервалів часу між повідомленнями,

характеру зміни напрямків передавання повідомлень з метою виявлення шаблонів комунікації та розкриття топології мережі.

Джерелом розкриття відомостей про користувача (ідентифікація користувача в Інтернет) може бути інформація, яку передає програмне забезпечення в обхід блокувальних міжмережевих екранів, проксі-серверів тощо, для забезпечення нормальної та ефективної роботи в складних мережевих умовах. Це, як правило, різного роду дані, передавання яких передбачено специфікацією до програмного продукту.

Таким чином, поширення мережевих додатків, які динамічно змінюють порти транспортних протоколів, використовують шифрування та інкапсуляцію трафіку в тунельний протокол, призводить до низької ефективності класифікації трафіку на основі портів чи корисного навантаження. Вирішення цих проблем пов'язується з використанням методів інтелектуального аналізу даних, де, як правило, ознаками класифікації трафіку є статистичні характеристики потоку пакетів. При цьому застосування методів маскування трафіку дозволяє знизити точність його класифікації і відповідно знизити витікання конфіденційної інформації про мережеву поведінку користувача. Однак замаскований трафік, як правило, значно збільшує вимоги до пропускної здатності каналу, вводить в оману механізми і політики QoS. Забезпечення анонімності поведінки та місця перебування автора за рахунок децентралізованих систем типу Tor та I2P так само збільшує накладні витрати і сповільнює мережевий трафік користувача, однак деаноні-

мізація вимагатиме на декілька порядків більше витрат та зусиль. Тому залишається відкритим питання компромісу між захищеністю каналу зв'язку та накладними витратами для її забезпечення.

1.4. Виявлення атак та керування безпекою в інфокомунікаційній системі

Ефективне керування ресурсами інформаційно-телекомунікаційної системи обов'язково вміщує забезпечення надійності функціонування та безпеки інформації. Існує необхідність виявляти та протидіяти інформаційним впливам протиправного та диструктивного характеру, які ще називають атаками, або вторгненнями. Для цього використовують системи виявлення атак (Intrusion Detection System, IDS), що займаються аналізом використання доручених їм ресурсів, і, у разі виявлення яких-небудь підозрілих чи просто нетипових подій здатні вживати певних самостійних дій щодо виявлення, ідентифікації та усунення їх причин.

У процесі розвитку комп'ютерних та мережевих технологій список можливих типів мережевих атак на інформаційно-комунікаційні системи постійно розширюється. У загальному випадку можна виділити чотири групи найбільш поширених атак [27]:

1) U2R (user-to-root) атаки, що передбачають одержання зареєстрованим користувачем привілеїв локального суперкористувача (адміністратора);

2) R2L (remote-to-local) атаки, які характеризуються одержанням доступу незареєстрованого користувача до комп'ютера з боку віддаленої машини;

3) Probe атаки, що полягають у скануванні мережевих портів із метою одержання конфіденційної інформації;

4) DoS (Denial of Service) атаки – це мережні атаки, спрямовані на виникнення ситуації, коли в системі, що атакується, відбувається відмова в обслуговуванні.

Залежно від джерела виявлення атак розрізняють системи IDS-рівнів хосту (Host-based IDS), мережі (Network Intrusion Detection), рівня додатків (Application IDS) та гібридні (Hybrid IDS), які поєднують комбіновані методи [28]. Перші ідентифікують вторгнення, аналізуючи події і трафік, що надходить на окремий комп'ютер, у той час як інші – досліджують мережевий трафік. Системи рівня додатків, як правило, розміщуються між веб-сервером і, наприклад SQL-сервером.

Існує багато методів виявлення атак, проте на даний момент можна виділити такі основні підходи:

1) аналіз сигнатур, побудований на простому збігові послідовностей зі зразком атаки;

2) аналіз аномалій, побудований на контролі частоти подій чи виявленні статистичних аномалій;

3) комбінований підхід.

Аналіз сигнатур був першим методом, застосованим для виявлення атак. У вхідному пакеті проглядається байт за байтом і порівнюється з сигнатурою (підписом) – характерним рядком програми, що свідчить про характеристику шкідливого трафіку. Такий підпис може містити ключову

фразу, команду або їх послідовність, пов'язаних з атакою. Системи аналізу сигнатур досить швидкі, оскільки не здійснюють повний і глибокий аналіз пакета та протоколу. При цьому вирішальні правила легко дописувати, редагувати і налаштовувати. Високий рівень підтримки комп'ютерним співтовариством у справі формування сигнатур для нових небезпек сприяє успіху застосування подібних систем при виявленні хакерських впливів на початкових етапах: простим атакам, як правило, передують дії, які легко розпізнати. Однак збільшення обсягу сигнатур значно знижує швидкість системи. За даними, наведеними в праці [29], в середньому кожен день з'являється близько 100 нових атак, що фізично ускладнює оновлювати бази даних сигнатур за такі проміжки часу. Тому велика кількість атак навіть при їх невеликих модифікаціях може не виявлятися такою системою. Система, побудована на аналізі сигнатур, здатна виявляти лише вже відомі атаки, сигнатури яких наявні в базі.

В основу методу аналізу аномалій покладено принцип порівняння поточних кількісних показників системи (частота звернень до служб, навантаження на вузли мережі тощо) з еталонним станом, сформованим за результатами моніторингу нормального функціонування інфокомунікаційної системи. Цей принцип дозволяє з успіхом виявити факт вторгнення і джерело загрози (внутрішнє або зовнішнє). У разі якщо джерело загрози є внутрішнім, то система аналізу аномалій дозволяє визначити, чи атака, яка відбувається від імені авторизованого користувача, надходить насправді від авторизованого користувача або від зловми-

сника, який маскується під нього. При цьому система аналізу аномалій дозволяє виявляти незначні модифікації відомих атак, але під час практичної реалізації такої системи існують ускладнення, пов'язані з виявленням та врахуванням раніше невідомих типів атак і впливів. Тому при синтезі вирішальних правил системи аналізу аномалій необхідно вирішувати такі задачі :

1) побудову еталонної множини інваріантних ознак нормального (семантично коректного) розвитку обчислювальних процесів в умовах апріорної невизначеності впливів зовнішнього та внутрішнього середовищ;

2) встановлення шкал вимірювання інваріантних ознак;

3) виявлення інформативних інваріантних ознак;

4) синтез вирішальних правил для розпізнавання аномалій.

Для формування еталонного стану інформаційно-телекомунікаційної системи або її підсистеми, як правило, використовують методи контрольованого та неконтрольованого машинного навчання.

Методи контрольованого машинного навчання використовують фіксований набір параметрів оцінювання та апріорні відомості про значення цих параметрів. Період навчання та перенавчання характеризується невеликими часовими рамками. При цьому серед методів контрольованого навчання найчастіше розглядають метод моделювання правил, описову статистику та нейронні мережі [28, 29].

Методи неконтрольованого навчання використовують набір параметрів оцінювання, склад яких може динамічно змінюватися. Навчання відбувається неперервно і еталон-

ний стан із часом адаптується до нової конфігурації. Серед методів неконтрольованого навчання найбільш досліджено застосування описової статистики та моделювання множини станів [30].

У лабораторії MIT Lincoln Labs з метою дослідження ефективності різних методів машинного навчання у завданні синтезу вирішальних правил для виявлення типових атак було сформовано набір зразків поширених на той час атак [27]. При цьому було запропоновано 41 ознаку для кожного TCP / IP-з'єднання. Ознаки були розподілені на три групи.

До ознак першої групи віднесено основні характеристики TCP / IP з'єднання, наприклад, тривалість зв'язку, тип протоколу, сервіс, число переданих байтів від джерела до приймача і у зворотному напрямі, окремі прапорці. Деякі значення ознак визначалися із затримкою впродовж певного часового інтервалу. Друга група ознак містить статистичні характеристики трафіку, обчислені з використанням 2-секундного часового вікна або впродовж більшого часового проміжку. Характеристики поділяють на дві групи: атрибути, що належать до конкретного хосту – комп'ютера чи сервісу. Окремі атаки сканування портів виконуються більше ніж 2 секунди. Тому ряд ознак обробляється вікном у 100 з'єднань. До ознак третьої групи належать ознаки всередині одного окремого з'єднання. На відміну від більшості DoS-атак і сканування портів, R2L та R2U-атаки характеризуются певними нетривалими проявами на окремому комп'ютері. У той час як DoS-атаки та Probing ініціалізують множинні з'єднання за короткий проміжок часу.

Упровадження стека протоколів IPv6, поява нових веб-сервісів і мережевих додатків провокують появу нових схем для реалізації мережевих атак, що зумовлює необхідність розширення і перегляду словника ознак.

У загальному випадку система виявлення вторгнень має ієрархічну організацію, в якій доцільно окремо розглядати локальну і глобальну архітектури. У рамках локальної архітектури реалізуються елементарні складові, які потім можуть бути об'єднані для обслуговування корпоративних систем. Основні елементи локальної архітектури і зв'язки між ними показано на рисунку 1.17 [28].

Рисунок 1.17 – Основні елементи локальної архітектури систем виявлення атак

Сенсори, виконані у вигляді автономних агентів, здійснюють первинне збирання даних. Реєстраційна інформація може вилучатися з системних та прикладних журналів або безпосередньо від ядра операційної системи. Також цю

інформацію можна одержати з мережі за допомогою відповідних механізмів активного мережевого обладнання або шляхом перехоплення пакетів за допомогою встановленої у режим моніторингу мережевої карти.

На рівні агентів (сенсорів) може виконуватися фільтрація даних із метою зменшення їх обсягу. Це вимагає від агентів деякого інтелекту, проте дозволяє розвантажити інші компоненти системи. Агенти передають інформацію в центр розподілу, де відбувається подальша фільтрація, зведення до єдиного формату, збереження в базі даних та передавання даних для статистичного і експертного аналізу.

Якщо в процесі статистичного або експертного аналізу виявляється підозріла активність, відповідне повідомлення спрямовується до блока прийняття рішень, де визначається виправданість тривоги і обирається спосіб реагування.

Глобальна архітектура відображає організацію однорангових і різнорангових зв'язків між локальними системами виявлення атак (рис. 1.18) [28].

На одному рівні ієрархії розміщуються компоненти, що аналізують підозрілу активність з різних точок зору. Наприклад, на хості можуть розміщуватися підсистеми аналізу поведінки користувачів і додатків. Їх може доповнювати підсистема аналізу мережевої активності. Коли один компонент виявляє щось підозріле, то в багатьох випадках доцільно повідомити про це сусідам або для вжиття заходів, або для посилення уваги до певних аспектів поведінки системи.

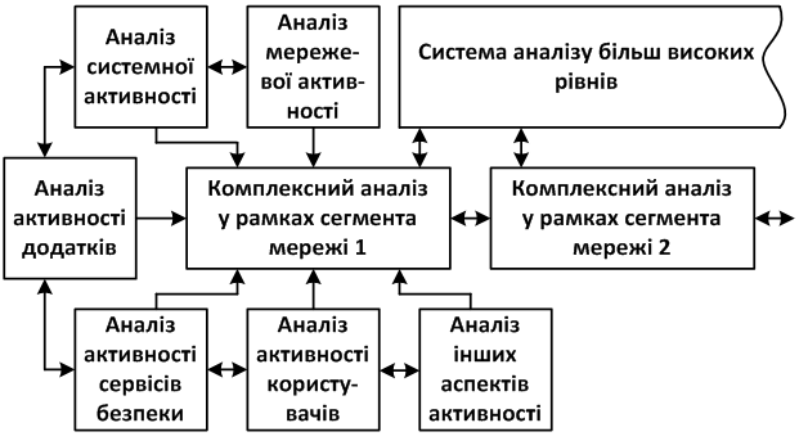


Рисунок 1.18 – Глобальна архітектура систем виявлення атак

Різнорангові зв'язки використовують для узагальнення результатів аналізу та одержання цілісної картини того, що відбувається. Іноді у локального компонента недостатньо підстав для порушення тривоги, але в сукупності підозрілі ситуації можуть перевищити поріг підозрілості. Цілісна картина, можливо, дозволить виявити скоординовані атаки на різні ділянки інформаційної системи та оцінити збитки у масштабі організації.

Група IDWG напрацювала специфікацію формату IDMEF (Intrusion Detection Message Exchange Format) – формат обміну даними між компонентами IDS [28]. Формат IDMEF використовується для передавання попереджувальних повідомлень про підозрілі події між системами виявлення атак. Цей формат повинен забезпечити суміс-

ність між комерційними і вільно поширюваними IDS і можливість їх спільного використання для забезпечення найвищого рівня захищеності.

Забезпечення захищеності інформаційно-телекомунікаційної системи потребує введення надлишковості конфігурації системи, пов'язаного з використанням додаткових засобів захисту інформації, що споживають певний обсяг системних ресурсів. Тому існує потреба у розробленні адаптивної системи керування захистом інформаційно-телекомунікаційної системи, яка буде здатна змінювати рівень захищеності об'єктів системи залежно від їх інформаційної цінності. При цьому показник інформаційної цінності ресурсів може визначатися реальною вартістю або обсягом збитків у разі знищення чи втрати конфіденційності інформації, що в них міститься. Однак цінність інформації змінюється з плином часу. Залежно від динаміки і типу інформації можна виділити такі випадки [31]:

– цінність інформації стаціонарна в часі (бази даних, інформація в яких актуальна впродовж тривалих періодів часу);

– цінність інформації постійно збільшується (бази даних у момент накопичення інформації);

– цінність інформації постійно зменшується (бази даних, актуальність інформації в яких знижується);

– цінність інформації має верхній екстремум (інформація, що в певний момент часу змінює свій статус, наприклад про разробки, які патентуються, про передвиборну кампанію);

– цінність інформації має нижній екстремум (теоретично можливий випадок).

Адаптивна система керування захистом повинна динамічно оцінювати цінність інформаційних ресурсів і адаптивно змінювати захищеність інформаційно-телекомунікаціної системи. Архітектура такої системи може мати вигляд, показаний на рисунку 1.19 [31].

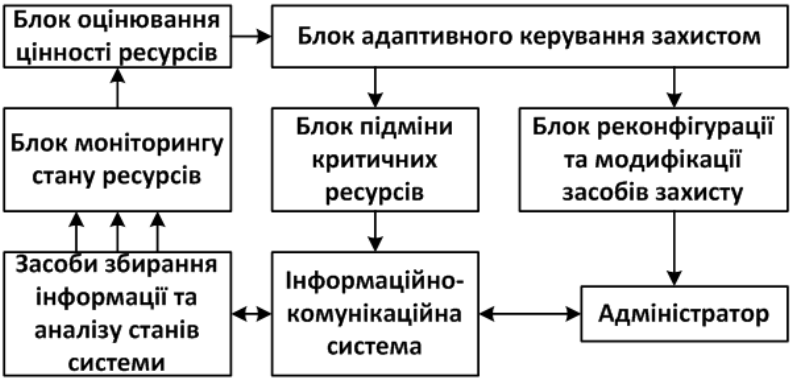


Рисунок 1.19 – Архітектура системи адаптивного керування захищеністю інформаційно-текомунікаційної системи

До засобів збирання інформації та аналізу стану системи відносять класичні системи виявлення вторгнень і засоби захисту (антивіруси, міжмережеві екрани, апаратні засоби захисту та контролю доступу) та засоби аналізу захищеності інформаційно-телекомунікаційної системи. Ці засоби при виявленні підозрілої активності, яку не можна достовірно ідентифікувати, відносять її до потенційних

атак.

Блок моніторингу станів ресурсів інформаційно-телекомунікаційної системи одержує інформацію від засобів збирання інформації та аналізу станів системи. При цьому кожен запис про підозрілу активність конвертується у певний формат, який вносить дані про суб'єкта (ініціатора події), параметри дій суб'єкта з реалізації атаки, можливі цілі, час, місце, задіяні засоби і ступінь успішності атаки. Крім того, завданням блока моніторингу є формування ймовірностей вторгнень.

Блок оцінювання цінності ресурсів виконує подальший аналіз інформації, одержаної від блока моніторингу безпеки. Для кожного ресурсу обчислюється непрямий показник його інформаційної цінності. Ці дані є базовими для адаптивної зміни рівня захищеності ресурсів із урахуванням цінності інформації, яку вони обробляють чи зберігають.

Блок адаптивного керування захищеністю ресурсів інформаційно-комунікаційної системи диференційовано для кожного ресурсу виконує порівняння поточної цінності ресурсу зі значеннями, одержаними на попередньому кроці аналізу, і визначає необхідний рівень захищеності для кожного з них. Потім інформація передається блоку реконфігурації та модифікації засобів захисту для налаштування нових параметрів засобів захисту або зміни їх конфігурації. Крім того, цей блок забезпечує необхідну інформацію для блока підміни критичних ресурсів, що дозволяє додатково підвищити захищеність критичних даних.

Блок реконфігурації та модифікації засобів захисту здійснює цілеспрямовану зміну структури системи захисту

інформації шляхом перерозподілу засобів захисту від об'єктів, що мають мінімальну інформаційну цінність, або об'єктів, що не зазнають на цей момент атак, до об’єктів, на які спрямовано на цей час диструктивний вплив.

Блок підміни критичних ресурсів виробляє емуляцію вразливостей, системних відомостей і критичних інформаційних ресурсів, на основі даних блока визначення потенційної цінності ресурсів. При цьому залежно від характеру дій і цілей зловмисника можуть застосовуватися як приховування та дезінформація, так і повна підміна критичних даних.

Адміністратор виконує загальний контроль всієї системи адаптивного керування, відповідає за зв'язок і взаємодію інформаційно-телекомунікаційної системи із засобами збирання інформації та оцінювання стану системи. Крім того, він приймає остаточне рішення про емуляцію вразливостей, реконфігурацію системи захисту і подальші дії щодо порушників.

Таким чином, у сучасних інформаційно-телекомунікаційних системах все частіше впроваджуються засоби інформаційної безпеки, побудовані на ідеях і методах машинного навчання та розпізнавання образів. Архітектура системи виявлення вторгнень набирає ієрархічної структури, на всіх рівнях якої знаходяться здатні до взаємодії інтелектуальні автономні агенти. При цьому керування безпекою інформаційно-телекомунікаційної системи полягає у своєчасній реакції на мережеві атаки та забезпеченні економічно доцільного рівня захищеності об’єктів системи відповідно до їх інформаційної цінності.

1.5. Контрольні запитання та завдання для самопідготовки

1. Які функції виконує IP-мережа?

2. Які найбільш важливі характеристики пакетних IP-мереж?

3. Що називається смугою пропускання?

4. Як визначається затримка доставляння пакета?

5. Що називається варіацією затримки пакета?

6. Як визначається коефіцієнт втрати пакетів?

7. Як визначається згідно з рекомендаціями Y.1541 коефіцієнт помилок пакетів?

8. Що розуміється під якістю обслуговування QoS (Quality of Service) телекомунікаційної системи?

9. Які існують класи якості обслуговування телекомунікаційної системи?

10. Які відповідності між класами якості обслуговування і додатками?

11. Що називається якістю обслуговування QoE (Quality of Experiense)?

12. Які активні об’єктивні методи оцінювання якості послуги?

13. Які пасивні об’єктивні методи оцінювання якості послуги?

14. Що називається угодою про рівень послуг SLA (Service-level agreement)?

15. Що називається доступністю мережі та які її показники?

16. Що називається доступністю сервісу?

17. Які існують типии SLA?

18. Що називається угодою про рівень очікуваної якості послуг ELA (Experience Level Agreements)?

19. Що називається мережею майбутнього FN (Future Network)?

20. Що відносить до засобів керування ресурсами?

21. Які функції виконує класифікатор пакетів?

22. Як здійснюється вирівнювання трафіку?

23. Як здійснюється маршрутизація за пунктом призначення пакетів?

24. Які особливості технології маршрутизації MPLS (Multiprotocol Label Switching)?

25. Що називається мережевим агентом?

26. Що називається інтелектуальним агентом?

27. Що називається еластичністю ресурсів?

28. Які основні методи ідентифікації мережевого трафіку?

29. Що називається потоком пакетів?

30. Які основні процедури виконує типовий модуль маскування трафіку?

31. Що називається проксі-сервером?

32. Що називається віртуальною приватною мережею VPN (Virtual Privats Network)?

33. Яке призначення I2P-мереж?

34. Яке призначення Tor (The Onion Router)-мереж?

35. Яке призначення системи виявлення атак IDS (Intrusion Detection System)?

36. Які основні типи атак на інфокомунікаційну систему?

37. Які основні методи виявлення атак?

38. Яка ідея покладена в основу методу аналізу сигнатур для виявлення атак?

39. Яка ідея покладена в основу методу аналізу аномалій для виявлення атак?

40. У чому полягає комбінований метод виявлення атак?

РОЗДІЛ 2
КЕРУВАННЯ РЕСУРСАМИ ТА ПЛАНУВАННЯ ЗАВДАНЬ В ІНФОКОМУНІКАЦІЙНІЙ СИСТЕМІ

2.1. Системи розподілених обчислень

Потреба в обчислювальних ресурсах суттєво зросла останніми десятиліттями у багатьох галузях наукових досліджень. Із збільшенням кількості задач, що розв'язуються в мережі, мережеві додатки породжують постійно зростаюче навантаження на сервери, що надають послуги. У зв'язку з цим великого поширення дістали системи паралельного та розподіленого оброблення даних, що містять обчислювальні вузли (сайти), об'єднані мережею передавання даних.

Сучасні розподілені обчислення мають усталені напрями та тенденції розвитку. Прийнято виділяти такі сегменти в області розподілених обчислень: кластерні системи, GRID-мережі та хмаринні технології. При цьому технологіям розподілених обчислень властива конвергенція, суть якої полягає у системному сприйнятті обчислювального середовища, що передбачає об’єднання ресурсів та уніфікацію технологій розподіленого оброблення даних.

Визначення 2.1.1. Обчислювальний кластер – група комп'ютерів, об'єднаних високошвидкісними каналами зв'язку і які з точку зору користувача являють єдиний апаратний ресурс.

Як правило, вузли кластерних систем не розподілені географічно і керування ними здійснюється проміжним

програмним забезпеченням централізовано. Кластери можуть бути однорідними за складом апаратного забезпечення або складатися з набору різних за конфігурацією (гетерогенних) вузлів-обробників. На вхід системи надходять прикладні задачі оброблення, які можуть мати суттєво відмінні ресурсні вимоги.

У світі спостерігається тенденція нарощування великого парку недостатньо завантажених обчислювальних потужностей. З метою підвищення їх віддачі шляхом об'єднання в систему колективного доступу була створена та практично реалізована концепція GRID-середовища.

Визначення 2.1.2. GRID-системою називається система для забезпечення інтеграції, віртуалізації та керування послугами і ресурсами в розподіленому, гетерогенному середовищі, що підтримує колекції користувачів і ресурсів (віртуальних організацій) у традиційних адміністративних і організаційних доменах (реальних організацій) (рис. 2.1) [32].

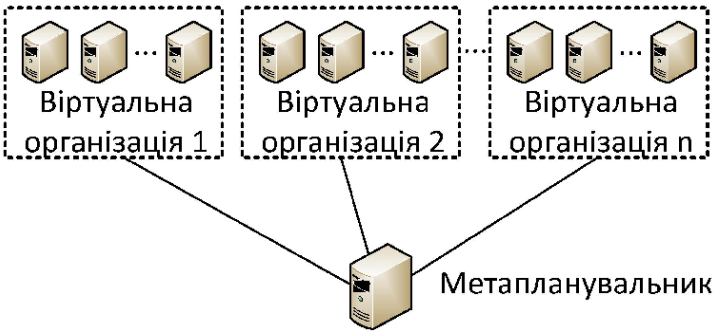


Рисунок 2.1 – Схема організації GRID-обчислень

Визначення 2.1.3. GRID-мережа (або метакомп'ютер) – мережа гетерогенних географічно розподілених обчислювальних ресурсів, що використовуються для паралельного оброблення обчислювальних завдань [33].

GRID являє собою програмно-апаратну інфраструктуру для роздільного використання обчислювальних вузлів, мереж, баз даних та інших ресурсів, які знаходяться в юрисдикції різних географічно розподілених організацій.

Для керування ресурсами GRID використовується проміжне програмне забезпечення, причому керування найчастіше децентралізоване.

Відмінними особливостями GRID-систем є [32, 33]:

1) розподіленість компонентів – вузли системи можуть знаходитися в географічно віддалених один від одного регіонах, що впливає на оперативність взаємодії;

2) метакомп'ютер може динамічно змінювати конфігурацію – система підтримки прозоро для користувача здійснює розподіл задач за компонентами системи з урахуванням динамічного під’єднання/від’єднанняя віддалених ресурсів;

3) неоднорідність системи – до складу GRID-системи можуть входити вузли з різним складом програмно-апаратних ресурсів;

4) метакомп'ютер об'єднує ресурси різних організацій, кожна з яких може мати власну політику доступу до ресурсів.

З точки зору користувача відмінною особливістю таких систем є відсутність контролю над множиною задач, що обробляються на кожному конкретному вузлі. Крім того,

наперед відомо, які ресурси матиме у розпорядженні система на певний момент часу. Також відмінною особливістю є націленість GRID-системи на розв’язання обчислювально трудомістких наукових задач.

За типом сайтів GRID-системи можна поділити на два типи:

1) GRID робочих станцій, що об'єднує звичайні домашні та офісні комп'ютери або мобільні пристрої з метою використання їх під час простою;

2) сервісні GRID, що об'єднують спеціально виділені машини з метою їх використання в монопольному режимі.

Реалізація GRID робочих станцій (так званих добровільних GRID) часто має клієнт-серверну або однорангову архітектуру. У разі клієнт-серверних GRID робочих станцій на обчислювальних вузлах встановлюється і настроюється клієнтське програмне забезпечення, яке виконує періодичні запити віддаленому серверу на наявність задач для своєї платформи. Якщо на сервері такі задачі є, то клієнтська машина завантажує задачу у вигляді виконав-чого файла з необхідними даними і запускає його. Результат роботи додатка повертається назад на сервер. Однорангові GRID робочих станцій мають встановлені на кожному вузлі програмне забезпечення клієнта та серверу, що дозволяє користувачам мережі як надавати свої обчислювальні потужності іншим учасникам, так і використовувати чужі ресурси у своїх цілях. GRID робочих станцій дозволяють концентрувати великі обчислювальні ресурси і характеризуються при цьому мінімальною собіварстістю.

Як подальший розвиток GRID-комп’ютингу була за-

пропонована концепція хмарних обчислень (Cloud computing), що передбачає віддалену роботу з ресурсом, обсяг якого може варіюватися залежно від потреб.

Концепція «хмарних обчислень» полягає в перенесенні організації обчислень на web-сервіси. Програми запускаються і видають результати роботи у вікно стандартного веб-браузера на локальному комп'ютері користувача, при якому всі додатки та їх дані, необхідні для роботи, знаходяться на віддаленому сервері у глобальній мережі Інтернет.

Визначення 2.1.4. Хмарні обчислення – це модель, яка забезпечує зручний мережевий доступ за вимогою до спільних конфігурованих обчислювальних ресурсів (мереж, серверів, сховищ даних, додатків і сервісів), що оперативно надається з мінімальними зусиллями щодо керування і взаємодії з сервіс-провайдером [34].

Визначення 2.1.5. Оброблення даних у хмарах – це парадигма, згідно з якою програми і дані постійно зберігаються у віддалених від користувача центрах оброблення даних (ЦОД) і доступ до обчислювального сервісу здійснюється прозоро для користувача за допомогою Інтернету і клієнтського пристрою (персонального комп'ютера, ноутбука, планшетного комп'ютера, смартфона тощо).

В основу хмарних обчислень покладено принцип самообслуговування, відповідно до якого одержувач дістає доступ до необхідних йому обчислювальних ресурсів і програмних додатків, що містяться в хмарі, через спеціальний портал. При цьому забезпечується динамічна розширюваність, тобто еластичність.

Визначення 2.1.6. Під еластичністю послуг розуміється властивість інфокомунікаційної системи надавати їх у розширеному або звуженому обсязі за вимогою кінцевих клієнтів сервісу в будь-який момент часу, що дозволяє їм одержувати сервіси з високим рівнем доступності.

В еластичних системах виділення необхідних ресурсів повинно здійснюватися автоматично, тобто без участі технічного персоналу провайдера хмари.

Основні класи послуг, що надаються на основі хмарної платформи, – це доступ до стандартних програмних додатків (Software as a service, SaaS) та стандартних платформ розроблення програмних продуктів (Platform as a service, PaaS), доступ до використання хмарної інфраструктури для розміщення додатків споживача (Infrastruc-ture as a Service, IaaS) (рис. 2.2).

Рисунок 2.2 – Класи послуг хмарної платформи

Хмара, в якій розміщуються доступні споживачам до-

датки і сервіси, розгортається в центрах оброблення даних республіканського і регіонального масштабів. Ці хмари можуть бути публічними, тобто загальнодоступними, або приватними – доступними для обмеженого кола користувачів. У відповідності до визначенняя NIST існують три реалізації хмарної моделі: приватна хмара; публічна хмара; змішана хмара [34] (рис. 2.3).

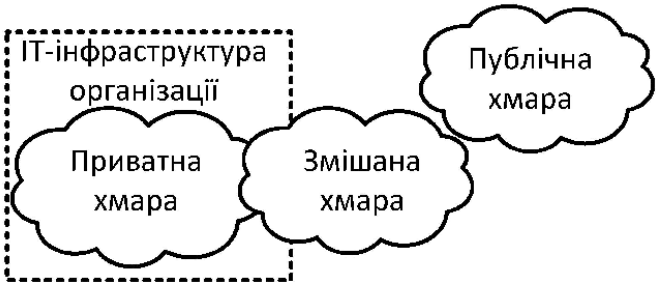


Рисунок 2.3 – Категорії хмар

Визначення 2.1.7. Приватна хмара (Private Cloud) – це спосіб організації IT-інфраструктури в інтересах однієї організації та експлуатується даною організацією.

Організація може керувати приватною хмарою самостійно чи доручити цю задачу сторонньому провайдеру. Інфраструктура може розміщуватися або в приміщеннях замовника, або у зовнішнього провайдера, або частково у замовника і частково у провайдера. Приватна хмара забезпечує більш високий рівень доступності й безпеки для користувачів, повний контроль з боку власника ресурсів, спеціалізацію на певному сегменті інформаційних послуг.

Визначення 2.1.8. Публічна хмара (Private Cloud) – це IT-інфраструктура, яка використовується одночасно численними організаціями і користувачами.

Користувачі публічних хмар не мають можливості керувати і обслуговувати цю хмару, вся відповідальність з даного питання покладена на власника цієї хмари. IT-операції і функції надаються як послуга через мережу Інтернет.

Визначення 2.1.9. Змішана хмара (Hybrid Cloud) – це IT-інфраструктура, яка використовує кращі якості публічної та приватної хмари.

Організація розподілених обчислювальних середовищ дозволяє економити фінансові засоби на боці провайдера хмарних послуг завдяки масштабам виробництва, а також на боці клієнтів у відповідності до принципу використання ресурсів на вимогу. На боці провайдера розв'язуються багато в чому подібні до кластерних систем технічні задачі.

Для реалізації ідеї хмарних обчислень поєднано використання таких давно відомих технологій і понять, як віртуалізація, розподілені обчислення, мережеві технології, сервіс-орієнтована архітектура (Service-oriented Architecture), широкосмугові лінії зв'язку, відкрите і вільне програмне забезпечення, мережеві додатки і багато іншого, пов'язаного з поняттям Web 2.0 [35].

Віртуалізація ресурсів, покладена в основу парадигми хмарних обчислень, дозволяє як розподіляти певний єдиний фізичний ресурс (сервер, операційну систему, додаток або систему зберігання даних) на множину логічних, так і навпаки – множину фізичних ресурсів інтегрувати в один

логічний. Серед основних підходів до віртуалізації варто виділити такі [36]:

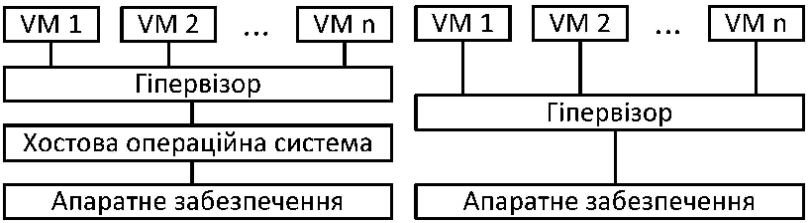
1) віртуалізація різноманітних додатків, які в межах однієї операційної системи працюють в ізольованому середовищі (у різних віртуальних контейнерах), одержуючи віртуальний адресний простір у пам'яті, віртуальний реєстр, віртуальну файлову систему і т. д., що дозволяє уникнути конфліктів і проблем несумісності;

2) віртуалізація платформ, коли кілька операційних систем працюють одночасно в межах одного фізичного комп'ютера, кожна операційна система (гостьова операційна система) працює в ізольованому середовищі на окремому віртуальному комп'ютері – віртуальній машині (Virtual Machine, VM).

Перший підхід передбачає спеціальну підготовку додатків – установлення в спеціальний віртуальний контейнер, його доставлення і запуск у цільовій операційній системі чи на віддаленому сервері. Така віртуалізація додатків найчастіше застосовується для зниження адміністративних зусиль, спрямованих на усунення конфліктів між додатками або їх різними версіями, встановленими на одному комп'ютері.

Другий підхід найчастіше використовуєть для консолідації серверів у центрах оброблення даних. Консолідація забезпечує більш ефективне використання наявного «заліза», скорочення часу відновлення серверів після відмови, а також продовження використання успадкованих додатків у «рідній» для них операційній системі. Реалізація цього підходу, як правило, здійснюється за допомогою програми-

монітора віртуальних машин або гіпервізора, що може працювати поверх хостової операційної системи (рис. 2.4 *а*) або безпосередньо поверх апаратного забезпечення сервера (рис. 2.4 *б*).



а *б*

Рисунок 2.4 – Віртуалізація за допомогою гіпервізора: *а* – поверх хостової операційної системи; *б* – поверх апаратного забезпечення сервера

У випадку гіпервізора, що працює під керуванням хостової операційної системи, на ній повинні бути встановлені драйвери фізичних пристроїв. Для гіпервізора, що працює безпосередньо з апаратним забезпеченням хостової машини, можливі такі варіанти:

1) драйвери пристроїв встановлюються в гіпервізорі («товстий» гіпервізор);

2) драйвери пристроїв установлюються в першій віртуальній машині («мікроядерний» гіпервізор);

3) драйвери пристроїв установлюються в кожній із операційних систем («тонкий» гіпервізор).

Працюючи поверх «голого заліза», товстий гіпервізор також емулює для всіх віртуальних машин однаковий чіп-

сет, а всі виклики віртуальних машин до цього чіпсету транслює в звернення до реального «заліза». «Товстий» гіпервізор повинен містити набір драйверів для всього можливого «заліза», «на всі випадки життя» і що важливо, вони написані спеціально під нього. Мікроядерна архітектура дозволяє використовувати стандартні драйвери, які написані для батьківської операційної системи, встановленої у розділі першої віртуальної машини. Іншим віртуальним машинам доступ до цих драйверів надається у вигляді певних «синтетичних» пристроїв. Усі віртуальні машини за допомогою драйверів синтетичних пристроїв перенаправляють виклики до своїх віртуальних пристроїв через спеціальну високошвидкісну шину віртуальних машин (VM Bus) драйверам у першій віртуальній машині. Для цього драйвери синтетичних пристроїв повинні бути встановлені в операційній системі кожної віртуальної машини.

У загальному випадку як GRID, так і хмарні системи були створені для забезпечення доступу користувачів до певних комп'ютерних ресурсів через мережу. Сервіс-орієнтована архітектура як основа розвитку цих парадигм покращує масштабованість за рахунок зменшення зв'язаності мережевої служби та клієнта. При цьому служба являє собою самодостатню реалізацію певних функцій з чітко визначеним інтерфейсом, який встановлює шаблони для обміну повідомленнями, що використовуються під час взаємодії з функціями. У такому випадку сервіс-орієнтована архітектура розглядається як деяка сукупність служб (сервісів). Рисунок. 2.5 ілюструє простий цикл взаємодії серві-

су з клієнтом, який починається з того, що цей сервіс сповіщає про своє існування і властивості за допомогою сервісу реєстрації (1), властивості та способи взаємодії з яким повинні бути наперед відомі клієнтам.

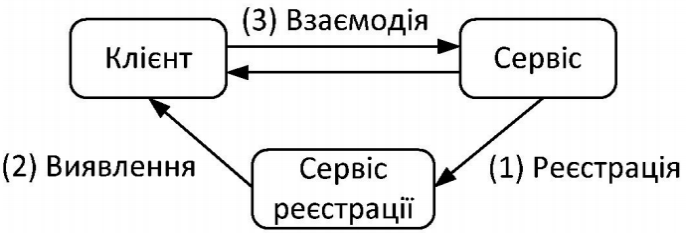


Рисунок 2.5 – Спрощена схема взаємодії сервісів у сервіс-орієнтованому середовищі

Потенційний клієнт, який може бути іншим сервісом чи людиною, робить запит до сервісу реєстрації (2), щоб знайти сервіс, що задовольняє його потреби. Реєстраційний сервіс повертає (можливо пустий) список відповідних сервісів; клієнт обирає один із них і передає йому запит, використовуючи будь-який протокол, що взаємно розпізнається (3). У відповідь сервіс передає або результат потрібної операції, або повідомлення про помилку.

Особливість сервісно-орієнтованої архітектури полягає в мінімальних знаннях взаємодіючих компонентів один про одного, оскільки вони відшукують необхідну інформацію безпосередньо перед взаємодією. Наприклад, дізнавшись про існування сервісу, клієнт може з'ясувати його можливості, умови надання послуг, його місце знаходжен-

ня, його інтерфейси та підтримувані протоколи. Як тільки ці відомості одержані, клієнт може відразу ж звернутися до сервісу, використовуючи будь-який взаємно прийнятний протокол.

У таблиці 2.1.1 наведено основні переваги хмарних технологій та відповідні їх недоліки.

Таблиця 2.1.1 – Порівняльна характеристика хмар

За	Проти
Багатофункціональність	Велика кількість та різнорідність сервісів, складних у використанні
Уніфікація віртуальних ресурсів	Різнорідність фізичних ресурсів у хмарі
Висока масштабованість	Некотрольованість ресурсів
Технологічна надійсність	Аспекти безпеки даних
Динамічне виділення ресурсів	Нестаціонарність обчислювального середовища
Оплата «за фактом»	Правовий статус композитних додатків

Технологія хмарних обчислень так само як і GRID-обчислення використовує розподілені ресурси для досягнення цілей. Однак хмарні обчислення мають перевагу за рахунок використання технологій віртуалізації на різних рівнях (апаратних та програмних платформ) для реалізації сумісного використання і динамічного надання ресурсів. Крім того, у хмарних системах користувачі самі визнача-

ють характер розв'язуваних задач. Порівняльна характеристика хмар та GRID наведена у таблиці 2.1.2.

Таблиця 2.1.2 – Порівняння хмар та GRID-систем

	GRID	Хмара
Організація	Розподілені обчислення	Розподілені обчислення
Основна перевага	Розв'язання складних обчислювальних задач	Забезпечення масштабованого стандарту середовища для мережевих програм, орієнтованих на розроблення та обслуговування
Сервіси	Короткоіснуючі порційні процеси (запуск завдань на виконання)	Довгоіснуючі сервіси, побудовані на апаратній віртуалізації
Складність	Складно	Легко
Цільова група	Наукове співтовариство	Бізнес

Таким чином, застосування розподілених систем зумовлено рядом переваг: віддалений доступ до ресурсів, можливості розпаралелювання обчислень, висока масштабованість, еластичність виділених ресурсів, сумісна робота користувачів зі спільними ресурсами та інше. При цьому сучасні розподілені обчислювальні системи є різнорідни-

ми, мультиархітектурними, динамічно масштабованими системами оброблення інформації, що ускладнює організацію їх ефективного функціонування.

2.2. Методи планування завдань та ресурсів

Ефективне керування IT-інфраструктурою систем розподілених обчислень полягає у врахуванні та узгодженні інтересів користувача та власника ресурсів. Важливу роль тут відіграє планувальник обчислювальних завдань, який приймає рішення про надання обчислювальному завданню доступних ресурсів на основі метаданих вхідного завдання, прийнятих політик обслуговування та інформації про стан ресурсів.

Визначення 2.2.1. Обчислювальна задача – потік інструкцій для процесора з єдиними адресним простором, значеннями регістрів процесора, стеком, відкритими файлами, глобальними змінними тощо.

У загальному випадку на вхід системи планування надходять завдання, що складаються з набору задач (рис. 2.6).

Незалежні задачі виконуються паралельно, а за наявності залежностей – послідовно. Залежності визначають порядок виконання робіт і потоки даних між задачами. У випадку незалежних задач відмінності між плануванням завдань та задач відсутні. Планувальник, крім виділення чи резервування ресурсів, на вузлах оброблення встановлює порядок, в якому задачі із вхідної черги будуть виконуватися на цих ресурсах.

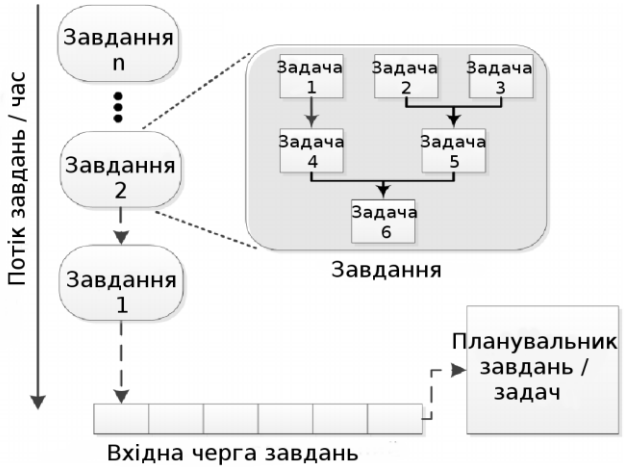


Рисунок 2.6 – Схема надходження завдань
на вхід планувальника

Для розміщення обчислювальних завдань обираються прийнятні за ресурсом, часом та ціною слоти (рис. 2.7).

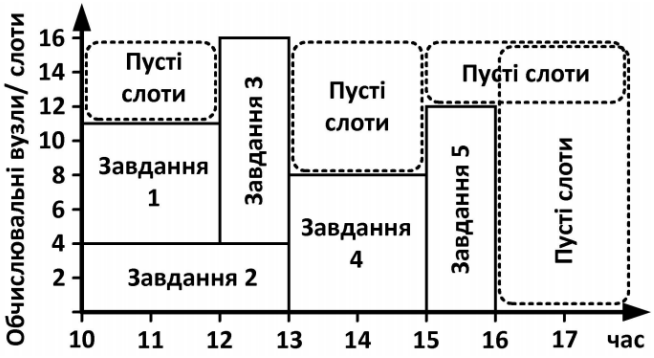


Рисунок 2.7 – Карта розподілу обчислювальних
ресурсів для вхідних завдань

Для запуску будь-якого багатопроцесорного завдання здійснюється узгоджене виділення потрібних для її виконання слотів. У випадку однорідних обчислювальних вузлів сукупність слотів для виконання завдання являє собою прямокутне «вікно», а для процесорів із різною продуктивністю це вікно характеризується нерівним правим краєм (рис. 2.8).

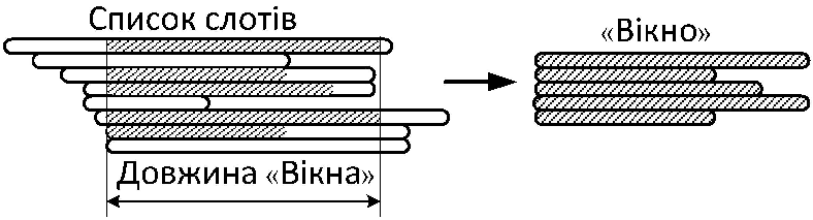


Рисунок 2.8 – «Вікно» неоднорідних за продуктивністю слотів для багатопроцесорного завдання

У загальному випадку ресурсами обчислювального середовища вважаються сукупність програмних та апаратних засобів для виконання обчислювальних задач. Прикладами ресурсів є процесор, середовище передавання даних, прикладне програмне забезпечення, система зберігання даних і тому подібне.

Процесори на обчислювальних вузлах розподіленого обчислювального середовища часто керують окремими ресурсами, які розподіляються з декількома іншими процесорами або пов’язані тільки з цими процесорами, тому доцільно розглядати узагальнене поняття віртуального вузла (рис. 2.9).

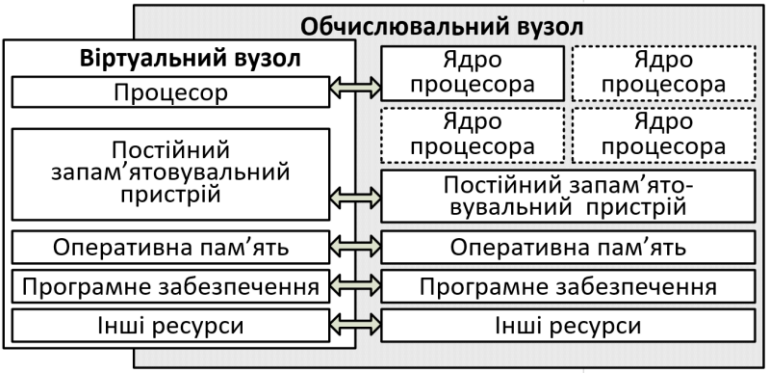


Рисунок 2.9 – Схема відношення ресурсів віртуального та фізичного вузлів розподіленої обчислювальної системи

Визначення 2.2.2. Віртуальний вузол – один обчислювальний елемент (ядро процесора) у зв'язці із зіставленими йому ресурсами.

Якщо декілька процесорів використовують одні й ті самі ресурси, що характеризуються обсягом (оперативна пам'ять, постійна пам'ять і так далі), то часто вважають, що обсяг цих ресурсів поділяється рівномірно між усіма віртуальними вузлами, які відповідають цим процесорам. На практиці можлива конкуренція віртуальних вузлів за ресурси фізичного вузла, яку можна спрогнозувати і запобігти цій ситуації.

Алгоритми планування завдань в розподіленому обчислювальному середовищі можна поділити на статичні та динамічні [37].

Під час статичного планування розрахунок вартісного

оцінювання обчислень здійснюється до початку виконання завдання, коли інформація щодо всіх ресурсів розподіленого обчислювального середовища і всіх завдань вже доступна. Одна з головних переваг статичної моделі – це простота реалізації планувальника. Однак вартісне оцінювання, засноване на статичній інформації, є не адаптивним до ситуацій, коли час відгуку стає більш тривалим, ніж очікувалося через високе завантаження системи, або коли один із обчислювальних вузлів виходить з ладу, або стає ізольованим внаслідок мережевих відмов. Для вирішення проблеми використовують допоміжні механізми, наприклад, механізм перепланування.

Динамічне планування зазвичай застосовується, коли важко оцінити обчислювальну вартість додатків, що надходять на виконання динамічно в режимі (online). Динамічне планування завдань містить в собі два важливих компоненти: оцінювання стану системи і прийняття рішення про зв'язування завдання з черги з обраним ресурсом. Для збереження оптимального стану обчислювальної системи використовується балансування завантаження всіх її ресурсів. Перевага динамічного балансування завантаження над статичним плануванням полягає в тому, що система не зобов'язана знати про поведінку додатка до його запуску. Особливо цей підхід корисний у системі, де основною метою є максимізація утилізації ресурсу, а не мінімізація часу виконання окремих завдань.

У динамічних сценаріях планування відповідальність за прийняття глобальних рішень планування ресурсів може лежати на одному централізованому планувальнику або

декількох розподілених планувальниках. Централізована стратегія має перевагу, яка полягає в простоті реалізації, але вона погано масштабується, невідмовостійка і часто стає вузьким місцем для продуктивності системи. У цьому випадку, коли вся інформація щодо стану ресурсів і завдань відома, оптимальне прив'язування завдань до ресурсів може бути зроблено на підставі окремої цільової функції, наприклад, мінімізація часу виконання завдань чи максимізація утилізації ресурсів.

Розподілені алгоритми планування можна поділити на зв'язані або незв'язані, залежно від того, як працюють вузли, що використовуються під час планування завдань, спільно або незалежно (несумісно). У окремому випадку локальні планувальники діють як автономні сутності і приймають рішення з урахуванням їх власних цільових функцій. У загальному випадку кожен планувальник в розподіленому обчислювальному середовищі несе відповідальність за виконання його власної частини завдання планування, але при цьому всі планувальники працюють з однією спільною метою в масштабі всієї системи.

Під час організації розподілених обчислень найпростішими є спискові алгоритми планування, які дозволяють обрати із списку вхідних завдань одне з них для призначення на вільні віртуальні вузли.

Спискові алгоритми планування впорядковують вхідні завдання в чергу за деяким їх параметром, наприклад, за кількістю необхідних для виконання процесорів або часом її виконання [38], а потім переглядають її, починаючи з початку, і призначають завдання на вільні обчислювальні

ресурси. Найбільш відомими списковими алгоритмами планування є такі: першим прийшов, першим обслуговується (First Come First Served, FCFS), найкоротше перше завдання (Shortest Job First, SJF), найдовше перше завдання (Longest Job First, LJF), випадкове завдання обслуговується першим (Random Job First, RJF) і так далі.

Найбільш простою в реалізації є дисципліна FCFS, відповідно до якої задачі обслуговуються «у порядку черги», тобто в порядку їх появи. Ті задачі, які були заблоковані в процесі роботи (потрапили в будь-який із станів очікування, наприклад, через операції введення/виведення), після переходу в стан готовності стають в чергу готовності перед тими задачами, які ще не виконувалися. Іншими словами, утворюється дві черги: одна черга утворюється з нових задач, а друга черга – з тих, що раніше виконувалися, але потрапили в стан очікування. Такий підхід дозволяє реалізувати стратегію обслуговування «по можливості закінчувати обчислення в порядку їх появи». Ця дисципліна обслуговування не потребує зовнішнього втручання в хід обчислень, при ній не відбувається перерозподілу процесорного часу. До переваг цієї дисципліни, перш за все, можна віднести простоту реалізації та малі витрати системних ресурсів на формування черги задач, а також немає необхідності в інформації про тривалість виконання завдань. Часто на практиці FCFS доповнюють модифікаціями інших алгоритмів. Однак поряд із простотою використання дисципліна FCFS призводить до того, що при збільшенні завантаження обчислювальної системи зростає і середній час очікування обслуговування, причому короткі

завдання (потребують невеликих витрат машинного часу) змушені очікувати стільки ж, скільки і трудомісткі завдання.

Дисципліна обслуговування SJF, що означає: наступним буде виконуватися найкоротше завдання, вимагає, щоб для кожного завдання було відоме оцінювання потребах машинного часу. Необхідність повідомляти планувальнику характеристики завдань, в яких описувалися б потреби в ресурсах обчислювальної системи, привела до того що були розроблені відповідні мовні засоби. Зокрема, мова JCL (Job Control Language) була однією з найбільш відомих мовних засобів. Користувачі змушені були вказувати передбачуваний час виконання, і для того щоб вони не зловживали можливістю зазначити свідомо менший час виконання (з метою одержати результати раніше за інших), ввели підрахунок реальних потреб. Диспетчер завдань порівнював замовлений час і час виконання і у разі перевищення зазначеного оцінювання у цьому ресурсі ставив дане завдання не на початок, а на кінець черги. Дисципліна обслуговування SJF припускає, що є тільки одна черга завдань, готових до виконання нею. І завдання, що в процесі свого виконання були тимчасово заблоковані (наприклад, очікували завершення операцій введення/виведення), знову потрапляють на кінець черги готових до виконання нарівні із новими вхідними завданнями. Це призводить до того, що завдання, яким потрібно дуже небагато часу для свого завершення, вимушені очікувати у черзі нарівні з тривалими роботами, що не завжди добре. Для усунення цього недоліку була запропонована дисципліна, за якою

наступним обслуговується завдання, що вимагає найменшого часу для свого завершення (Shortest Remaining Time, SRT).

Розглянуті дисципліни обслуговування можуть використовуватися для пакетних режимів оброблення, коли користувач не змушений очікувати реакції системи, а просто віддає своє завдання і через кілька годин одержує результати обчислень. Для інтерактивних обчислень бажано насамперед забезпечити прийнятний час реакції системи і рівність в обслуговуванні.

Досить популярний алгоритм кругової диспетчеризації (Round Robin, RR), де віртуальні вузли утворюють віртуальне кільце з маркером, а наступна робота, готова до виконання, призначається на вузол, що має маркер, потім маркер передається наступному вузлу [39]. Дисципліна кругової диспетчеризації найбільше підходить для випадку, коли всі завдання мають однакові права на використання ресурсів центрального процесора. Однак одні завдання завжди потрібно вирішувати в першу чергу, тоді як інші можуть почекати. Це можна реалізувати за рахунок того, що одній задачі диспетчер завдань привласнює один пріоритет, а іншій задачі – інший. Завдання в черзі будуть розміщуватися відповідно до їх пріоритетів.

На цей час для планування обчислень в масштабованих системах активно використовують такі алгоритми, як міграція процесів і завдань, планування групами завдань (Gang-scheduling) і алгоритм зворотного заповнення (Backfilling) [39].

Алгоритм зворотного заповнення вимагає інформації

про тривалість виконання завдань. Мета алгоритму – найбільш щільно заповнити пусті вікна. Для цього серед наявних вікон обирається найбільш широке вікно, тобто з максимальною кількістю процесорів, і наступні завдання, які потрапляють в чергу, будуть призначатися на процесори цього вікна. Якщо нове завдання не поміщається в жодне з доступних вікон, то вона роміщується в кінці черги. Таким чином, роботи поширюються у зворотний бік щодо шкали часу. Алгоритм зворотного заповнення досить часто використовується в системах черг, що здійснюють «справедливий» доступ користувальницьких задач до ресурсів багатопроцесорних систем. Переваги алгоритму зворотного заповнення:

1) дозволяє складати розклад для гетерогенних розподілених обчислювальних систем;

2) уникає зависання низькопріоритетних завдань в чергах, гарантуючи їх запуск;

3) може складати досить щільні за часом розклади;

4) має прийнятні характеристики швидкості роботи.

Алгоритм планування групами завдань (Gang-scheduling) здійснює розподіл ресурсів багатопроцесорної системи між групами завдань. Завдання об'єднані в групи за пріоритетами. Завдання однієї групи поділяють безліч процесорів таким же чином, як і у разі алгоритму Backfill, однак допускається переривання завдань, якщо на багатопроцесорну систему надходить група завдань з високим пріоритетом.

Задача складання розкладу належить до класу NP-повних задач, в яких зростає складність розв’язання при

збільшенні розмірності. Існуючі точні методи розв'язання таких задач у гіршому разі під час пошуку здійснюють перебирання всіх можливих варіантів розподілу завдань за виконавцями, що вимагає великих обчислювальних витрат для випадку високої розмірності. Тому знаходження оптимального розв'язку задачі розподілу за прийнятний час стає важкодосяжним.

Вирішення подібного класу завдань вимагає великих обчислювальних і відповідно часових ресурсів для знаходження оптимального розв'язку задачі розподілу ресурсів, внаслідок чого виграш від використання знайденого розв'язку не покриває величезні витрати на його одержання. На практиці для розв'язку NP-складних задач часто використовують евристичні методи, що не гарантують знаходження оптимального розв'язку, але дозволяють досить швидко одержувати рішення прийнятної якості.

Евристичні методи оптимізації побудовані на використанні різних розумних, здебільшого заснованих на життєвих та природних аналогіях, звичках, правилах, спрямованих на досягнення компромісу між прагненням до найкращого результату і до скорочення часу на перебирання варіантів дій для досягнення цього результату. Незважаючи на недостатню теоретичну обґрунтованість, ці методи дозволяють одержувати прийнятні рішення при порівняно невеликих витратах часу та інших ресурсів. До переваг евристичних методів можна також віднести зручність їх реалізації на електронно-обчислювальних машинах навіть при вирішенні завдань високої розмірності. Недоліки даних методів полягають у складності оцінювання їх факти-

чної ефективності, тобто близькості одержаних рішень до оптимальних. Крім того, для кожного евристичного підходу існують завдання, для яких застосування цього підходу або неможливе, або призводить до відверто поганих результатів. Це вимагає ретельного експериментального дослідження евристичних методів із метою виділення класів завдань, при вирішенні яких ці методи найбільш ефективні.

До найбільш ефективних і популярних евристичних методів відносять так звані метаевристичні – узагальнені стратегії пошуку оптимуму в просторі рішень. Як приклад можна навести алгоритм імітації відпалу (Simulated Anneal), жадібний алгоритм (greedy algorithm), генетичні (Genetic Algorithms) та мурашині алгоритми (Ant colony optimization) [40].

У більшості розглянутих методів передбачається, що ресурси є однорідними, зокрема часто робиться припущення, що всі ресурси мають однакову продуктивність та ціну. У цьому разі задача вибору ресурсів значно спрощується, оскільки існує взаємозамінність ресурсів. Проте на цей час на практиці ресурси все частіше є гетерогенними і відрізняються один від одного архітектурою, продуктивністю, пам’яттю, ціною, пропускною смугою. Вибір придатного ресурсу для завдання користувача стає багатопараметричною задачею, причому окремі параметри є взаємозалежними. Відповідно ускладнюється і задача організації обчислювального процесу з урахуванням як інтересів користувачів, так і власників ресурсу.

Під час вирішення оптимізаційної задачі планування,

як правило, використовуються «системно-центричні» (system-centric) критерії, що не враховують вимоги споживачів до якості обслуговування. Подібні методи планування розміщують завдання у відповідності до системних параметрів, які стосуються завантаження ресурсів або пропускної здатності системи. Планувальники приділяють увагу або мінімізації часу відгуку (response time), сумарного часу очікування (waiting time) і фактичного часу виконання завдання (turnaround time), або максимізації загального завантаження ресурсів (resource utilization). У цій ситуації метою планування виявляється поліпшення становища системи в цілому, але не вимоги, пропоновані окремими споживачами.

Традиційні підходи до керування ресурсами орієнтовані на критерії, що відносяться до всієї системи, такі як використання інфраструктури і/або пропускна здатність. У той самий час потреби хмарних інфраструктур вимагають розвитку таких моделей, які значною мірою відображали б інтереси користувачів. Іншими словами, потрібно створення таких алгоритмів розподілу завдань, які забезпечували б максимум корисності для індивідуального споживача.

Внаслідок принципово егоїстичної поведінки користувачів розподіленого обчислювального середовища їх важко змусити об'єктивно враховувати загальносистемні критерії. Тому паралельно розвивається використання стимулювальних економічних механізмів для організації керування ресурсами. Наприклад, ціна обчислень встановлюється відповідно до сумарної кількості заявок попиту і пропозиції. Часові відхилення від директивних термінів виконання

завдань можуть штрафуватися подорожчанням процесів проведення обчислень в обчислювальній системі. Ринково-вартісні міркування є важливою складовою сучасних досліджень в галузі розподілу обчислювальних ресурсів.

Необхідно зазначити, що застосування відомих алгоритмів організації обчислювального процесу в традиційних розподілених середовищах є малоефективним у хмарних системах у зв'язку з такими факторами:

1) випадкова зміна кількості користувачів, наслідком чого є непередбачуваною динаміка попиту/ пропозиції/ доступності ресурсів, що вимагає динамічної масштабованості ресурсів;

2) зміна кількості користувачів і структури попиту викликає необхідність внесення коректив у плани обчислень;

3) багатофакторність, тобто наявність безлічі різних критеріїв, політик, переваг та обмежень на виконання обчислювальних завдань, що призводить до необхідності балансування між ними;

4) різноманітність вимог і переваг користувачів обчислювальної системи вимагає індивідуального підходу до споживачів сервісів;

5) внаслідок неточності знань про характеристики фізичних вузлів, віртуальних машин та задач відбувається розбалансування навантаження у ході виконання завдань, тобто система ресурсів перестає відповідати загальній стратегії планування за кількістю, якістю обслуговування та продуктивністю.

Таким чином, сучасні тенденції в організації розподілених обчислень вимагають динамічної масштабованості,

динамічного балансування навантаження за умови неповної інформації про характеристики фізичних і віртуальних машин, поведінку та ресурсні вимоги завдань оброблення з урахуванням інтересів користувачів та власників ресурсів.

2.3. Енергозбереження в інфокомунікаційних системах

Останніми роками дуже популярним і важливим став критерій енергоефективності обчислювальних систем, оскільки споживання електроенергії становить основну частину експлуатаційних витрат у розподілених обчислювальних системах. Наприклад, обсяг споживання електроенергії в центрах оброблення даних, що підтримують масові хмарні сервіси, може перевищувати обсяг споживання електроенергії цілих міст.

Існуючі праці з цього питання можна поділити на два основні напрямки [41]:

1) архітектура з низьким енергоспоживанням (статичний підхід);

2) зниження енергоспоживання програмними засобами (динамічний підхід).

Статичний підхід передбачає побудову обчислювальної системи з компонентів, що характеризуються низьким енергоспоживанням. Статичний підхід характеризується дорожнечою, оскільки в його основу покладене застосування нестандартних компонентів та інженерних рішень. Однак останнім часом в цьому напрямі сталися значні зрушення, всі виробники обладнання намагаються знижувати його енергоспоживання, а конкуренція приводить до

зниження вартості таких рішень.

Паралельно зі статичним підходом до зниження енергоспоживання розвивається підхід на підставі використання програмних засобів. Цей підхід об'єднує ряд методів, що дозволяють досягти енергоефективного виконання потоку завдань за рахунок вибіркового вимкнення або зниження продуктивності компонентів обчислювальної системи, коли вони простоюють або завантажені частково. Можна виділити три основні способи програмного підвищення енергоефективності:

1) вимкнення компонентів обчислювальної системи, що простоюють;

2) перерозподіл обчислювальних завдань за часом за умови наявності багатотарифної схеми оплати електроенергії (наприклад, день-ніч);

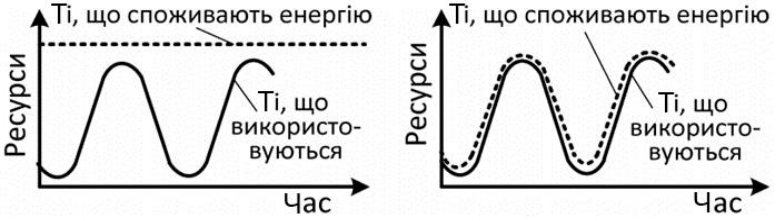
3) програмне керування продуктивністю компонент обчислювальної системи.

У нічний час, коли, як правило, тариф на електроенергію нижчий, варто збільшувати обчислювальне навантаження, а вдень знижувати і вимикати компоненти, що простоюють. Загальна кількість спожитої енергії звичайно не зміниться, проте зменшиться її вартість, що також розглядається як підвищення енергоефективності.

Сучасні процесори і оперативна пам'ять мають можливість динамічно змінювати свою частоту і робочу напругу. Такий механізм має назву DVS (Dynamic Voltage and frequency Scaling). Основний принцип цього механізму полягає в тому, що при зниженні напруги процесора час обчислень збільшується, проте загальна кількість енергії,

витраченої на обчислення, зменшується.

Важливий аспект динамічного керування енергоспоживанням полягає в тому, що зміна стану системи (підключення, зміна продуктивності тощо) має певну вартість, що виражена у додатковій кількості спожитої енергії, затримці або втраті продуктивності, що, коротко кажучи, не гарантує зниження енерговитрат при переведенні системи у сплячий режим за відсутності роботи і назад у міру потреби. Технологія віртуалізації, що покладена в основу концепції хмарних обчислень, забезпечує гнучкість надання ресурсів і економію, пов'язану з мінімізацією обсягу невикористаних ресурсів, що споживають електроенергію (рис. 2.10).



а *б*

Рисунок 2.10 – Режими використання ресурсів хмари: *а* – з простоями; *б* – оптимальне використання

За рахунок віртуалізації можна суттєво зменшити кількість серверів і необхідну площу дата-центру, а також знизити вимоги до потужності та охолодження в ньому. Хоча загальне енергоспоживання центра оброблення даних знижується внаслідок впровадження віртуалізації, проте окре-

мо взятий сервер потребуватиме більшої потужності. Так само при зменшенні загальної кількості серверів у дата-центрі кожна стійка (шасі) потребуватиме більше енергії, ніж раніше (рис. 2.11) [42].

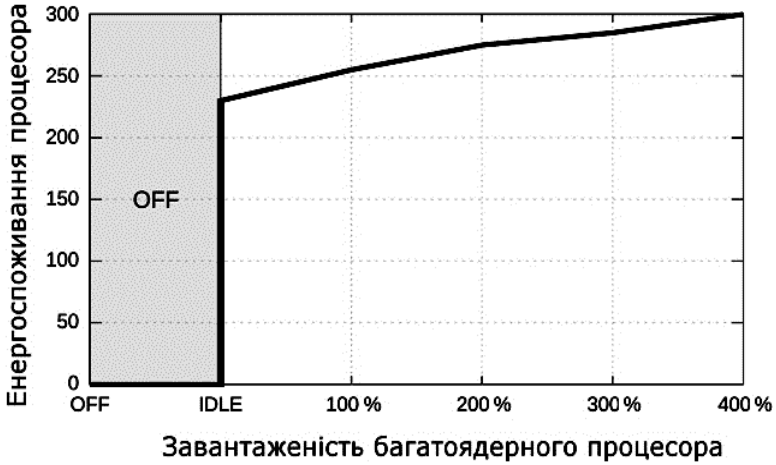


Рисунок 2.11 – Залежність енергоспоживання багатоядерного процесора від його завантаженості при незмінній структурі споживання ресурсів

Як відомо, віртуалізація дозволяє на запит розгортати, переміщувати чи клонувати додатки з однієї платформи на іншу навіть у процесі функціонування (рис. 2.12) [43]. При цьому потреби в енергозбереженні та охолодженні також можуть переміщуватися по площі дата-центру. Однак існуюча інфраструктура часто не в змозі забезпечити ці переміщення, не порушуючи обмеження, зазначені в SLA.

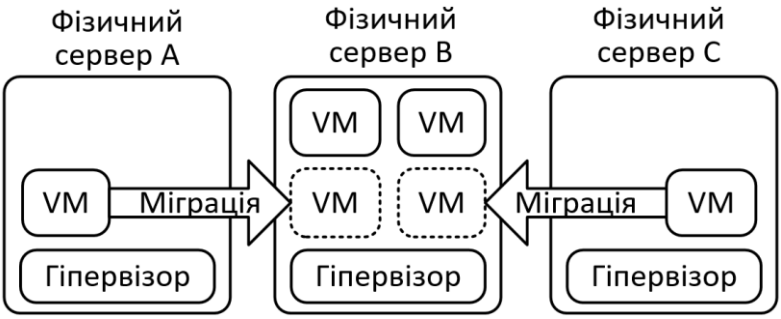


Рисунок 2.12 – Міграція віртуальних машин
на інший фізичний сервер

Площа дата-центру завдяки віртуалізації зменшиться, але його загальна ефективність може залишитися субоптимальною, оскільки центр оброблення даних за рахунок віртуалізації оптимізований тільки частково. Підвищення щільності розміщення віртуальних машин в умовах гетерогенності додатків та конкуренції за спільні ресурси фізичних серверів може порушити умови SLA. Для комплексної оптимізації енерговитрат розподіленого середовища центрів оброблення даних потрібно вирішити ряд важливих завдань :

1) одержання моделі для прогнозування ситуації порушення умов SLA при різних комбінаціях розміщення віртуальних машин на фізичних серверах;

2) визначення моментів часу для переміщення віртуальної машини з перевантаженого сервера з метою уникнення зниження продуктивності та для переміщення віртуальної машини з недовантаженого сервера з метою підвищення ефективності використання ресурсів і зведення до

мінімуму споживання електроенергії;

3) визначення оптимальної підмножини віртуальних машин, які підлягатимуть перерозподілу на інші сервери, з точки зору системних критеріїв ефективності та SLA;

4) визначення оптимального місця розміщення нової віртуальної машини або машин, обраних для міграції з точки зору системних критеріїв ефективності та SLA;

5) визначити, коли і які фізичні сервери будуть вимкнені для енергозбереження і навпаки вчасно увімкнені для уникнення порушень SLA.

При розв'язанні цих задач потрібно враховувати, що міграція віртуальної машини викликає комунікаційні та інші накладні витрати. Наприклад, міграція між серверами однієї стійки (шасі) може призводити до 100 мс простою, а міграція між центрами оброблення даних може займати більше однієї хвилини. Для завдань оброблення даних, що направляються на віртуальні машини, потрібно додержуватися правила: дані повинні бути розміщені ближче до процесу їх оброблення. Тому алгоритми динамічного планування завдань та керування ресурсами повинні враховувати рівень локальності даних (Data Locality). Прийнято виділяти чотири рівні локальності даних [44]: локальність на вузлі (Node Locality), локальність на сервері (Server Locality), локальність на серверних стійках (Rack Locality) та локальність поза серверною стійкою (Off Rack Locality). Чим більша локальність під час міграції віртуальних машин або завантаженні даних на оброблення, тим менше простоїв у системі й відповідно менше нераціональних витрат електроенергії.

Таким чином, основним способом енергозбереження в сучасних розподілених обчислювальних середовищах є використання технології віртуалізації та алгоритмів перерозподілу навантаження між фізичними серверами для мінімізації їх простою. При цьому існує ризик порушення умов SLA, тому для реалізації цього способу необхідно додактково використовувати методи прогнозування функціонального стану фізичних серверів та віртуальних машин.

2.4. Прогнозування функціонального стану інфокомунікаційної системи

Розроблення ефективних алгоритмів розподілу ресурсів в обчислювальному середовищі ускладнено відсутністю інформації про ресурсні вимоги та поведінку конкретного завдання на навантажених гетерогенних вузлах. Апріорна невизначеність функціонального стану вузла при виконанні завдання і неспроможність точного оцінювання часу його виконання можуть призвести до виділення надлишкових ресурсів, які будуть простоювати, знижуючи завантаженість обчислювального середовища, або виділення недостатнього обсягу ресурсів, що призводить до накладних витрат, пов'язаних із процесом введення до експлуатації додаткових ресурсів або міграцією задач на інший вузол.

Для одержання або уточнення знань про поведінку додатків на вузлах оброблення, як правило, використовують дані трасування їх роботи. У загальному випадку ці дані

можуть бути наведені у вигляді часових рядів споживання ресурсів, де k-та точка ряду містить позначки математичного сподівання $\mu[k]$ та середньоквадратичного відхилення $\sigma[k]$ (рис. 2.13) [45].

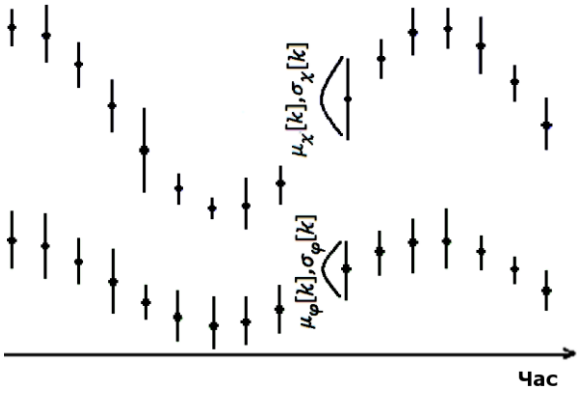


Рисунок 2.13 – Два часові ряди споживання обчислювального ресурсу із заданим у кожній точці середньоквадратичним відхиленням

Часові ряди, як правило, зашумлені під впливом тимчасових невідомих станів пристроїв. Тому для подальшого аналізу рядів здійснюється їх попередня фільтрація, наприклад, за допомогою фільтра низьких частот Чебишева, та нормалізація, в результаті якої значення ряду розміщені в діапазоні від 0 до 1.

З метою передбачення поведінки нового чи невідомого додатка в розподіленому обчислювальному середовищі система керування розподілом ресурсів може здійснити порівняння часових рядів споживання ресурсів відповід-

ними задачами додатка з наперед відомими еталонами, які зберігаються у базі профілів. У базі профілів містяться еталонні зразки рядів споживання ресурсів для задач, що часто виконуються в середовищі. При цьому для кожного з профілів задано оптимізовані значення конфігураційних параметрів. Збіг поведінки нового додатка з відомим шаблоном споживання ресурсів дозволяє реалізувати автоматичне підстроювання розподіленого обчислювального середовища до вимог обчислювальних завдань, що покращує результати динамічного розподілу ресурсів і зменшує ймовірність порушення SLA.

Для формування еталонних зразків здійснюється профілювання відомих додатків із різними вхідними даними і настроюваннями (рис. 2.14).

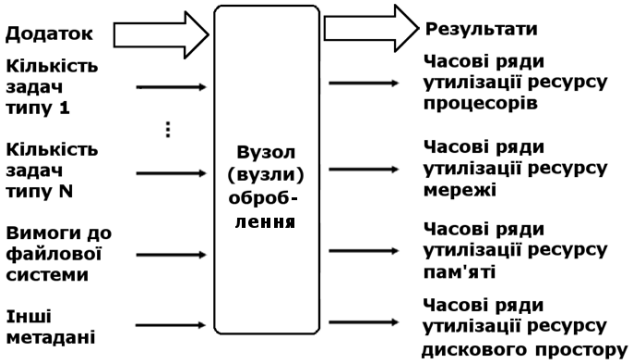


Рисунок 2.14 – Модель для профілювання додатків розподіленого обчислювального середовища

При цьому відібрані завдання виконуються на різних

обчислювальних вузлах декілька десятків разів. Утиліти моніторингу виконують трасування виконання завдань та обчислення статистичних характеристик споживання ресурсів в різні моменти часу від початку запуску.

Для обчислення подібності вхідного часового ряду з еталонним зразком, які мають різну довжину, часто використовують алгоритм динамічної трансформації шкали часу (Dynamic Time Warping, DTW). У цьому алгоритмі нерівності довжин долають повторним формуванням підвибірки з одного ряду перед порівнянням з іншим рядом. Для визначення подібності між двома часовими рядами споживання обчислювального ресурсу $X = [x_1, x_2, ..., x_N]$ та $Y = [y_1, y_2, ..., y_M]$ за умови $N \geq M$ DTW-алгоритм використовує такі рекурсивні формули:

$$D(i, j) = \min \begin{cases} D(i, j-1), \\ D(i-1, j) + d(x_i, y_j), \\ D(i-1, j-1), \end{cases} \qquad (2.4.1)$$

$$d(x_i, y_j) = \left\| CPU(x_i) - CPU(y_j) \right\|, \qquad (2.4.2)$$

де $d(x_i, y_j)$ – відстань Евкліда між відповідними точками в обох часових рядах;

$CPU(x_i)$ – значення споживання обчислювального ресурсу в точці x_i ряду X.

Результатом обчислень за формулою (2.4.1) є матриця

$D(X,Y)$, кожний елемент якої $D(i,j)$ відображає мінімальну відстань між часовими рядами $[X(x_1),Y(y_1)]$ і $[X(x_i),Y(y_j)]$, тобто матриця $D[N,M]$ характеризує подібність рядів X та Y. За умови $N \geq M$ для порівняння використовується ряд Y', який формується з ряду Y і береться за довжиною, що дорівнює довжині ряду X. Тобто Y' вирівнюється до ряду X і формується шляхом ітераційного повторення вибірки елементів із ряду Y, що грунтується на $D(X,Y)$.

Після пошуку мінімальної відстані між двома рядами за алгоритмом DTW буде сформовано новий ряд Y'. Остаточне вимірювання подібності між рядами Y' та X може бути здійснене шляхом обчислення відповідного кореляційного коефіцієнта [45]:

$$CORR(X,Y') = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu_X)(Y'_i - \mu_{Y'}). \quad (2.4.3)$$

Кореляційний коефіцієнт показує, на скільки два часові ряди корелюють чи подібні. Якщо $CORR(X,Y') = 0$, то схожість між рядами відсутня, а у разі $CORR(X,Y') = 1$ можна зробити висновок про точний збіг. Чим більше значення кореляційного коефіцієнта, тим більша подібність рядів. Для практичних задач, як правило, обирається емпіричний поріг $CORR(X,Y') \geq 0,9$, перевищення якого береться за збіг рядів.

У будь-якій інформаційній системі надійність є однією

з найважливіших характеристик. У розподіленій обчислювальній системі внаслідок реалізації масштабованої та гетерогенної архітектури збільшується її складність і ймовірність перевантажень та відмов в обслуговуванні. Найбільш загальною платформою для реалізації великомасштабних обчислювальних систем є кластери, зокрема віртуальні. З метою підвищення відмовостійкості обчислювальної системи та доступності її сервісів використовуються алгоритми прогнозування перевантажень та відмов при запуску завдань користувача. Керуючі вузли кластерних систем повинні мати систему контролю доступу (Admission Control) з метою визначення сумісності завдань користувача з поточними можливостями гетерогенного обчислювального середовища. Система контролю доступу відхиляє рішення щодо призначення задач користувача конкретному вузлу, якщо існує висока ймовірність порушення конфігураційних обмежень, заданих в обчислювальному середовищі.

Конфігураційні обмеження обов’язково містять задані адміністратором ресурсні обмеження вузлів та задані чи обрані користувачем вимоги до умов обслуговування. Ресурсні обмеження характеризують ємність мережевих, процесорних і дискових ресурсів вузла та граничний обсяг частки ресурсів, яка може виділятися задачам користувача залежно від його ролі (адміністратор, привілейований користувач, авторизований користувач, гість). Користувач, у свою чергу, визначає бюджет, призначений для оплати тарифікованих ресурсів, граничний час оброблення задач (дедлайн) та заданий користувачем пріоритет «важли-

вості».

У праці [46] запропоновано евристичний алгоритм для контролю доступу до ресурсів кластера на основі моделі косинусної подібності. Суть алгоритму полягає у порівнянні вектора обсягу доступних на вузлі ресурсів із вектором ресурсів, потрібних задачі для її виконання. При цьому як міра подібності використовується косинус кута між векторами.

Для з'ясування загального обсягу зайнятих ресурсів у обчислювальному середовищі складається вектор задач, що виконуються на i-му вузлі,

$$T_{compound}\ (i) = T_1 + T_2 + ... + T_k + ... + T_n,$$

де T_k – вектор ресурсних вимог кожної із задач.

Для кожної задачі, що перебуває в черзі для виконання, теж складається вектор її ресурсних вимог:

$$T_q = E_{cpu} e_1 + E_{mem} e_2 + E_{disk} e_3 + E_{nw} e_4,$$

де e_1, e_2, e_3, e_4 – базисні вектори;

E_{cpu} – середній обсяг процесорного ресурсу;

E_{mem} – середній обсяг оперативної пам'яті;

E_{disk} – середній обсяг дискового простору;

E_{nw} – середній обсяг мережевого ресурсу.

Вектор доступних ресурсів обчислюється як різниця

між загальним обсягом ресурсів R_{total} та компонентами вектора $T_{compound}$, тобто

$$T_{availability} = R_{total} - T_{compound}.$$

У разі призначення q-ї задачі на один із вузлів оброблення обсяг невикористаних ресурсів можна обчислити за формулою

$$T_{unused} = T_{availability} - T_q.$$

Завдання, що виконуються на вузлах оброблення, не повинні конфліктувати та конкурувати за спільні ресурси, тому для перевірки сумісності обчислюють косинусну міру подібності вектора вільних ресурсів до вектора ресурсних вимог за формулою

$$similarity = \frac{T_q \cdot T_{availability}(i)}{\left\|T_q\right\| \cdot \left\|T_{availability}(i)\right\|} = cos\ (\overrightarrow{T_q} \wedge \overrightarrow{T_{availability}(i)})$$

Косинусна міра відповідає косинусу кута між вектором T_q та $T_{availability}(i)$ а її значення лежить в діапазоні [-1; 1]. Чим більше значення міри подібності, тим більша сумісність задачі з доступними ресурсами. При цьому більші значення косинусної міри відповідають меншим значенням невикористаних ресурсів. Тому при прийнятті рі-

шення щодо розміщення задачі на конкретному вузлі оброблення міру подібності потрібно об'єднати з оціненням обсягу невикористаних ресурсів T_{unused} у вигляді нерівності

$$C_{heu} \le \alpha \cdot similarity + (1-\alpha) \cdot \left| T_{unused} \right|,$$

де α, C_{heu} – параметри конфігурації, що задаються адміністратором розподіленого обчислювального середовища.

У разі гетерогенних ресурсів та обчислювальних задач визначення порогових значень параметрів, що характеризують місткість вузлів оброблення, є складним завданням, яке адміністратор не може вирішити без використання інструментів статистичного аналізу або інтелектуального аналізу даних (Data Mining). Тому одним із найефективніших підходів до прогнозування стану перевищення ресурсних обмежень перед призначенням задачі на обчислювальний вузол є побудова вирішальних правил на основі машинного навчання за даними передісторії, де роль вчителя виконує прапорець (не)перевищення ресурсних обмежень [46, 47]. При цьому вектор ознак, що розпізнається здатним навчатися класифікатором, складається з ресурсних ознак задачі, яка знаходиться в черзі, та ресурсних ознак вузла, на який передбачається переміщення задачі (рис. 2.15).

Ресурсні ознаки задачі формуються на основі унікального ідентифікатора задачі та передісторії її оброблен-

ня на заданому вузлі. До ресурсних ознак задачі відносять середнє вибіркове значення та вибіркову дисперсію обсягу використаних ресурсів процесора, оперативної пам'яті, пропускної здатності мережі та пропускної здатності для операцій введення-виведення під час роботи з жорстким диском. Додатковими ознаками задачі можуть бути метадані вхідних блоків даних, які корелюють з ресурсними вимогами.

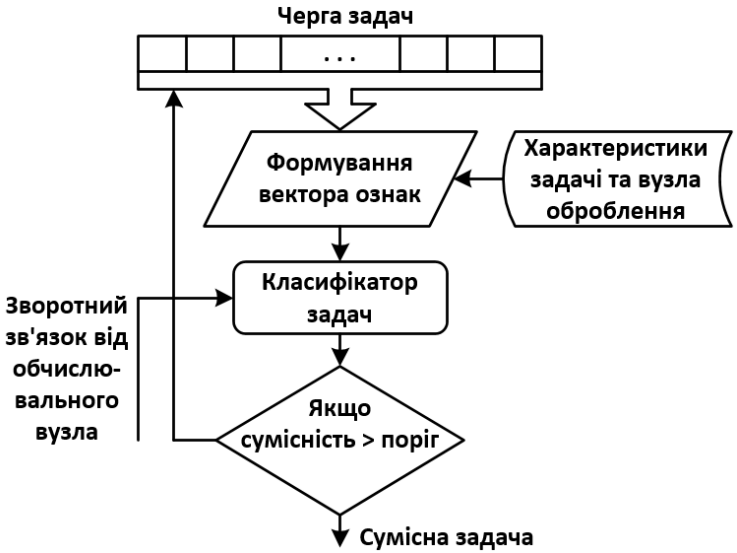


Рисунок 2.15 – Структурна схема алгоритму вибору задач для їх розміщення на вільному вузлі оброблення

Ресурсні ознаки вузла поділяють на статичні, які не змінюються впродовж функціонування обчислювального вузла, та динамічні, що змінюються в процесі його функ-

ціонування. До статичних характеристик вузла відносять кількість процесорів, частоту процесорів, архітектуру процесорів, повний обсяг фізичної оперативної та віртуальної (файл підкачування) пам'яті, обсяг кеш-пам'яті, обсяг пам'яті жорстких дисків, тип та версію операційної системи. До динамічних характеристик відносять кількість використаних на вузлі процесорів, рівень завантаження процесорів, обсяг вільної оперативної пам'яті, інтенсивність операцій введення/виведення, пропускну здатність мережі для приймання та передавання інформації, обсяг вільної віртуальної пам'яті, залишок дискового простору, споживану потужність та ін.

Крім перелічених ознак розпізнавання, з метою врахування локальності даних у розподіленій обчислювальній системі як додаткову інформативну ознаку потрібно розглядати мережеву відстань між розміщенням задачі та розміщенням блоків даних.

Якщо планувальнику не вдається знайти для задачі вузол, який буде задовольняти ресурсні обмеження, то задача буде очікувати на появу необхідних ресурсів або буде повернута користувачу для корекції вимог до умов обслуговування.

Віртуалізація є однією з ключових технологій реалізації концепції хмарних обчислень. Проте задача оптимального розміщення віртуальних машин на фізичних серверах досі залишається актуальною і полягає як в оптимізації енергоспоживання та рівня утилізації ресурсів відімкнених серверів, так і забезпеченні заданого рівня обслуговування. При цьому прогнозування зміни продуктивності віртуаль-

них машин при їх розміщенні є основною проблемою, оскільки це пов'язано з втратами і штрафами внаслідок порушення вимог облуговування. На рисунку 2.16 показано схему керування хмарною інфраструктурою із забезпеченням умов SLA.

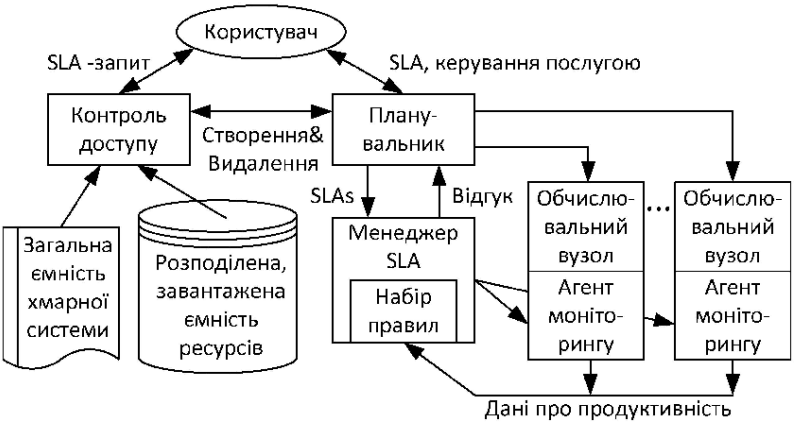


Рисунок 2.16 – Схема системи керування хмарною інфраструктурою

Агенти моніторингу надають інформацію планувальнику про поточний стан фізичних та віртуальних ресурсів. Менеджер SLA на основі набору правил відкидає рішення планувальника, які призводять до деградації продуктивності в системі.

Для зменшення простою ресурсів, як правило, використовують динамічну оптимізацію розміщення віртуальних машин, зокрема з реалізацією оверселінгу, що полягає у перепродажі незадіяних (проте зарезервованих) ресурсів

[48]. При цьому система керування ІТ-інфраструктурою повинна розв’язувати задачі енергозбереження шляхом розвантаження і вимкнення живлення слабозавантажених фізичних машин та попередження зниження продуктивності віртуальних машин, що розміщені на хостовому сервері з високим рівнем утилізації ресурсів. Основною причиною зниження продуктивності віртуальних машин, які працюють на спільній інфраструктурі апаратного забезпечення, є їх недостатня ізоляція. Ефект змагання віртуальних машин за різні фізичні ресурси ще називають «інтерференцією». Інтереференція виникає на рівні апаратних компонентів, таких як процесори, пам'ять, засоби введення-виведення та мережевий канал, і підсилюється їх спільним впливом [46, 48]. Прогнозування зниження продуктивності віртуальних машин дозволяє підтримувати заданий рівень обслуговування SLA шляхом урахування відповідної інформації при міграції існуючих або розміщенні нових віртуальних машин на фізичних хостових серверах.

Обсяг різного типу ресурсів, який використовує будь-яка віртуальна машина, залежить від додатків, що працюють під її керуванням. Для спрощення віртуальні машини можуть бути класифіковані на такі типи:

1) віртуальні машини, що інтенсивно використовують ресурс процесорів;

2) віртуальні машини, які інтенсивно використовують ресурс оперативної пам’яті;

3) віртуальні машини, що інтенсивно використовують ресурс дискової пам’яті;

4) віртуальні машини, які інтенсивно використовують

мережевий ресурс.

Для точної картини використання ресурсів віртуальною машиною необхідно розглядати вектор із чотирма компонентами, що відповідають рівню використання ресурсів кожного типу :

$$R_{VM} = < E_{cpu}, E_{mem}, E_{disk}, E_{bw} >,$$

де E_{cpu} – середній відсоток використання процесора;

E_{mem} – середній відсоток використання оперативної пам'яті;

E_{disk} – середній відсоток використання дискової пам'яті;

E_{bw} – середній відсоток використання ресурсу мережі.

У разі гетерогенного дата-центра ресурсні вимоги віртуальної машини необхідно описувати вектором ресурсів, де кожен компонент розраховується як відсоток від загального обсягу ресурсів конкретного сервера, що є кандидатом на розміщення віртуальної машини. Наприклад, рівень споживання процесорного ресурсу може бути розрахований за формулою

$$\mathrm{E_{cpu}} = \frac{\mathrm{V_{cpu}^{VM}}}{\mathrm{C_{cpu}^{PM}}},$$

де V_{cpu}^{VM} – середній обсяг використання процесорного ресурсу віртуальною машиною;

C_{cpu}^{PM} – максимальна ємність процесорного ресурсу фізичного сервера.

Для оцінювання можливостей фізичного сервера щодо розгортання додаткової віртуальної машини потрібно обчислити вектор сумарного використання ресурів сервера всіма віртуальними машинами, розміщеними на ньому :

$$R_{PM}^{used} = < E_{cpu}^{used}, E_{mem}^{used}, E_{disk}^{used}, E_{bw}^{used} >,$$

де $E_{cpu}^{used}, E_{mem}^{used}, E_{disk}^{used}, E_{disk}^{used}$ – відсоток зайнятого ресурсу процесора, оперативної пам'яті, диска та мережі відповідно.

Аналогічно можна описати вектор незайнятих ресурсів фізичного сервера :

$$R_{PM}^{free} = < E_{cpu}^{free}, E_{mem}^{free}, E_{disk}^{free}, E_{bw}^{free} >,$$

де $E_{cpu}^{free}, E_{mem}^{free}, E_{disk}^{free}, E_{disk}^{free}$ – відсоток зайнятого ресурсу процесора, оперативної пам'яті, диска та мережі відповідно.

Якщо віртуальна машина щойно створена, то її шаблон споживання ресурсів апріорно невідомий і може бути заданий за замовчуванням, наприклад у вигляді вектора з

такими значеннями:

$$R_{VM} = < 25\ \%, 25\ \%, 25\ \%, 25\ \% > .$$

Для уникнення конкуренції віртуальних машин за доступ до ресурсів фізичного сервера варто розміщувати віртуальні машини за принципом неподібності до їх ресурсних вимог, або за принципом подібності набору вільних ресурсів фізичного сервера до ресурсних вимог віртуальної машини, що переміщується. Як критерій подібнності (неподібності) можна використати косинусну міру, що обчислюється за однією з двох формул :

$$similarity = \frac{R_{VM} \cdot R_{PM}^{used}}{\left\| R_{VM} \right\| \cdot \left\| R_{PM}^{used} \right\|}, \qquad (2.4.4)$$

$$similarity = \frac{R_{VM} \cdot R_{PM}^{free}}{\left\| R_{VM} \right\| \cdot \left\| R_{PM}^{free} \right\|}. \qquad (2.4.5)$$

Пошук максимальної неподібності моделей споживання ресурсів за формулою (2.4.4) дозволяє підвищити консолідацію гетерогенних віртуальних машин. Пошук максимальної подібності за формулою (2.4.5) дозволяє підвищити раціональність використання ресурсів на фізичних машинах і одночасно зменшити ймовірність виникнення конкуренції віртуальних машин за спільні ресурси сервера.

У разі обмежених ресурсів дата-центра прогнозування

рівня інтерференції дозволяє приймати рішення, оптимальні у вартісному сенсі. При цьому повторюваний характер задач, що розв'язуються додатками віртуальних машин, забезпечує можливість застосування методів машинного навчання для аналізу log-даних трасування роботи віртуальних машин і синтезу моделі прогнозування рівня їх інтерференції [48]. На рисунку 2.17 показано узагальнену схему системи прогнозування зниження продуктивності віртуальних машин внаслідок розміщення на сервері додаткової віртуальної машини.

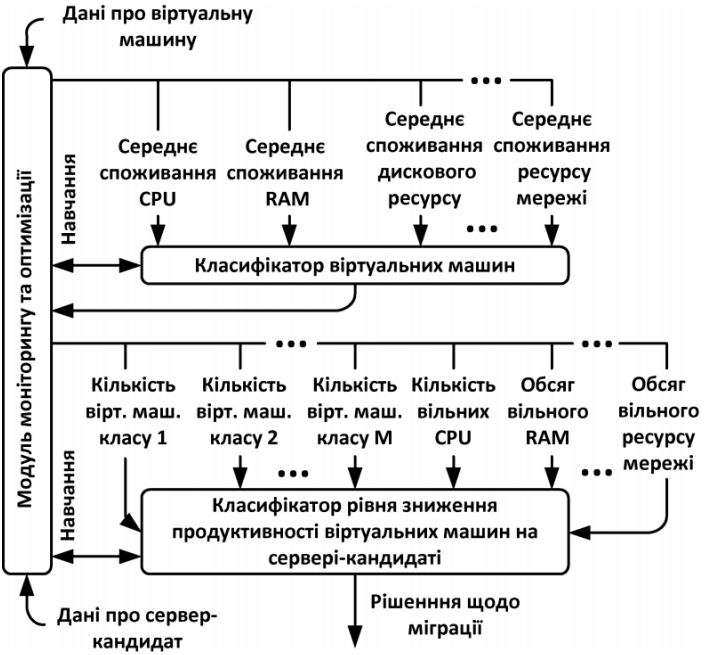


Рисунок 2.17 – Схема системи прогнозування рівня зниження продуктивності віртуальних машин

Вхідний математичний опис показаної на рисунку 2.17 системи може бути сформований за результатами кластер-аналізу даних трасування роботи віртуальних машин. Утворені таким чином групи (класи) віртуальних машин мають подібні середні значення споживання різного типу ресурсів фізичного сервера. При цьому словник ознак класифікатора віртуальних машин повинен містити середній обсяг використання ресурсу процесорів, оперативної пам'яті, файла підкачування, мережевого каналу, дискового простору, середню інтенсивність операцій введення-виведення з дисковою пам'яттю тощо. Так само в процесі кластер-аналізу може бути сформований алфавіт рівнів інтерференції, де словник ознак містить записи про зміну у відсотках сумарного споживання різного типу ресурсів, зміну метрик продуктивності, відсоток помилок оброблення задач на віртуальній машині, відсоток часу перебування процесора в стані блокування та інші зміни, викликані розміщенням нової віртуальної машини.

Останній крок аналізу даних передісторії полягає у зв'язуванні кожної знайденої комбінації розміщення віртуальних машин на фізичних серверах із відповідним класом рівня інтерференції. Як ознаки можна розглядати кількість розміщених на хості віртуальних машин кожного класу та метрики споживання ресурсів процесора, оперативної та дискової пам'яті та мережевого каналу.

У процесі машинного навчання здійснюється синтез класифікатора рівнів інтерференції віртуальних машин за різних комбінацій їх розміщення на фізичних серверах. При цьому належність віртуальної машини до одного з

шаблонів споживання ресурсів ураховується під час прогнозу рівня інтерференції.

Як бізнес-модель хмарні обчислення покликані надавати користувачам високодоступного, надійного, масштабованого і недорогого динамічного обчислювального середовища. Проте поширення хмарних обчислень, а отже, і створення великомасштабних центрів оброблення даних по всьому світу, які містять величезну кількість взаємозв'язаних обчислювальних вузлів, призвели до нових проблем у керуванні цим обчислювальним середовищем: відмови та високе енергоспоживання. Відмови вузлів є характерними для великомасштабованих розподілених систем, які можуть мати тисячі вузлів, що виконують різноманітні завдання. На думку дослідників із Лос-Аламоської національної лабораторії (Los Alamos National Laboratory), середній час між відмовами вузла в петафлоп-системі становить 1,25 години [49]. Не викликає сумнівів доступна інформація про те, що випадки відмов стануть набагато поширенішими в найближчі десять років, що серйозно впливатиме на продуктивність та витрати на експлуатацію. Крім того, внаслідок поширення хмарних обчислень значно зростає обсяг споживання електричної енергії центрами оброблення даних, що призводить до високих експлуатаційних витрат і викидів вуглекислого газу в довкілля. Ці фактори змушують постачальників хмарних сервісів шукати шляхи скорочення експлуатаційних витрат.

На рис. 2.18 показано графік очікуваного збільшення інтенсивності відмов при подвоєнні кількості працюючих процесорних ядер кожні 18, 24 і 30 місяців відповідно.

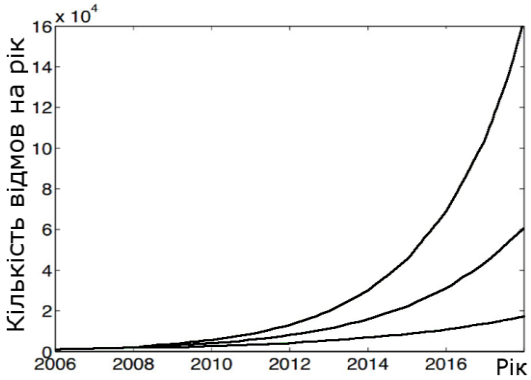


Рисунок 2.18 – Очікуване зростання частоти відмов у хмарному середовищі

Відмови, що виникають в обчислювальному середовищі, швидко поширюються і чинять вплив на роботу сервісів та користувачів. При цьому виявлення збоїв та відмов часто займає значний час, оскільки апаратний рівень фізичних серверів прихований за шаром віртуалізації. Тому досі лишається актуальним завдання оперативного виявлення несправностей і реагування на них, перш ніж вони заподіють збитків. Один із перспективних підходів до оперативного виявлення несправностей полягає у прогнозуванні будь-якої відмови сервісів на основі аналізу журналів, повідомлень, що містять інформацію про роботу пристроїв обчислювальної системи.

У праці [50] запропоновано використовувати приховану напівмарківську модель для аналізу порядку повідомлень в журналі та виявлення послідовностей, пов’язаних із відмовами. Проте застосування цього методу до великої

системи, наприклад хмарних центрів оброблення даних, ускладнено рядом проблем:

1) формати повідомлень у гетерогенній системі різні;

2) порядок повідомлень у журналі може не відповідати порядку їх генерації в системі внаслідок затримок під час збирання й записування;

3) застарівання результатів навчання прогностичної системи через оновлення апаратного чи програмного забезпечення компонентів розподіленого обчислювального середовища.

Дослідниками з компанії Fujitsu було запропоновано знаходити статистичну залежність між численними повідомленнями у журналі подій та відмовами шляхом зіставлення комбінації повідомлень, одержаних упродовж заданого періоду, із прецедентами передаварійних повідомлень. При цьому ознаками передаварійного стану є кількість повідомлень заданого типу, одержаних упродовж усього часового вікна спостереження (рис. 2.19) [51].

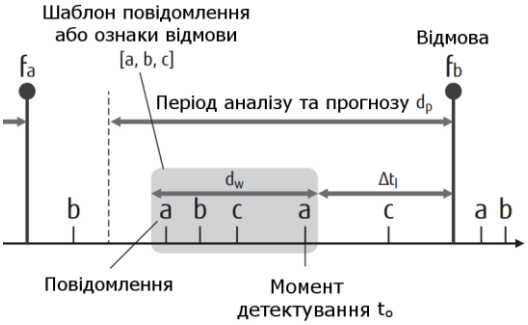


Рисунок 2.19 – Періоди аналізу повідомлень і прогнозування

Обчислення кожної ознаки здійснюється за результатом класифікації типу повідомлення шляхом розбиття тексту кожного повідомлення на слова та обчислення статистики входження слів повідомлення до словника повідомлень.

З метою підтримання релевантності вирішальних правил усі виявлені відмови в сервісах повинні зіставлятися з комбінацією повідомлень упродовж часового вікна спостереження. Крім того, у міру накопичення архівних даних повинна оновлюватися вхідна навчальна матриця передаварійних ситуацій. Так само для доповнення навчальної матриці необхідно використовувати ознаки спрогнозованих відмов, що пройшли валідацію.

Таким чином, функціональний стан компонентів розподіленого обчислювального середовища може бути спрогнозований, що дозволяє зменшити ймовірність зниження продуктивності, перевантажень та відмов під час оптимізації енерговитрат.

2.5. Контрольні запитання та завдання для самопідготовки

1. Що називається обчислювальним кластером?
2. Що називається GRID-системою?
3. Що називається GRID-мережею?
4. Які основні технічні особливості GRID-систем?
5. Які основні особливості GRID-систем із точки зору користувача?
6. Які типи GRID-систем?

7. Що називається хмарними обчисленнями?

8. Яка основна концепція оброблення даних у хмарах?

9. Що називається еластичністю послуг?

10. Які класи послуг для хмарної платформи?

11. Які категорії хмар?

12. Що називається приватною хмарою?

13. Що називається публічною хмарою?

14. Що називається змішаною хмарою?

15. Які основні два підходи до віртуалізації ресурсів при хмарних обчисленнях?

16. Що називається гіпервізором?

17. Що називається «товстим» гіпервізором?

18. Що називається «мікроядерним» гіпервізором?

19. Що називається «тонким» гіпервізором?

20. Нарисуйте спрощену схему взаємодії сервісів у сервіс-орієнтованому середовищі?

21. Що називається обчислювальною задачею?

22. Що називається віртуальним вузлом?

23. Що називається статичним плануванням завдань?

24. Що називається динамічним плануванням завдань?

25. Які особливості мають непов’язані алгоритми планування завдань?

26. Які особливості мають пов’язані алгоритми планування завдань?

27. Яке призначення спискових алгоритмів планування завдань?

28. Наведіть приклади найбільш відомих спискових алгоритмів планування завдань.

29. Які особливості алгоритмів кругової диспетчери-

зації завдань?

30. Які особливості застосування алгоритму зворотного заповнення під час планування завдань?

31. Які особливості алгоритму планування групами завдань?

32. У яких випадках доцільно використовувати евристичні методи оптимізації рішень при плануванні завдань?

33. Які особливості евристичних методів оптимізації рішень під час планування завдань?

34. Які особливості методів оптимізації завдань на основі системно-центричних критеріїв?

35. Які основні особливості алгоритмів організації хмарних обчислень?

36. Які основні напрями енергозбереження в інфокомунікаційних системах?

37. Які особливості статичного підходу до енергозбереження в інфокомунікаційних системах?

38. Які особливості динамічного підходу до енергозбереження в інфокомунікаційних системах?

39. Яке призначення бази профілів?

40. Як формуються еталонні зразки часових рядів споживання ресурсів?

41. Яке призначення алгоритму динамічної трансформації шкали часу (Dynamic Time Varping, DTV?

42. Яке призначення системи контролю доступу (Admission Control)?

43. Яка ідея евристичного контролю доступу до ресурсів кластера на основі моделі косинусної надійності?

44. Які ресурсні ознаки обчислювального вузла є

статичними?

45. Які ресурсні ознаки відносять до динамічних?

46. Яке призначення оверселінгу?

47. Назвіть основні типи віртуальних машин.

48. Що називається інтерференцією в інфокомунікаційній системі?

РОЗДІЛ 3
ОПТИМІЗАЦІЯ СИСТЕМИ КЕРУВАННЯ РОЗПОДІЛЕНИМ ОБЧИСЛЮВАЛЬНИМ СЕРЕДОВИЩЕМ

3.1. Інформаційні характеристики системи керування

Центральним питанням інформаційного аналізу і синтезу системи керування будь-яким інфокомунікаційним сервісом є вибір і конструювання критеріїв оптимізації його параметрів функціонування. Оскільки інфокомунікаційна система здійснює одержання, оброблення, збереження і передавання інформації, то завдання її оптимізації полягає в максимізації інформаційної спроможності системи.

Вчення про інформацію перебуває в процесі інтенсивного розвитку, його значення зростає через безперервне ускладнення та інтелектуалізацію комп'ютеризованих систем керування, що застосовуються в різних галузях соціально-економічної сфери суспільства [52–54].

Визначення 3.1.1. Під інформацією розуміють відомості, які містяться в повідомленні та є об'єктом одержання, оброблення, збереження і передавання.

Таке визначення інформації у широкому розумінні підкреслює її ціннісний аспект, необхідність її використання.

Визначення 3.1.2. У логіко-гносеологічному аспекті інформація розглядається як перетин категорії відбиття з категорією різноманітності, тобто є відбитою різноманітністю.

Формування теорії інформації було зумовлено практичними потребами суспільства і мало вирішальне значення

в становленні такої науки про керування, як кібернетика.

Визначення 3.1.3. З логіко-гносеологічної точки зору кібернетика визначається як перетин категорії керування з відбитою різноманітністю, тобто інформацією (рис. 3.1).

Таким чином, природа кібернетики має інформаційну основу.

Рисунок 3.1 – До пояснення природи кібернетики

Інформацію як міру знятої невизначеності можна виміряти. За одиницю вимірювання інформації береться біт (бінарна одиниця)

Визначення 3.1.4. Біт – це кількість інформації, яку одержує особа, що приймає рішення щодо вибору однієї гіпотези із двох рівноймовірних, взаємовиключних гіпотез.

В обчислювальній техніці біт широко вживається як одиниця ємності пам'яті пристрою. Ця одиниця збігається з інформаційною одиницею у тому разі, коли є обґрунтування рівноймовірної появи, наприклад, «1» або «0» у бінар-

ній комірці пам'яті.

Існують різні інформаційні міри. Найпростішою такою мірою є кількість повідомлень:

$$N = m^{n}, \qquad (3.1.1)$$

де m – кількість якісних ознак у повідомленні; n – кількість елементів у повідомленні.

Недоліком міри (3.1.1) є те, що при $m = const$ між кількістю повідомлень і кількістю елементів не існує адитивної залежності. У 1924 році Л. Хартлі запропонував інформаційну міру у вигляді

$$I = \log_2 N = n \log_2 m. \qquad (3.1.2)$$

Аналіз виразу (3.1.2) показує, що в мірі Хартлі існує адитивна залежність між кількістю інформації та кількістю елементів у повідомленні, але ця міра не враховує ймовірнісних показників повідомлення.

Серед імовірнісних мір інформації найбільшого поширення дістала міра Шеннона [52]. Кількість безумовної інформації за Шенноном, яка міститься в повідомленні $A = \{a_i\}$, дорівнює

$$I = -\sum_{i=1}^{m} p(a_i) \log_2 p(a_i), \qquad (3.1.3)$$

де $p(a_i)$ – безумовна ймовірність появи в повідомленні елемента a_i.

Перехід статистичної міри Шеннона в граничному ви-

падку при $p_i = 1/m$ у детерміновану міру Хартлі

$$I = -n\sum_{i=1}^{m} p_i \log_2 p_i = -n\sum_{i=1}^{m} \frac{1}{m} \log \frac{1}{m} = n \log_2 m$$

відповідає співвідношенню категорій випадковості й детермінованості та підтверджує її релевантність в рамках детерміновано-статистичного підходу до інформаційного синтезу інтелектуальних систем керування.

Визначення 3.1.5. Питома кількість інформації, яка міститься в одному незалежному повідомленні *A,* називається загальною (середньою) безумовною ентропією і визначається за формулою

$$H(A) = \frac{I(A)}{n} = -\sum_{i=1}^{m} p(a_i) \log_2 p(a_i) \left[\frac{біт}{символ} \right]. \qquad (3.1.4)$$

При цьому кількість інформації, яка міститься в одному незалежному елементі a_i, називається частинною безумовною ентропією:

$$H(a_i) = -p(a_i) \log_2 p(a_i).$$

Таким чином, ентропія, як і кількість інформації, є мірою невизначеності системи. Розглянемо основні властивості безумовної ентропії, позначивши ймовірність $p(a_i)$ як p_i.

1. Ентропія є величина дійсна і знакододатна $(H \geq 0)$,

оскільки $0 \le p_i \le 1$, то $\log_2 p_i \le 0$.

2. Безумовна ентропія для детермінованих змінних ($p_i = 1$ або $p_i = 0$) дорівнює нулю.

Оскільки $\sum_{i=1}^{m} p_i = 1$, то для $= 1$ і $p_i = 1$ ентропія дорівнює $H = -p_i \log_2 p_i = -1 \log_2 1 = 0$. Для $m > 2$ необхідно довести, що ентропія дорівнює нулю і при ймовірності $p_i = 0$, яку будуть мати m–1 елементів повідомлення. Оскільки функція (3.1.3) в цьому випадку дає невизначеність $\{0 \cdot \infty\}$, то для використання правила Лопіталя її необхідно перетворити до вигляду (∞ / ∞) шляхом підстановки $k = 1/p_i$. Тоді можна записати

$$\lim_{p_i \to 0} \frac{\log_2 \frac{1}{p_i}}{\frac{1}{p_i}} = \left[\lim_{p_i \to 0} \frac{\log_2 k}{k} \right]^{L} = \lim_{k \to \infty} \frac{1}{k} = 0,$$

де L – знак процедури Лопіталя.

Таким чином, для детермінованих станів і при $m \ge 2$ функція (3.1.4) дорівнює нулю, оскільки дорівнюють нулю всі її складові.

3. Безумовна ентропія максимальна при рівноймовірних подіях ($p_i = 1/m$) і дорівнює $H_{\max} = \log_2 m$.

Для $m \le 2$ ця властивість доводиться шляхом дослідження функції (3.3.1) на екстремум із підстановкою в неї, наприклад, для $m = 2$ змінних $p_i = p$ і $p_2 = 1–p$. Для доведення при $m > 2$ можна за допомогою нормованої функції

$\sum_{i}^{m} p_i = 1$ сформувати функціонал [53]:

$$\Phi = -\sum_{i=1}^{m} p_i \log_2 p_i - \lambda \sum_{i=1}^{m} p_i = -\sum \left(p_i \log_2 p_i - \lambda p_i \right) = \sum_{i=1}^{m} \Phi_i ,$$

де λ — невизначений множник Лагранжа.

Взявши похідну та прирівнявши її до нуля,

$$\text{тобто} \quad \frac{d\Phi_i}{dp_i} = -\frac{\left(\ln p_i + 1 \right)}{\ln 2} + \lambda = 0 ,$$

одержуємо $p_i = \exp\left(\lambda \ln 2 - 1 \right)$. Оскільки

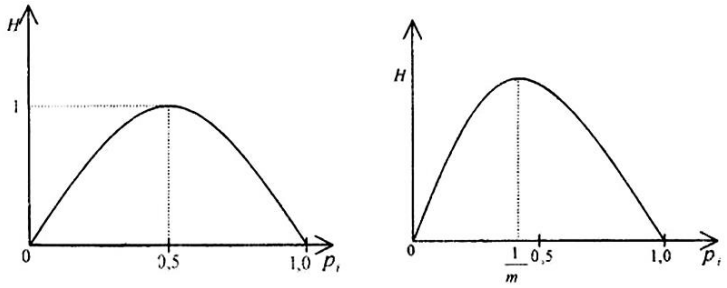
$$\sum_{i=1}^{m} \exp\left(\lambda \ln 2 - 1 \right) = m \exp\left(\lambda \ln 2 - 1 \right) = 1 ,$$

то

$$p_i = \exp\left(\lambda \ln 2 - 1 \right) = \frac{1}{m} .$$

Таким чином, і для $m > 2$ при $p_i = 1/m$ функція (3.1.4) має екстремум. Внаслідок першої та другої властивостей можна стверджувати без аналізу другої похідної, що цей екстремум є максимумом функції.

На рисунку 3.2 наведено графіки функції (3.1.4) при $m = 2$ (рис. 3.2 *а*) і $m > 2$ (рис. 3.2 *б*).



а *б*

Рисунок 3.2 – Графіки залежності ентропії від імовірності:
а – $m = 2$; *б* – $m > 2$

4. Ентропія об'єднання незалежних повідомлень $H(A,B)$ дорівнює їх сумі.

Відомо із теорії ймовірностей, що для двох незалежних подій має місце $p(a_i, b_i) = p(a_i)\,p(b_j)$. Тоді

$$H(A,B) = -\sum_{i,j} p(a_i) p(b_j) \log_2 p(a_i)\,p(b_j) =$$

$$= -\sum_{i,j} p(a_i)\,p(b_j)\left[\log_2 p(a_i) + \log_2 p(b_j)\right] =$$

$$= -\sum_i p(a_i) \log_2 p(a_i) \sum_j p(b_j) - \sum_i p(a_i) \sum_j p(b_j) \log_2 p(b_j).$$

Оскільки $\sum_i p(a_i) = 1$ і $\sum_j p(b_j) = 1$, то

$$H(A,B) = H(A) + H(B),$$

що і треба було довести.

При детальному розгляді всі події є залежними від певних умов. Так, вихід об'єкта керування $B=\{b_j\}$ залежить від вектора вхідних даних $A=\{a_i\}$ і випадкового збурення $f(t)$, як це показано на рис. 3.3.

Рисунок 3.3 – Узагальнена інформаційна модель об'єкта керування

Рішення, які приймаються системою керування, характеризуються у цьому разі умовною апріорною ймовірністю $p(b_j/a_i)$ появи на виході об'єкта керування події b_j за умови, що на його вході має місце подія a_i. При цьому умовна ймовірність $p(a_i/b_j)$ наявності події a_i на вході об'єкта керування за умови, що на виході одержано подію b_j, називається апостеріорною.

За апріорні інформаційні характеристики, які є мірою невизначеності виходу об'єкта керування (рис. 3.3), можна розглядати частинну умовну ентропію

$$H(b_j/a_i)=-\sum_j p(b_j/a_i)\log_2 p(b_j/a_i) \qquad (3.1.5)$$

і середню (загальну) умовну ентропію $H(B/A)$, яка є математичним сподіванням від відповідної частинної ентропії:

$$H\left(B/A\right)=M\left[H\left(B/A\right)\right]=\sum_i p\left(a_i\right)H\ \left(b_j/a_j\right)= \\ =-\sum_i p\left(a_i\right)\sum_j p\ \left(b_j/a_j\right)\log_2 p\ \left(b_j/a_j\right). \qquad (3.1.6)$$

Аналогічно визначаються апостеріорні частинна умовна ентропія:

$$H\left(a_i/b_j\right)=\sum_i p\left(a_i/b_j\right)\log_2 p\left(a_i/b_j\right) \qquad (3.1.7)$$

і середня умовна ентропія як математичне сподівання від частинної умовної ентропії:

$$H\left(A/B\right)=\sum_j p\ \left(b_j\right)H\left(a_i/b_j\right)= \\ =-\sum_j p\left(b_j\right)\sum_i p\left(a_i/b_j\right)\log_2 p\left(a_i/b_j\right). \qquad (3.1.8)$$

Середня умовна ентропія має такі специфічні властивості.

1. Об'єднання ентропій двох взаємозалежних повідомлень A і B дорівнює сумі середньої безумовної ентропії одного повідомлення і середньої умовної ентропії іншого повідомлення відносно першого:

$$H\left(A,B\right)=H(A)+H\left(B/A\right)=H\left(B\right)+H\left(A/B\right)=H\left(B,A\right). \qquad (3.1.9)$$

Під час доведення (3.1.9) потрібно взяти до відома, що

для взаємозалежних подій має місце $p(a_i / b_j) = p(a_i)\, p(b_j / a_i)$. Тоді

$$H(A,B) = -\sum_{i,j} p(a_i) p(b_j / a_i) \log_2 p(a_i)\, p(b_j / a_i) =$$

$$= -\sum_{i,j} p(a_i)\, p(b_j / a_i) \left[\log_2 p(a_i) + \log_2 p(b_j / a_i) \right] =$$

$$= -\sum_i p(a_i) \log_2 p(a_i) \sum_j p(b_j / a_i) -$$

$$-\sum_i p(a_i) \sum_j p(b_j / a_i) \log_2 p(b_j / a_i) = H(A) + H(B / A),$$

оскільки $\sum_i p(b_j / a_i) = 1$ як сума ймовірностей подій, що складають повну групу.

2. Умовна ентропія незалежних подій A і B дорівнює відповідній безумовній ентропії:

$$H(A / B) = -\sum_j p(b_j) \sum_i p(a_i / b_j) \log_2 p(a_i / b_j) =$$

$$= -\sum_J p(b_J) \sum_i p(a_i) \log_2 p(a_i) = H(A),$$

оскільки $\sum_j p(b_j) = 1$.

3. Умовна ентропія повідомлень A і B, які мають детерміновану (функціональну) залежність, дорівнює нулю.

Під час доведення необхідно взяти до уваги, що для подій a_i і b_j з детермінованим зв'язком має місце $p(a_i / b_j) = 1$ і $p(b_j / a_i) = 1$.

Функціональна ефективність системи керування розпо-

діленим обчислювальним середовищем залежить від інформаційного навантаження ознак розпізнавання, які характеризують відповідний функціональний стан інформаційно-комунікаційного сервісу. Тому важливою характеристикою словника ознак розпізнавання є його інформаційна надлишковість, яка обчислюється за формулою

$$D = 1 - \frac{H}{H_{\max}}, \qquad (3.1.10)$$

де H – ентропія вектора ознак розпізнавання; H_{max} – максимальна ентропія вектора ознак розпізнавання рівноймовірними елементами.

Надлишковість інформації не може розглядатися виключно як негативне явище, оскільки її збільшення підвищує завадозахищеність повідомлення, що є основним принципом теорії завадозахищеного кодування.

Однією із основних інформаційних характеристик є кількість середньої (загальної) умовної інформації про повідомлення A (або B), яка міститься у взаємозалежному повідомленні B (або A) і визначається за симетричною формулою

$$I_B(A) = H(A) - H(AB) = H(B) - H(BA) = I_A(B). \qquad (3.1.11)$$

Середня кількість умовної інформації виражається через імовірності подій на вході та виході, наприклад об’єкта керування (рис. 3.1) у вигляді [16] :

$$I_A\left(B\right)=\sum_{i,j}p\left(a_i,b_j\right)\log_2\frac{p\left(a_i/b_j\right)}{p\left(a_i\right)\,p\left(b_j\right)}=$$

$$=\sum_{i,j}p\left(b_j\right)\,p\left(a_i/b_j\right)\log_2\frac{p\left(a_i/b_j\right)}{p\left(a_i\right)}.$$

Середню кількість умовної інформації можна розглядати як математичне сподівання відповідної кількості часткової умовної інформації, яка міститься в елементі a_i, (векторі B) про вектор B (елемент a_i) :

$$I_A\left(B\right)=M\left[I_{a_i}\left(B\right)\right]=\sum_j p\left(b_j\right)I_{a_i}\left(B\right). \qquad (3.1.12)$$

Звідси з урахуванням (3.1.12) кількість часткової умовної інформації визначається як

$$I_{a_i}\left(B\right)=\sum_i p\left(a_i/b_j\right)\log_2\frac{p\left(a_i/b_j\right)}{p\left(a_i\right)}. \qquad (3.1.13)$$

Аналогічно кількість часткової умовної інформації є математичним сподіванням відповідної індивідуальної умовної інформації, яка міститься в елементі повідомлення $a_i\,(b_j)$ про елемент $b_j\,(a_i)$:

$$I_{a_i}\left(B\right)=M\left[I_{a_i}(b_j)\right]=\sum_i p\left(a_i\right)I_{a_i}\left(b_j\right). \qquad (3.1.14)$$

З урахуванням (3.1.14) одержуємо формулу кількості індивідуальної умовної інформації :

$$I_{a_i}(b_j) = \log_2 \frac{p(a_i / b_j)}{p(a_i)} = \log_2 \frac{p(b_j / a_i)}{p(b_j)} = I_{b_j}(a_i). \quad (3.1.15)$$

Аналіз виразу (3.1.15) свідчить, що індивідуальна умовна інформація може бути від'ємною, тобто розглядатися як дезінформація.

Важливою характеристикою системи керування інформаційно-телекомунікаційним сервісом є інформаційна пропускна спроможність

$$C_I = \frac{I_A(B)}{\tau} = \frac{H(A) - H(B/A)}{\tau} \left[\frac{біт}{од.часу} \right], \quad (3.1.16)$$

де τ – часовий інтервал функціонування системи керування.

Основними шляхами підвищення інформаційної перепускної спроможності системи керування є:

1) конструктивні рішення, пов’язані із забезпеченням захисту інформації;

2) застосування криптографічних методів захисту інформації;

3) завадозахищене кодування інформації.

Таким чином, теоретико-інформаційний підхід дозволяє створити вхідний математичний опис системи керування будь-яким слабко формалізованим об’єктом, до яко-

го належить інформаційно-телекомунікаційна система.

3.2. Критерії оцінювання функціональної ефективності та оптимізації системи керування, що навчається

Керування інформаційно-телекомунікаційною системою здійснюється за довільних початкових умов та впливу зовнішніх і внутрішніх збурювальних факторів, що обумовлює апріорну невизначеність. На практиці надання системі керування властивості адаптивності здійснюється шляхом її машинного навчання. Одним із перспективних напрямків інформаційного аналізу і синтезу систем керування є застосування ідей і методів інформаційно-екстремальної інтелектуальної технології (ІЕІ-технології) аналізу даних, яка ґрунтується на максимізації інформаційної спроможності системи в процесі її машинного навчання [56–58]. У рамках ІЕІ-технології підвищення функціональної ефективності, здатної навчатися системи керування розподіленим обчислювальним середовищем, здійснюється шляхом цілеспрямованого в процесі навчання пошуку глобального максимуму інформаційного критерію функціональної ефективності (КФЕ) у робочій (допустимій) області визначення його функції.

У загальному випадку КФЕ, здатної навчатися системи керування, повинен відповідати таким основним вимогам:

- бути прямим і об’єктивним критерієм;
- бути релевантним, тобто характеризувати ступінь відповідності системи своєму призначенню, та економічно обґрунтованим;

- мати конструктивний характер, тобто дозволяти розробляти методи аналізу та синтезу системи керування, що навчається;
- бути універсальним, тобто здатним оцінювати функціональну ефективність систем керування широкого призначення;
- бути чутливим до зміни параметрів функціонування і характеристик системи;
- дозволяти оптимізувати параметри функціонування системи керування з метою максимізації її функціональної ефективності;
- мати функціональний зв′язок із точнісними характеристиками рішень, що приймаються системою керування;
- дозволяти прогнозувати зміну функціональної ефективності та надійності системи керування.

Ці вимоги повністю задовольняють критерії, які характеризують інформаційну спроможність інтелектуальної системи, що приймає рішення за апріорної невизначеності.

У методах IEI-технології для оптимізації параметрів навчання системи керування широко застосовується ентропійний КФЕ, який має такий нормований вигляд [52] :

$$E = \frac{H_0 - H(\gamma)}{H_0}, \qquad (3.2.1)$$

де H_0 – апріорна (безумовна) ентропія:

$$H_0 = -\sum_{l=1}^{M} p\ (\gamma_l)\ \log_2\ p\ (\gamma_l); \qquad (3.2.2)$$

$H(\gamma)$ – апостеріорна умовна ентропія, яка характеризує залишкову невизначеність після прийняття рішень:

$$H(\gamma) = -\sum_{l=1}^{M} p(\gamma_l) \sum_{l=1}^{M} p(\mu_m / \gamma_l) \log_2 p(\mu_m / \gamma_l). \quad (3.2.3)$$

У формулах (3.2.1) і (3.2.2) відповідно $p(\gamma_l)$ – безумовна ймовірність прийняття апріорної гіпотези γ_l і $p(\mu_m / \gamma_l)$ – умовна ймовірність прийняття апостеріорної гіпотези μ_m за умови існування апріорної гіпотези γ_l; M – кількість гіпотез.

На практиці під час оцінювання функціональної ефективності, здатної навчатися системи керування, можуть мати місце такі припущення :

- рішення є двохальтернативним ($M = 2$);
- оскільки система керування функціонує за умови невизначеності, то за принципом Бернуллі-Лапласа виправдано прийняття на практиці рівноймовірних як апріорних, так і апостеріорних гіпотез, тобто $p(\gamma_1) = p(\gamma_2) = 0,5$ і $p(\mu_1) = p(\mu_2) = 0,5$ відповідно.

Тоді критерій (3.2.1) з урахуванням виразів (3.2.2) і (3.2.3) набирає такого частинного вигляду :

$$E = 1 + \frac{1}{2} \sum_{l=1}^{2} \sum_{m=1}^{2} p(\mu_m / \gamma_l) \log_2 p(\mu_m / \gamma_l). \quad (3.2.4)$$

За двохальтернативних рішень ($M = 2$) за основну бе-

реться гіпотеза γ_1 про знаходження значення ознаки розпізнавання в симетричному двобічному полі допусків δ і як альтернативну їй – гіпотезу γ_2.

На рисунку 3.4 показано чотири можливих результати оцінювання виміру ознаки розпізнавання, одержані за двохальтернативних рішень. Тут величини *x* і *z* – виміряне і дійсне значення ознаки розпізнавання відповідно.

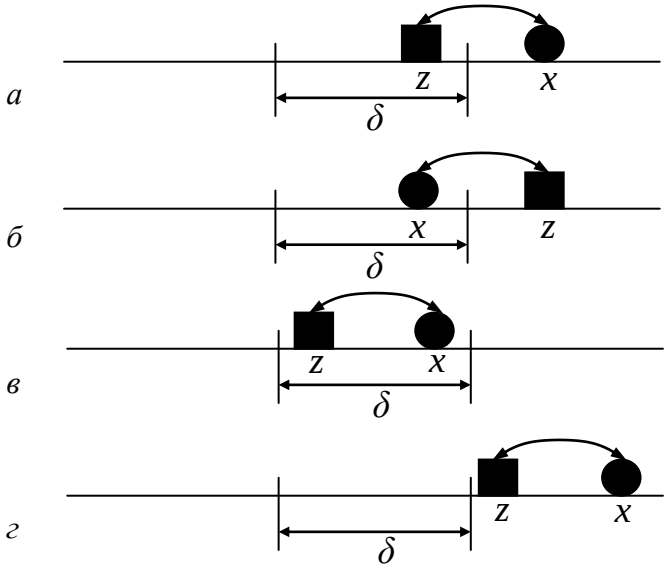


Рисунок 3.4 – Можливі наслідки контролю ознак розпізнавання за двохальтернативних рішень

Результати контролю ознак розпізнавання за двохальтернативних рішень характеризуються такими ймовірностями – точнісними характеристиками :

- помилка першого роду – $\alpha = \rho(x \notin \delta / z \in \delta)$ (рис. 3.4 *а*);
- помилка другого роду – $\beta = \rho(x \in \delta / z \notin \delta)$ (рис. 3.4 *б*);
- перша достовірність – $D_1 = \rho(x \in \delta / z \in \delta)$ (рис. 3.4 *в*);
- друга достовірність – $D_2 = \rho(x \notin \delta / z \notin \delta)$ (рис. 3.4 *г*).

Розіб'ємо множину значень ознак на область μ_1, яка містить значення, розміщені в допуску δ, і область μ_2 – не в допуску. Тоді можна позначити :

$$\alpha = p(\gamma_2 / \mu_1),\ \beta = p(\gamma_1 / \mu_2),\quad D_1 = p(\gamma_1 / \mu_1),$$
$$D_2 = p(\gamma_2 / \mu_2). \qquad (3.2.5)$$

Виразимо апостеріорні умовні ймовірності $p(\mu_m / \gamma_l)$ через апріорні за формулою Байєса [55], взявши $p(\mu_1) = p(\mu_2) = 0{,}5$, що є виправданим за умови апріорної невизначеності:

$$p(\mu_1 / \gamma_1) = \frac{p(\gamma_1 / \mu_1)}{p(\gamma_1 / \mu_1) + p(\gamma_1 / \mu_2)} = \frac{D_1}{D_1 + \beta},$$
$$p(\mu_2 / \gamma_1) = \frac{\beta}{D_1 + \beta},\ p(\mu_1 / \gamma_2) = \frac{\alpha}{\alpha + D_2},$$
$$p(\mu_2 / \gamma_2) = \frac{p_2 D_2}{p_1 \alpha + p_2 D_2}. \qquad (3.2.6)$$

Після підстановки (3.2.6 в 3.2.4) одержуємо формулу для обчислення ентропійного КФЕ :

$$E = 1 + \frac{1}{2}\left(\frac{\alpha}{\alpha + D_2} \log_2 \frac{\alpha}{\alpha + D_2} + \frac{D_1}{D_1 + \beta} \log_2 \frac{D_1}{D_1 + \beta} + \right.$$

$$\left. + \frac{\beta}{D_1 + \beta} \log_2 \frac{\beta}{D_1 + \beta} + \frac{D_2}{\alpha + D_2} \log_2 \frac{D_2}{\alpha + D_2} \right). \quad (3.2.7)$$

У загальному випадку побудований за 3.2.7 графік функції $E = f(D_1, D_2)$ є поверхнею у тривимірному просторі (рис. 3.5). При цьому враховано, що $D_1 + \alpha = 1$ і $D_2 + \beta = 1$ як суми ймовірностей подій відповідної групи.

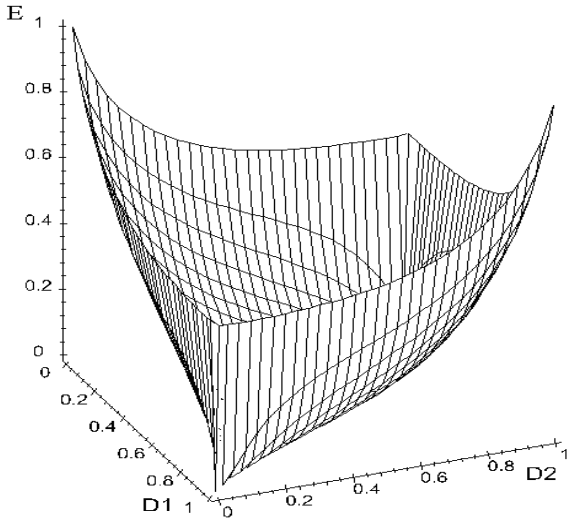


Рисунок 3.5 – Графік залежності критерію (3.2.7) від точнісних характеристик (M = 2)

Як бачимо з рисунка 3.5, функція (3.2.7) не є взаємно-

однозначною. На практиці цей недолік обходять, обмеживши значеннями достовірностей в інтервалі [0,5;1], який характеризує робочу область визначення функції критерію (3.2.7). Таким чином, під час обчислення критерію (3.2.7) в робочій області проблема поділу на нуль не виникає.

На практиці за обчислення інформаційного КФЕ машинного навчання системи керування замість ймовірностей користуються оцінками точнісних характеристик, які для двохальтернативних рішень подамо у вигляді

$$D_1 = \frac{K_1}{n_{\min}},\ \alpha = \frac{K_2}{n_{\min}},\ \beta = \frac{K_3}{n_{\min}},\ D_2 = \frac{K_4}{n_{\min}}, \qquad (3.2.8)$$

де K_1 – кількість подій, що полягають в належності реалізації, яка розпізнається, своєму класу X_1^o (рис. 3.4 *в*); K_2 – кількість подій, що полягають в належності класу X_2^o реалізації класу X_1^o (рис. 3.4 *а*); K_3 – кількість подій, які полягають в належності класу X_1^o реалізації класу X_2^o (рис. 3.4 *б*); K_4 – кількість подій, що полягають в належності реалізації, що розпізнається, своєму класу X_2^o (рис. 3.4 *г*); $n_{\min}$ – мінімальний обсяг репрезентативної навчальної вибірки.

Після підстановки оцінювань (3.2.8) в (3.2.7) одержуємо робочу формулу для обчислення ентропійного КФЕ машинного навчання системи керування розпізнавати реалізації класу X_1^o для двохальтернативних рішень при рів-

ноймовірних гіпотезах:

$$E_1 = 1 + \frac{1}{2}\left(\frac{K_1}{K_1 + K_3} \log_2 \frac{K_1}{K_1 + K_3} + \frac{K_2}{K_2 + K_4} \log_2 \frac{K_2}{K_2 + K_4} + \right.$$

$$\left. + \frac{K_3}{K_1 + K_3} \log_2 \frac{K_3}{K_1 + K_3} + \frac{K_4}{K_2 + K_4} \log_2 \frac{K_4}{K_2 + K_4} \right). \quad (3.2.9)$$

За обчислення критерію за формулами (3.2.7) і (3.2.9) у робочій області визначення функції КФЕ проблема поділу на нуль не виникає, оскільки в цій області перша і друга достовірності набувають значення більше 0,5.

Ентропійна міра Шеннона, яка є інтегральною мірою, серед логарифмічних статистичних інформаційних мір дістала великого поширення. Водночас ще недостатньо уваги приділяється вивченню властивостей міри Кульбака, що дозволяє оцінювати диференційну інформативність ознак розпізнавання.

Розглянемо модифіковану інформаційну міру Кульбака для двохальтернативних рішень, що застосовувалася в праці [56] для оцінювання функціональної ефективності навчання системи розпізнавати реалізації класу X_m^o,

$$E_m = [P_{t,m} - P_{f,m}] \cdot \log_2 \frac{P_{t,m}}{P_{f,m}}, \quad (3.2.10)$$

де $P_{t,m}$ – повна ймовірність правильного розпізнавання

реалізації класу X_m^o; $P_{f,\,m}$ – повна ймовірність неправильного розпізнавання реалізації класу X_m^o.

Повні ймовірності правильного і неправильного прийняття системою керуючих рішень для двохальтернативної системи їх оцінювання згідно з теоремою про повну ймовірність відповідно запишемо як

$$P_{t,\,m}^{(k)} = p\,(\mu_1)\,p\,(\gamma_1/\mu_1) + p(\mu_2)\,p(\gamma_2/\mu_2),$$
$$P_{f,m}^{(k)} = p(\mu_1)\,p(\gamma_2/\mu_1) + p(\mu_2)\,p(\gamma_1/\mu_2), \quad (3.2.11)$$

де $p(\mu_1)$ – безумовна (апріорна) ймовірність належності реалізації, що розпізнається, класу X_1^o; $p(\mu_2)$ – безумовна ймовірність належності реалізації класу X_2^o.

Оскільки умовні ймовірності у виразі (3.2.11) є точнісними характеристиками двохальтернативних рішень (3.2.5), то після відповідної заміни критерій (3.2.10) при рівноймовірних гіпотезах ($p(\mu_1) = p(\mu_2) = 0{,}5$) набирає вигляду

$$E_m = \frac{1}{2}[(D_{1,\,m} + D_{2,m}) - (\alpha_m + \beta_m)]\cdot \log_2\left(\frac{D_{1,m} + D_{2,m} + 10^{-r}}{\alpha_m + \beta_m + 10^{-r}}\right), \quad (3.2.12)$$

де 10^{-r} – достатньо мале число, яке введено для уникнення поділу на нуль.

У формулі (3.2.12) величину константи r рекоменду-

ється вибирати залежно від кількості знаків у мантисі значення КФЕ, але на практиці вона задається в межах $1 < r \le 3$.

Якщо попередньо виразити першу і другу достовірності відповідно через помилки першого і другого роду, тобто

$$D_{1,m} = 1 - \alpha_m,\ D_{2,m} = 1 - \beta_m,$$

зробити їх підстановку у вираз (3.2.12) і замінити відповідні точнісні характеристики їх оцінками (3.2.8), то одержимо робочу формулу для обчислення інформаційного критерію Кульбака за двохальтернативних рішень у вигляді

$$E = \frac{1}{n}[n - (K_2 + K_3)] \cdot \log_2 \left\{ \frac{2n - \left[K_2^{(k)} + K_3^{(k)} \right] + 10^{-r}}{\left[K_2^{(k)} + K_3^{(k)} \right] + 10^{-r}} \right\}. \quad (3.2.13)$$

Одним із шляхів підвищення функціональної ефективності машинного навчання системи керування є перехід від двохальтернативної системи оцінювань рішень, що приймаються, до трьохальтернативної у формі «Менше норми» – «Норма» – «Більше норми».

Із метою встановлення залежності інформаційної міри (3.2.10) від точнісних характеристик за трьохальтернативної системи оцінювань рішень введемо такі позначення: γ_1 – основна гіпотеза про належність ознаки розпізнавання показника «Норма»; γ_2 – гіпотеза про належність ознаки показника «Менше норми»; γ_3 – гіпотеза про належність

ознаки показника «Більше норми».

Відповідно позначимо апостеріорні гіпотези: μ_1 – гіпотеза про те, що значення ознаки дійсно знаходиться в полі допуску δ; μ_2 – значення ознаки – зліва від поля допусків і μ_3 – значення ознаки – правіше поля допусків. Показані на рис. 3.6 можливі результати трьохальтернативних рішень будемо оцінювати такими дев'ятьма точнісними характеристиками:

- перша достовірність $D_{1,m} = p(\gamma_1 / \mu_1)$ (рис. 3.6 *а*);
- перша помилка першого роду $\alpha_{1,m} = p(\gamma_2 / \mu_1)$ (рис. 3.6 *б*);
- друга помилка першого роду $\alpha_{2,m} = p(\gamma_3 / \mu_1)$ (рис. 3.6 *в*);
- друга достовірність $D_{2,m} = p(\gamma_2 / \mu_2)$ (рис. 3.6 *г*);
- перша помилка другого роду $\beta_{1,m} = p(\gamma_1 / \mu_2)$ (рис. 3.6 *д*);
- друга помилка другого роду $\beta_{2,m} = p(\gamma_3 / \mu_2)$ (рис. 3.6 *е*);
- третя достовірність $D_{3,m} = p(\gamma_3 / \mu_3)$ (рис. 3.6 *є*);
- перша помилка третього роду $\sigma_{1,m} = p(\gamma_1 / \mu_3)$ (рис.2.5 *ж*);
- друга помилка третього роду $\sigma_{2,m} = p(\gamma_2 / \mu_3)$ (рис.2.5 *з*).

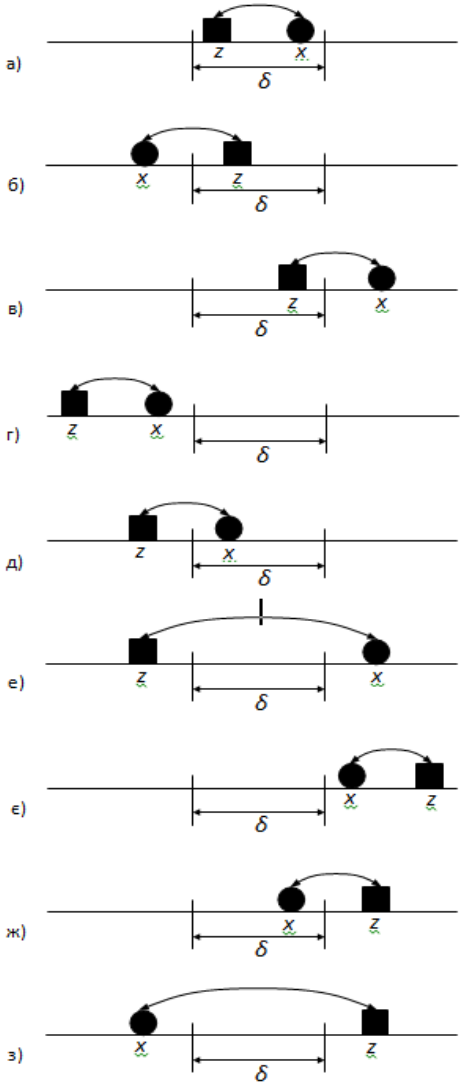


Рисунок 3.6 – Можливі наслідки трьохальтернативних рішень

На практиці можна прийняти такі припущення:

- двобічна система контрольних допусків на ознаки розпізнавання є симетричною;
- точнісні характеристики $\beta_{2,\,m}$ (рис. 2.5 *е*) і $\sigma_{2,\,m}$ (рис. 2.5 *з*) є малоймовірними, тому ними можна знехтувати.

У цьому разі з метою спрощення подальших перетворень візьмемо

$$\alpha_m^{(k)} = \alpha_{1,\,m}^{(k)} = \alpha_{2,\,m}^{(k)},\ \beta_m^{(k)} = \beta_{1,\,m}^{(k)},\ \sigma_m^{(k)} = \sigma_{1,\,m}^{(k)}. \qquad (3.2.14)$$

Із урахуванням припущень (3.2.14) за трьохальтернативних рішень повні ймовірності $P_{t,\,m}$ і $P_{f,\,m}$ відповідно дорівнюють

$$P_{t,m} = p(\mu_1)D_{1,m} + p(\mu_2)D_{2,m} + p(\mu_3)D_{3,m},$$
$$P_{f,m} = p(\mu_1)\alpha_m + p(\mu_2)\beta_m + p(\mu_3)\sigma_m. \qquad (3.2.15)$$

Згідно з принципом Бернуллі-Лапласа під час прийняття рішень за умови апріорної невизначеності доцільно вважати апостеріорні гіпотези рівноймовірними :

$$p(\mu_1) = p(\mu_2) = p(\mu_3) = \frac{1}{3}, \qquad (3.2.16)$$

тобто розглядати найгірший у статистичному розумінні випадок.

Після підстановки виразів (3.2.14) і (3.2.15) у формулу (3.2.8) одержимо

$$E_m^{(k)} = \frac{1}{3}\{[D_{1,\,m} + D_{2,\,m} + D_{3,m}] - [\alpha_m + \beta_m + \sigma_m]\} \times$$
$$\times \log_2 \frac{D_{1,\,m} + D_{2,\,m} + D_{3,m}}{\alpha_m + \beta_m + \sigma_m}. \qquad (3.2.17)$$

Крім того, із урахуванням вищеприйнятих допущень мають місце такі співвідношення між точнісними характеристиками для кожної із трьох груп подій:

$$D_{1,\,m}^{(k)} + 2\alpha_m^{(k)} = 1,$$
$$D_{2,\,m}^{(k)} + \beta_m^{(k)} + \sigma_m^{(k)} = 1,$$
$$D_{3,\,m}^{(k)} + \beta_m^{(k)} + \sigma_m^{(k)} = 1. \qquad (3.2.18)$$

Врахувавши відношення (3.2.18), подамо формулу (3.2.17) у вигляді

$$E_m^{(k)} = \frac{1}{3}\{D_{1,\,m}^{(k)} + 1 - 2[\beta_m^{(k)} + \sigma_m^{(k)}]\} \times$$
$$\times \log_2 \frac{2D_{1,\,m}^{(k)} + 4 - 4[\beta_m^{(k)} + \sigma_m^{(k)}] + 10^{-r}}{1 - D_{1,\,m}^{(k)} + 2[\beta_m^{(k)} + \sigma_m^{(k)}] + 10^{-r}}. \qquad (3.2.19)$$

Для обчислення інформаційного критерію Кульбака (3.2.8) машинного навчання системи керування оцінювання точнісних характеристик трьохальтернативних рішень

подамо у такому вигляді :

$$D_{1,\,m}=\frac{K_{1,\,m}}{n_{\min}},\ \alpha_1=\frac{K_{2,\,m}}{n_{\min}},\quad \alpha_2=\frac{K_{3,\,m}}{n_{\min}},$$

$$D_{2,m}=\frac{K_{4,\,m}}{n_{\min}},\ \beta_{1,m}=\frac{K_{5,\,m}}{n_{\min}},\ \beta_{2,m}=\frac{K_{6,\,m}}{n_{\min}},$$

$$D_{3,m}=\frac{K_{7,\,m}}{n_{\min}},\ \sigma_{1,m}=\frac{K_{8,\,m}}{n_{\min}},\ \sigma_{2,m}=\frac{K_{9,\,m}}{n_{\min}},\quad (3.2.20)$$

де $K_{1,\,m}$ – кількість подій, які полягають в належності реалізації, що розпізнається, своєму класу X_m^o, що характеризує функціональний стан «Норма» (рис. 3.6 *а*); $K_{2,m}$ – кількість подій, які полягають в належності реалізації класу «Норма» класу «Менше норми» (рис. 3.6 *б*); $K_{3,\,m}$ – кількість подій, що полягають в належності реалізації класу «Норма» класу «Більше норми» (рис. 3.6 *в*); $K_{4,\,m}$ – кількість подій, що полягають в належності реалізації класу «Менше норми» своєму класу (рис. 3.6 *г*); $K_{5,\,m}$ – кількість подій, що полягають в належності реалізації класу «Менше норми» класу «Норма» (рис. 3.6 *д*); $K_{6,\,m}$ – кількість подій, що полягають в належності реалізації класу «Менше норми» класу «Більше норми» (рис. 3.6 *е*); $K_{7,m}$ – кількість подій, що полягають в належності реалізації класу «Більше норми» своєму класу (рис. 3.6*є*); $K_{8,\,m}$ – кількість подій, що полягають в належності реалізації класу «Більше норми»

класу «Норма» (рис. 3.6 *ж*); $K_{9,\,m}$ – кількість подій, що полягають в належності реалізації класу «Більше норми» класу «Менше норми» (рис. 3.6 *з*); $n_{\min}$ – мінімальний обсяг репрезентативної навчальної вибірки .

Процедура обчислення коефіцієнтів $K_{1,\,m}$ – $K_{9,\,m}$ у рамках IEI-технології залежить від типу вирішальних правил. Розглянемо вирішальні правила, побудовані в просторі ознак при відновленні в процесі навчання гіперсферичних контейнерів класів розпізнавання. Нехай вхідними даними є : $x^{(j)}$, $j=\overline{1,n}$ – реалізація образу, що розпізнається; X_1^o – клас розпізнавання, який характеризує функціональний стан системи керування «Норма»; X_2^o – клас – «Менше норми»; X_3^o – клас – «Більше норми»; x_1, x_2 і x_3 – еталонні (усереднені) вектори класів X_1^o, X_2^o і X_3^o відповідно.

З урахуванням оцінювань відповідних точнісних характеристик (3.2.20) формула (3.2.19) набирає робочого вигляду, придатного для обчислення критерію Кульбака в процесі трьохальтернативного машинного навчання системи керування:

$$E_m = \frac{1}{3n_{\min}}\{K_{1,\,m} + n_{\min} - 2[K_{5,\,m} + K_{8,\,m}]\}\times$$

$$\times \log_2 \frac{2K_{1,\,m} + 4n_{\min} - 4[K_{5,\,m} + K_{8,\,m}] + 10^{-r}}{n_{\min} - K_{1,\,m} + 2[K_{5,\,m} + K_{8,\,m}] + 10^{-r}} \,. \qquad (3.2.21)$$

Процедура обчислення в процесі навчання коефіцієнтів

$K_{1,m}^{(k)}$, $K_{2,m}^{(k)}$ і $K_{3,m}^{(k)}$, що входять до формули (3.2.21) в предикатній формі, має такий вигляд :

$$
\begin{aligned}
1)\ (\forall(x^{(j)} \in X_1^o)\{&If\ \ d(x^{(j)} \oplus x_1) \le d_1[k] \\
&then\ \ K_1[j] := K_1[j-1]+1 \\
&else\ \ if\ \ d(x^{(j)} \oplus x_2) \le d_2[k] \\
&then\ \ K_2[j] := K_2[j-1]+1 \\
&else\ \ if\ \ d(x^{(j)} \oplus x_3) \le d_3[k] \\
&then\ \ K_3[j] := K_3[j-1]+1\};
\end{aligned}
$$

$$
\begin{aligned}
2)\ \forall(x^{(j)} \in X_2^o)\{&If\ \ d(x^{(j)} \oplus x_2) \le d_2[k] \\
&then\ \ K_4[j] := K_4[j-1]+1 \\
&else\ \ if\ \ d(x^{(j)} \oplus x_1) \le d_1[k] \\
&then\ \ K_5[j] := K_5[j-1]+1 \\
&else\ \ if\ \ d(x^{(j)} \oplus x_3) \le d_3[k] \\
&then\ \ K_6[j] := K_6[j-1]+1\};
\end{aligned}
$$

$$
\begin{aligned}
3)\ \forall(x^{(j)} \in X_3^o)\{&If\ \ d(x^{(j)} \oplus x_3) \le d_3[k] \\
&then\ \ K_7[j] := K_7[j-1]+1 \\
&else\ \ if\ \ d(x^{(j)} \oplus x_1) \le d_1[k] \\
&then\ \ K_8[j] := K_8[j-1]+1 \\
&else\ \ if\ \ d(x^{(j)} \oplus x_3) \le d_2[k] \\
&then\ \ K_9[j] := K_9[j-1]+1\}.
\end{aligned}
$$

Критерій (3.2.21) є ненормованим і залежить як від значень коефіцієнтів $K_{1,\,m}$, $K_{5,\,m}$ і $K_{8,\,m}$, так і від обсягу навчальної вибірки $n_{\min}$ і показника степеня r.

Нормовану модифікацію критерію Кульбака для оцінювання функціональної ефективності машинного навчання системи керування за трьохальтернативною системою оцінювання рішень подамо у вигляді

$$\hat{E}_m = \frac{E_m}{E_{m,\,\max}}, \qquad (3.2.22)$$

де E_m – КФЕ, що обчислюється за формулою (3.2.21); $E_{m,\,\max}$ – максимальне значення критерію (3.2.21), що обчислюється за значень $K_{1,\,m} = n_{\min}$ $(D_{1,\,m}^{(k)} = 1)$ і $K_{5,\,m} = K_{8,\,m} = 0$ $(\beta_m^{(k)} = \sigma_m^{(k)} = 0)$.

Таким чином, оптимізація параметрів навчання системи керування інформаційно-телекомунікаційним сервісом може здійснюватися як за двохальтернативною системою оцінювання рішень, що приймаються, так і за трьохальтернативною. Перевагою двохальтернативної системи оцінювання рішень є менша обчислювальна трудомісткість порівняно з трьохальтернативною. Але якщо на практиці використання двохальтернативних рішень не дозволяє побудувати безпомилкові за навчальною матрицею вирішальні правила, то в цьому разі доцільно перейти до трьохальтернативних рішень. Наведені вище модифікації як ентропій-

ного критерію, так й інформаційної міри Кульбака на практиці, як правило, дають однакові оптимальні значення параметрів, оскільки вони повною мірою відповідають вимогам до інформаційних мір.

3.3. Оптимізація керування інфокомунікаційною системою за узагальненим критерієм ефективності

Сучасний підхід до синтезу системи керування інфокомунікаційним сервісом і відповідною ІТ-інфраструктурою передбачає розв’язання задачі мінімізації енерговитрат при забезпеченні заданої якості обслуговування користувачів.

Інфокомунікаційні системи характеризуються такими показниками, як доступність, надійність, продуктивність, конфіденційність, масштабованість, вартість сервісів, прибутковість, час обслуговування, функціональна ефективність, якість обслуговування тощо. Ці критерії можна розглядати як однопараметричні (частинні) критерії ефективності, які не дають достатньо повного уявлення про ефективність системи в цілому. Спроба вибору кращого варіанта проектування системи одночасно за декількома частинними критеріями зазвичай позбавлена змісту, оскільки, як правило, поліпшення одного критерію супроводжується погіршенням принаймні одного іншого критерію. Все це свідчить про необхідність використання узагальнених критеріїв, що пов'язують у необхідних пропорціях основні, найважливіші частинні критерії ефективності системи.

При проектуванні систем керування інфокомунікаційними сервісами та відповідною ІТ-інфраструктурою важ-

ливо не тільки забезпечити необхідні технічні характеристики, але і врахувати затрати на їх одержання. Цій вимозі відповідають узагальнені критерії ефективності, найбільш поширеним серед яких є критерій вигляду [52]

$$Q = (ефект)/(витрати).$$

Цей критерій ефективності можна розглядати як основу синтезу нових критеріїв ефективності. При цьому якщо оцінювати технічну ефективність лише з точки зору отримуваного прибутку, то загальну ефективність системи керування можна подати у вигляді [59]

$$Q_c = (C_v - C_s)/C_{vi},$$

де C_v – результат використання системи (реальний дохід); C_s – витрати на створення та експлуатацію системи; C_{vi} – результат застосування системи при виконанні всіх функцій і за відсутності витрат на їх здійснення (ідеальний випадок).

До задач, що виконує система керування інформаційно-телекомунікаційною інфраструктурою, відносять виконання контролю працездатності розподіленої мережевої інфраструктури, спостереження за станом серверів і систем зберігання, автоматизації рутинних операцій керування центрами оброблення даних, забезпечення доступності та належної продуктивності бізнес-додатків, які звертаються до хмарних ресурсів і використовують мобільні інтерфей-

си та інше. При цьому ефективність керування визначається багатьма факторами, найбільш негативними з яких є апріорна невизначеність поведінки вузлів та додатків на гетерогенних ресурсах, непередбачувана динаміка попиту, пропозиції та доступності ресурсів. З цією метою у великих інформаційно-телекомунікаційних системах використовуються різні засоби моніторингу та автоматизації керування, які генерують та накопичують великі обсяги даних, що містять цінну інформацію про можливості підвищення ефективності функціонування.

Під час функціонування інформаційно-телекомунікаційної системи практично неперервно і в дуже великій кількості генеруються дані лог-файлів, різноманітні метрики стану компонентів інфраструктури, інформація про події та інша телеметрія від засобів моніторингу та керування серверами, системами зберігання, мережевою та мобільною інфраструктурами і додатками. Виникає потреба у розвиненому аналітичному інструментарії, що дозволяє агрегувати дані систем керування ІТ-інфраструктурою в різних доменах, виявляти залежності та кореляції, передбачати перебої і спад продуктивності, надавати інформацію для оперативного і найбільш ефективного вирішення проблем прогнозування потреб сервісів у різних ресурсах. Тому система керування повинна містити у своєму складі систему підтримки прийняття рішень (СППР) для розв’язання задач відновлення, забезпечення продуктивності й розвитку інформаційно-телекомунікаційної інфраструктури. При цьому інформаційна спроможність СППР, яка характеризує її функціональну ефективність, є основ-

ною складовою загальної ефективності системи керування.

Згідно з працею [52] узагальнену ефективність системи керування можна визначити її двома складовими: інформаційною спроможністю системи та зведеною вартістю створення, експлуатації, зберігання та ліквідації системи. При цьому узагальнений функціонально-статистичний критерій ефективності І. В. Кузьміна має вигляд

$$E_{I,C} = K_I / K_{I0}, \qquad (3.3.1)$$

де K_I – узагальнена функціонально-статистична характеристика реальної системи:

$$K_I = I_{\max} / C, \qquad (3.3.2)$$

де $I_{\max}$ – максимальна інформаційна спроможність системи; C – зведені витрати на створення, експлуатацію та ліквідацію системи.

Узагальнена функціонально-статистична характеристика потенційної (ідеальної) системи визначається як

$$K_{I0} = I^0_{\max} / C_{\min}, \qquad (3.3.3)$$

де $I^0_{\max}$ – максимальна інформаційна спроможність потенційної системи; $C_{\min}$ – зведені витрати для потенційної системи.

Максимізація інформаційної спроможності системи керування передбачає зняття невизначеності щодо задово-

леності користувачів рівнем сервісу та функціонального стану компонентів або сервісів інформаційно-телекомунікаційного середовища. Вирішальні правила системи підтримки рішень можна одержати в процесі машинного навчання за архівними даними моніторингу та суб'єктивно-статистичних досліджень оцінювання якості сервісів. При цьому процес навчання системи полягає у пошуку оптимальних значень координат вектора просторово-часових параметрів функціонування, що забезпечують максимальне значення узагальненого критерію ефективності, який з урахуванням (3.1.3) можна подати у вигляді

$$J = \frac{\overline{E}}{E_{\max}} \cdot \frac{C_{\min}}{C_{training} + C_{error}}, \qquad (3.3.4)$$

де $\overline{E}$ – усереднене за алфавітом класів розпізнавання $\{X_m^o \mid m = \overline{1, M}\}$ значення інформаційного КФЕ машинного навчання системи; $E_{\max}$ – максимальне граничне значення КФЕ; $C_{\min}$ – мінімальне граничне значення витрат оператора/провайдера, пов'язаних із експлуатацією системи керування; $C_{training}$ – значення затрат на експлуатацію системи, зокрема витрати на формування вхідного математичного опису та вартість системних ресурсів, задіяних під час навчання (перенавчання); C_{error} – розраховані за

матрицею штрафів втрати оператора/провайдера інформаційно-телекомунікаційної системи, пов'язані з неоптимальним керуванням IT-інфраструктурою внаслідок помилкового прийняття рішень.

Склад алфавіту класів розпізнавання $\{X_m^o \mid m = \overline{1, M}\}$ залежить від кола розв'язуваних задач. У задачах, що відносять до аналітики доступності (availability analytics) [57], класами розпізнавання можуть бути шаблони використання чи поведінки компонентів інфраструктури або сервісів. У задачах проактивного аналізу (performance analytics) [50, 51, 60] інфокомунікаційного середовища класами розпізнавання можуть бути функціональні стани та ситуації, що передують відмовам та збоям. У задачах аналітики вивчення поведінки (behavior learning analytics) [30, 60] система навчається розпізнавати класи нормальної та аномальної поведінки сервісу, що дозволяє згенерувати сигнал під час виникнення можливих відхилень від норми. Розпізнавання відхилення від нормального функціонування є особливо цінним інструментом у процесах керування змінами, надаючи можливість на підставі метрик продуктивності, аналізу конфігурацій та моделі сервісу швидко оцінити наслідки внесення змін до компонентів інфраструктури для сервісу в цілому. У задачах аналітики потужностей (capacity analytics) [42, 45, 60] як класи розпізнавання можуть розглядатися тренди потреб у ресурсах. У задачах прогнозування сприйняття якості (Quality of Experience analytics) [9,10] інформаційно-телекомунікаційних сервісів класи розпізнавання характеризують рівень якості, що

сприймається користувачами.

У робочому режимі навчена система керування повинна спрогнозувати функціональний стан сервісів і компонентів інформаційно-телекомунікаційного середовища, що прямо впливають на додержання SLA-угод, за різних варіантів використання (розподілу) його ресурсів. При цьому завдання планування використання ресурсів є багатокритеріальною, оскільки необхідно одночасно забезпечити мінімум енергоспоживання, обсягу невикористаних ресурсів та порушень SLA-угод. Однак ці частинні критерії є попарно суперечливими, мають різну розмірність та є нелінійними функціями контрольованих характеристик і конфігурацій інформаційно-телекомунікаційної системи. Тому необхідно розглянути можливість зведення (згортки) вектора перелічених частинних критеріїв до одного комплексного критерію. Найбільш загальний підхід до згортки векторних критеріїв заснований на використанні принципу зваженої суми частинних критеріїв [61]. Розглянемо основні кроки реалізації згортки векторного критерію.

Крок 1. Нормалізація та масштабування вектора частинних критеріїв $\{k_i \mid i = \overline{1, I}\}$ за правилом

$$k_i = \begin{cases} 0, k_i \le k_i^{\min}, \\ \dfrac{(k_i - k_i^{\min})}{(k_i^{\max} - k_i)}, k_i^{\min} < k_i < k_i^{\max} \\ 1, k_i > k_i^{\max}, \end{cases}$$

де $k_i^{\min}$, $k_i^{\max}$ – відповідно нижня та верхня межі допустимої області значень *i*-го частинного критерію.

Крок 2. Визначення за допомогою певних формалізованих процедур або із застосуванням експертних оцінювань пріоритетів частинних критеріїв у вигляді вектора коефіцієнтів важливості $\{\lambda_i \mid i = \overline{1, I}\}$ кожного частинного критерію, для якого виконується умова

$$\sum_{i=1}^{I} \lambda_i = 1.$$

Крок 3. Обчислення зваженої суми частинних критеріїв за формулою

$$K = \sum_{i=1}^{I} \lambda_i k_i. \qquad (3.3.5)$$

Для обчислення частинних критеріїв, пов'язаних з недотриманням SLA-угод, використовують вирішальні правила, одержані в процесі навчання системи керування. За умови високої достовірності прогностичних вирішальних правил результат мінімізації зваженої суми частинних критеріїв приводитиме до задоволення вимог клієнтів та власників інформаційно-телекомунікаційної системи.

Таким чином, використання аналітичного інструменту в складі системи керування інформаційно-телекомунікаційним середовищем надає можливості оптимізації використання ресурсів інфраструктури в рамках реалізації окремих сервісів, планування потреб у ресурсах, уникнен-

ня відмов та збоїв, підтримки заданого рівня задоволеності користувачів, оцінювання ймовірності успіху внесення різноманітних змін у середовище. Ефективність аналітичного інструменту визначає ефективність системи керування в цілому. Тому основною складовою узагальненого критерію ефективності системи керування є інформаційний критерій якості її машинного навчання. При цьому вартісна складова узагальненого критерію є допоміжною. Вона слугує для контролю за доцільністю матеріальних витрат на подальше навчання (перенавчання) системи.

3.4. Методи автоматичної класифікації

Завдання класифікації полягає у визначенні груп подібних об'єктів. При цьому для формалізації поняття «подібність» вводиться функція відстані або метрика $d(x,y)$ в N-вимірному просторі ознак розпізнавання. Алгоритми, побудовані на аналізі подібності об'єктів, часто називають метричними.

Найбільш простим метричним алгоритмом є алгоритм найближчого сусіда (nearest neighbor, NN), який класифікує об'єкт як реалізацію того класу, якому належить найближчий об'єкт навчальної вибірки [63]. Навчання такого класифікатора зводиться до елементарного запам'ятовування навчальної вибірки $\{y_{m,i}^{(j)} \mid m=\overline{1,M};\ j=\overline{1,n};\ i=\overline{1,N}\}$, де M – потужність алфавіту класів розпізнавання; n – обсяг спостережень класу X_m^o; N – кількість ознак розпізнавання. Єдиною перевагою цього алго-

ритму є простота реалізація. Однак недоліків набагато більше. По-перше, наявність викидів у навчальній вибірці призводить до нестійкості та похибок. По-друге, відсутні параметри, які можна б було настроювати за навчальною вибіркою. Алгоритм повністю залежить від успішності вибору дистанційної міри $d(x, y)$.

Алгоритм *к*-найближчих сусідів (*k*-nearest neighbors), створений із метою згладжування шумового впливу викидів, класифікує об'єкти шляхом голосування за одного із *к* найближчих сусідів [63]. При цьому кожен із сусідів $\{y^{(j)} \mid j = \overline{1,k}\}$ голосує за віднесення об'єктів до свого класу. Алгоритм відносить вхідний об'єкт до того класу, який набере більшу кількість голосів. Оптимальне значення параметра k визначається за критерієм ковзного контролю з виключенням об'єктів по одному (leave-one-out).

Міри близькості обирають виходячи з властивостей об'єктів. Квадрат евклідової відстані застосовується для надання більшої ваги найвіддаленішим один від одного об'єктам :

$$d(x, y) = \sum_{i=1}^{N} (x_i - y_i)^2 .$$

Степенева відстань застосовується у тому разі, коли необхідно збільшити або зменшити вагу об'єктів, евклідові відстані між якими суттєво відрізняються. Степенева відстань розраховується за формулою

$$d(x,y) = \sqrt[r]{\sum_{i=1}^{N} (x_i - y_i)^p} ,$$

де r і p – параметри, визначені користувачем.

Параметр r відповідальний за поступове зважування різниць за окремими координатами, параметр p відповідальний за прогресивне зважування великих відстаней між об'єктами. Якщо r і p дорівнюють двом, то ця відстань збігається з відстанню Евкліда.

Відстань Журавльова обчислюється за формулою

$$d(x,y) = \sum_{i=1}^{N} d_i ,$$

де $d_i = \begin{cases} 1, \textit{якщо} \left| x_i - y_i \right| < \varepsilon, \\ 0, \textit{інакше} \end{cases}$.

При цьому ε – додатне як завгодно мале дійсне число (критерій мінімальної відстані між точками).

Міра близькості підбирається індивідуально для конкретних типів даних. Інколи адекватної міри близькості підібрати не вдається і доводиться обирати її евристично.

У практичних задачах контролю інфокомунікаційних систем вектор ознак функціонального стану складається як з неперервних (кількісних), так і дискретних (категоріальних) ознак, що набувають своїх значень з кінцевої неупорядкованої множини. З даними в межах номінальної шка-

ли, в якій вимірюються категоріальні ознаки, не можуть бути здійснені будь-які арифметичні операції, оскільки усі види числового оброблення стосуються упорядкування об'єктів у кожному класі. Зведення категоріальних первинних ознак до кількісних вторинних шляхом простої нумерації значень первинних ознак рідко призводить до задовільних результатів, оскільки алгоритми будуть враховувати впорядкованість, яка не має змісту. Тому оброблення даних змішаного типу на даний час викликає значні труднощі та є перспективною областю досліджень [64–66].

У працях [64, 65] було запропоновано використання Dummy-кодування категоріальних ознак, при якому кожна первинна ознака перекодується в декілька вторинних бінарних ознак з рівно однією одиницею. Подібне перекодування дозволяє застосовувати багато класичних алгоритмів навчання, однак це сильно збільшує розмірність простору ознак, а також накладає обмеження на структуру ознакового опису спостережень. Інший підхід пов’язаний із застосуванням в алгоритмі машинного навчання метрики перекриття (відстань за Хеммінгом), при використанні якої ступінь відмінності реалізацій образів визначається кількістю незбіжних значень первинних ознак. При цьому оброблення змішаних даних може здійснюватися шляхом використання гетерогенної дистанційної метрики (Heterogeneous Euclidean-Overlap Metric) [65]. Гетерогенна дистанційна метрика між двома векторами-реалізаціями $\{x_i \mid i = 1, N\}$ та $\{y_i \mid i = 1, N\}$ обчислюється за формулою

$$d(x,y)=\sqrt{\sum_{i=1}^{N} d_i^{\,2}}\,. \qquad (3.4.1)$$

Функція d_i у формулі (3.4.1) визначає відстань між двома значеннями i-ї ознаки у векторах x та y і має такий вигляд :

$$d_i=\begin{cases}1, \textit{ якщо } x \textit{ або } y \textit{ невідомі};\\ overlap_i(x_i,y_i)\ , \textit{ якщо } i-\textit{та ознака}\\ \qquad\qquad\qquad \textit{є категоріальною};\\ diff_i(x_i,y_i)\ , \textit{ інакше.}\end{cases}$$

За наявності пропущених значень ознак відстань між двома ознаками визначається як максимальна, тобто дорівнює одиниці. Врахування можливості пропущених значень в даних є досить важливим на практиці, оскільки неповнота та зашумленість вхідних даних часто обмежує застосування алгоритмів аналізу даних. Функція перекриття для окремої категоріальної ознаки має вигляд

$$overlap_i(x_i,y_i)=\begin{cases}0\ , \textit{ якщо } x_i=y_i,\\ 1\ , \textit{ інакше.}\end{cases}$$

Нормалізована відстань між кількісними ознаками задається формулою

$$diff_i(x_i, y_i) = \frac{|x_i - y_i|}{\max_i - \min_i},$$

де $\max_i$, $\min_i$ – відповідно мінімальне та максимальне можливі значення, що набуває i-та ознака.

Недоліком алгоритмів, що використовують подібну дистанційну метрику, є ігнорування додаткової інформації, що описується статистикою якісних ознак. Наприклад, для категоріальної ознаки можна обчислити її частоту (кількість спостережень) та моду (значення, яке має найбільшу частоту).

Метричні алгоритми здійснюють локальну апроксимацію вибірки, при якій обчислення відкладаються до моменту, поки не стане відомим вхідний об'єкт. Тому метричні алгоритми відносять до методів лінивого навчання (lazy learning). Основи методів наполегливого навчання (eager learning), що здійснюють глобальну апроксимацію вибірки, було закладено в теорії багатовимірного статистичного аналізу та теорії прийняття рішень [67].

Суть статистичних методів навчання полягає у відновленні роздільної функції шляхом мінімізації середнього ризику помилкового прийняття рішення. Статистичні методи дозволяють побудувати вирішальні правила у разі перетину класів розпізнавання, що має місце в практичних задачах контролю та керування слабоформалізованими процесами.

Одним із класичних статистичних методів класифікації є метод Байєса [63, 67], відповідно до якого прийняття

класифікаційних рішень здійснюється шляхом знаходження максимальної апостеріорної умовної ймовірності $p(X_m^o / x)$, обчисленої для заданого алфавіту класів розпізнавання $\{X_m^o \mid m = \overline{1, M}\}$ за формулою

$$p(X_m^o / x) = \frac{P(X_m^o) p(x / X_m^o)}{\sum_{k=1}^{M} P(X_k^o) p(x / X_k^o)},$$

де $P(X_m^o)$ – безумовна ймовірність появи класу X_m^o; $p(x / X_m^o)$ – значення функції правдоподібності (щільності розподілу ймовірностей) класу X_m^o для вхідної реалізації x.

Безумовна ймовірність появи класу X_m^o визначається як відношення кількості реалізацій, що належать класу X_m^o, до загальної кількості реалізацій :

$$P(X_m^o) = \frac{count(x^{(j)} \in X_m^o)}{n},$$

де $count(x^{(j)} \in X_m^o)$ – кількість реалізацій навчальної вибірки, що належать класу X_m^o; n – загальна кількість реалізацій образів у навчальній вибірці.

Значення функції правдоподібності класу X_m^o для реалізації x при статистичній незалежності ознак розпізнавання обчислюється за формулою

$$p(x / X_m^o) = \prod_{i=1}^{N} p(x_i / X_m^o),$$

де $p(x_i / X_m^o)$ – значення щільності розподілу ймовірностей i-ї ознаки у вхідній реалізації класу X_m^o.

Щільності розподілу ймовірностей $p(x_i / X_m^o)$ можуть бути оцінені в рамках припущення про тип розподілу. Наприклад, може використовуватися гіпотеза про нормальний закон розподілу

$$p(x_i / X_m^o) = \frac{1}{\sigma_{m,i}\sqrt{2\pi}} \exp\left(-\frac{\left(x_i - \bar{x}_{m,i}\right)^2}{2\sigma_{m,i}^2} \right),$$

де $\sigma_{m,i}^2$ – дисперсія i-ї ознаки у векторі-реалізації класу X_m^o; $\bar{x}_{m,i}$ – математичне очікування i-ї ознаки в реалізації класу X_m^o.

Перевагою баєсівських класифікаторів є простота реалізації алгоритмів класифікації, а недоліком – низька достовірність класифікації реалізацій образів у разі

обмеженого обсягу вибірок та перетину в просторі ознак класів розпізнавання. Основними недоліками статистичних методів, які обмежують їх використання на практиці, є необхідність великих обсягів статистики для апроксимації функції щільності розподілу ймовірностей, виконання жорстких умов для забезпечення статистичних стійкості та однорідності та висока чутливість до репрезентативності навчальних вибірок.

У рамках геометричношго підходу набрав популярності метод опорних векторів SVM (support vector machine) [63]. В основу SVM покладена ідея розділення згущення векторів гіперплощинами, що знаходяться на максимальній відстані від згущень при мінімізації зміщення реалізацій класу від опорного вектора. Задачу синтезу вирішальних правил за SVM для двокласової класифікації пояснює рисунок 3.7.

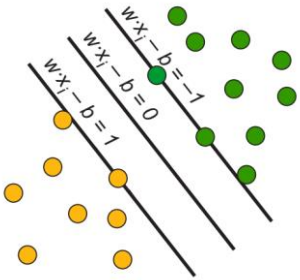


Рисунок 3.7 – Лінійне розділення образів у SVM

Проблема лінійної нероздільності в методі SVM вирішується шляхом переходу від початкового простору ознак

до нового розширеного простору за допомогою певного перетворення. На рисунку 3.8 показано приклад переходу від двовимірного простору до розширеного тривимірного шляхом перенесення точок на поверхню сфери.

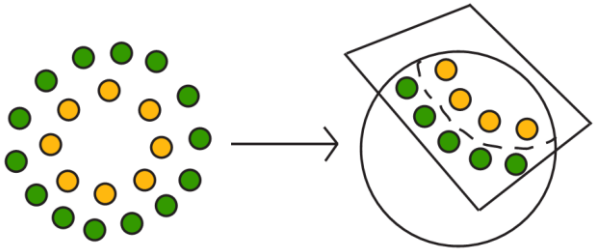


Рисунок 3.8 – Трансформація простору ознак у SVM

Аналіз рисунка 3.8 показує, що в новому просторі більшої розмірності вдалося розділити точки вибірок двох класів площиною.

Вирішальне правило двокласового SVM-класифікатора має вигляд

$$a(x) = sign\left[\sum_{j}^{n} \lambda_j c_j K(x, x^{(j)}) - \rho\right], \qquad (3.4.2)$$

де ρ – параметр порогу; λ – вектор коефіцієнтів, ненульові значення компонентів якого відповідають опорним векторам навчальної вибірки; c_j – позначка класу j-го вектора навчальної вибірки, яка набуває значення із мно-

жини $\{-1,+1\}$; $K(x, x^{(j)})$ – функція ядра, що відповідає скалярному добутку векторів у деякому розширеному просторі ознак :

$$K(x, x^{(j)}) = \varphi(x)^T \varphi(x^{(j)}),$$

де $\varphi(x^{(j)})$ – функція відображення вектора $x^{(j)}$ у розширений простір, де забезпечується лінійна роздільність класів розпізнавання.

Навчання за методом опорних векторів зводиться до пошуку коефіцієнтів λ_j (множники Лагранжа) та порогу ρ, знайти які можна, вирішивши таку задачу квадратичної оптимізації :

$$L(\lambda) = \sum_{j=1}^{n} \lambda_j - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j c_i c_j K(x_i, x_j) \to \max_{\lambda} \quad (3.4.3)$$

за умови

$$\begin{cases} \sum_{j=1}^{n} \lambda_j c_j = 0; \\ 0 \le \lambda_j \le C, \end{cases}$$

де C – константа регуляризації, яка на практиці обирається такою, що дорівнює

$$C = \frac{1}{nv},$$

де v – максимальна частка векторів навчальної послідовності, які можуть бути викидами.

Задача оптимізації (3.4.3) вирішується евристичними алгоритмами шляхом послідовного зменшення цільової функції.

Для вибору функції ядра використовують теорему Мерсера [63] : функція $K(x,x')$ є ядром тоді і тільки тоді, коли вона симетрична, $K(x,x') = K(x',x)$, і невід'ємно визначена. Однак вибір ядра для конкретних вхідних даних досі залишається нетривіальною задачею, а повний перебір усіх можливих ядер обумовлює невисоку оперативність машинного навчання за SVM.

Основним недоліком цього методу, що обмежує його застосування для задач класифікаційного аналізу функціонального стану інфокомунікаційної системи, є модельність алгоритму, обумовлена ігноруванням апріорного перетину класів розпізнавання в просторі ознак.

На усунення недоліків відомих методів машинного навчання спрямовано інформаційно-екстремальну інтелектуальну технологію (ІЕІ-технологію), яка ґрунтується на максимізації інформаційної спроможності системи розпізнавання шляхом введення на етапі навчання додаткових інформаційних обмежень [55–57]. Основні концептуальні положення ІЕІ-технології такі :

- ІЕІ-технологія ґрунтується на прямому оцінюванні

інформаційної спроможності системи, що навчається;

- ІЕІ-технологія дозволяє оптимізувати просторово-часові параметри функціонування системи, що навчається;
- прийняття рішень у рамках ІЕІ-технології здійснюється в рамках детерміновано-статистичного підходу шляхом побудови відносно простого детермінованого класифікатора, статистична корекція якого здійснюється в процесі навчання з метою побудови безпомилкових за навчальними матрицями вирішальних правил;
- методи ІЕІ-технології ґрунтуються на застосуванні гіпотез як чіткої, так і нечіткої компактності реалізацій образу, тобто є працездатним за умови перетину класів роз-пізнавання, що має місце в практичних задачах контролю та керування;
- методи ІЕІ-технології базуються на вибірковому підході математичної статистики і орієнтовані на застосування прийнятних із практичних міркувань обсягів репрезентативних навчальних вибірок.

Побудова «точного» контейнера класу розпізнавання складної геометричної форми у багатовимірному просторі ознак навіть для сучасних комп'ютерних комплексів має суттєві ускладнення. Тому при обґрунтованості гіпотези нечіткої компактності в працях [54, 55] згідно з принципом редукції у рамках детерміновано-статистичного підходу пропонується відновлювати контейнери спрощеної форми, наближеної до «точної» деяким оптимальним способом, і які формують радіально-базисні вирішальні правила, що забезпечують в режимі екзамену, тобто безпосередньо в робочому режимі, достовірність прийняття рішень, близь-

ку до максимальної асимптотичної. Але необхідною умовою прийняття в режимі екзамену високодостовірних рішень є побудова на етапі машинного навчання безпомилкових за навчальною матрицею вирішальних правил.

Згідно з принципом відкладених рішень О. Г. Івахненка у рамках ІЕІ-технології машинне навчання триває до побудови безпомилкових за навчальною матрицею вирішальних правил. Однією із стартових процедур інформаційно-екстремального навчання є оптимізація системи контрольних допусків на ознаки розпізнавання, в процесі якої адаптується вхідний математичний опис до максимальної інформаційної спроможності системи керування.

Визначення 3.4.1. Контрольним називається поле допусків, в якому i-та ознака розпізнавання, яка контролюється, знаходиться з імовірністю $0 < p_i < 1$ за умови, що функціональний стан системи керування характеризується максимальною функціональною ефективністю.

Область значень поля контрольних допусків визначається нормованим полем допусків на відповідну ознаку розпізнавання.

Визначення 3.4.2. Нормованим називається поле допусків, в якому i-та ознака розпізнавання, що

контролюється, знаходиться з імовірністю $p_i = 0$ або $p_i = 1$ за умови, що функціональний стан системи керування характеризується максимальною функціональною ефективністю.

На рисунку 3.9 показано категорійну модель інформаційно-екстремального навчання СППР із оптимізацією геометричних параметрів контейнерів класів розпізнавання, що відновлюються в радіальному базисі простору ознак, і системи контрольних допусків на ознаки розпізнавання.

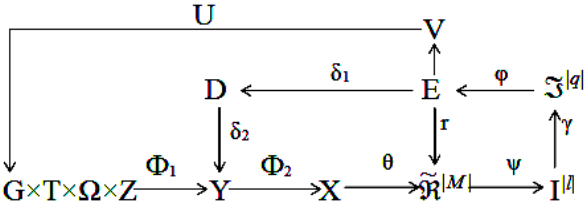


Рисунок 3.9 – Категорійна модель навчання СППР

Категорійна модель містить вхідний математичний опис, який подається у вигляді структури

$$\Delta_B = < G, T, \Omega, Z, Y, X; \Phi_1, \Phi_2 >,$$

де G – простір вхідних сигналів (факторів); T –

множина моментів часу одержання інформації; Ω – простір ознак розпізнавання; Z – простір функціональних станів інформаційно-телекомунікаційної системи; Y – вибіркова множина, що утворює вхідну навчальну матрицю; X – бінарна навчальна матриця; $\Phi_1 : G \times T \times \Omega \times Z \to Y$ – оператор формування вхідної навчальної матриці Y; $\Phi_2 : Y \to X$ – оператор перетворення матриці Y в бінарну матрицю X.

На рисунку 3.9 оператор $\theta : X \to \tilde{\Re}^{|M|}$ будує в загальному випадку нечітке розбиття $\tilde{\Re}^{|M|}$ бінарного простору ознак на класи розпізнавання, а оператор класифікації Ψ перевіряє основну статистичну гіпотезу про належність вхідної реалізації класу X_m^o і таким чином формує множину гіпотез $I^{|l|}$, де l – кількість статистичних гіпотез. Оператор γ шляхом оцінення прийнятих гіпотез формує множину точнісних характеристик $\Im^{|q|}$, де $q = l^2$, а оператор φ обчислює множину значень інформаційного КФЕ, який є функціоналом від точнісних характеристик. Контур моделі, що замикається оператором r, реалізує ітераційний процес оптимізації геометричних параметрів розбиття $\tilde{\Re}^{|M|}$ шляхом пошуку глобального максимуму КФЕ в робочій (допустимій) області визначення його

функції. Оптимізація контрольних допусків на ознаки розпізнавання здійснюється контуром операторів, який замикається через терм-множину D – систему контрольних допусків на ознаки розпізнавання.

Показана на рисунку 3.9 категорійна модель передбачає згідно з принципом відкладених рішень О. Г. Івахненка перехід у разі невисокої функціональної ефективності навчання СППР до більш складних типів радіально-базисних вирішальних правил. З цією метою її зовнішній контур містить множину V типів вирішальних правил, побудованих за різними радіально-базисними роздільними функціями. Процес навчання регламентується оператором $U : V \to G \times T \times \Omega \times Z$.

Таким чином, категорійна модель, показана на рис. 3.9, на відміну від інших теоретико-множинних моделей може розглядатися як узагальнена структура алгоритму інформаційно-екстремального синтезу, здатної навчатися СППР. Крім того, використання категорійних моделей відкриває шлях до застосування сучасних інформаційних інтелектуальних технологій, орієнтованих на прогресивне функціональне програмування.

У рамках ІЕІ-технології під машинним навчанням розуміється процедура оптимізації параметрів навчання за інформаційним критерієм, який характеризує функціональну ефективність системи.

Визначення 3.4.3. Параметром машинного навчання називається характеристика системи, що впливає на функціональну ефективність її навчання.

Серед параметрів інформаційно-екстремального навчання важливе місце займає параметр δ симетричного поля контрольних допусків на ознаку розпізнавання

Геометричний зміст параметра δ показано на рисунку 3.10.

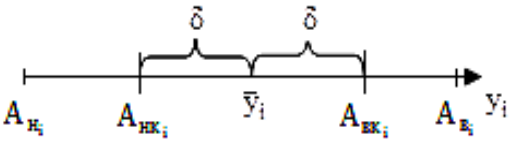


Рисунок 3.10 – Поле контрольних допусків на ознаку розпізнавання

На рисунку 3.10 подано такі позначення: A_{H_i}, A_{B_i} – нижні та верхні нормовані допуски на ознаку y_i відповідно; A_{HK_i}, A_{BK_i} – нижні та верхні контрольні допуски на ознаку y_i відповідно; $\overline{y_i}$ – номінальне (усереднене) значення ознаки y_i; δ – параметр поля контрольних допусків на ознаку розпізнавання.

Таким чином, параметр δ дорівнює половині

симетричного поля контрольних допусків на ознаку розпізнавання.

При цьому нижній A_{HK_i} і верхній A_{BK_i} контрольні допуски для i-ї ознаки розпізнавання обчислюються за формулами :

$$A_{H,i} = \overline{y_{1,i}} - \delta_i ,\ A_{B,i} = \overline{y_{1,i}} + \delta_i$$

де $\overline{y_{1,i}}$ – середнє вибіркове значення i-ї ознаки еталонного вектора-реалізації базового класу X_1^o, відносно якого встановлюється система контрольних допусків для заданого алфавіту $\{X_m^o\}$ класів розпізнавання.

Оптимізацію параметра $\delta = \delta_i, i = \overline{1,N}$ поля контрольних допусків на ознаки розпізнавання розглянемо на прикладі машинного навчання, в процесі якого відновлюються в бінарному просторі ознак гіперсферичні контейнери класів розпізнавання. У цьому разі навчання СППР здійснюється за ітераційною процедурою пошуку глобального максимуму інформаційного КФЕ [55] :

$$\delta^* = \arg\max_{G_\delta} \left\{ \frac{1}{M} \sum_{s=1}^{M} \left[\max_{G_E \cap G_d} E_m \right] \right\}, \qquad (3.4.4)$$

де E_m – критерій ефективності навчання класифікатора розпізнавати реалізації m-го класу; G_δ – область допусти-

мих значень параметра поля контрольних допусків на значення ознак; G_E – допустима область визначення функції критерію; G_d – область допустимих значень радіуса гіперсферичного контейнера класу X_m^o.

Розглянемо основні кроки реалізації, вкладеної в алгоритм (3.4.4) процедури, яка здійснює оптимізацію радіусів гіперсферичних контейнерів при заданій системі контрольних допусків на ознаки розпізнавання.

Крок 1. Формування масиву еталонних двійкових векторів $\{x_{m,i} \mid m = \overline{1,M},\ i = \overline{1,N}\}$, елементи яких визначаються за правилом

$$x_{m,i} = \begin{cases} 1, & if \quad \dfrac{1}{n}\sum_{j=1}^{n} x_{m,i}^{(j)} > \rho, \\ 0, & if \quad else, \end{cases}$$

де ρ – рівень селекції координат двійкових еталонних векторів класів розпізнавання, який за замовчуванням дорівнює $\rho = 0{,}5$.

Крок 2. Розбиття множини еталонних векторів на пари найближчих «сусідів» : $\Re_m^{|2|} = \langle x_m, x_c \rangle$, де x_c – еталонний вектор сусіднього класу X_c^o, здійснюється за схемою:

1) структурується множина еталонних векторів, почи-

наючи з вектора x_1 базового класу X_l^o, що характеризує найбільш бажаний функціональний стан;

2) будується матриця кодових відстаней між еталонними векторами розмірності $M \times M$;

3) для кожного рядка матриці кодових відстаней знаходиться мінімальний елемент, який належить стовпчику вектора, найближчого до вектора, що визначає рядок. За наявності декількох однакових мінімальних елементів вибирається з них будь-який, оскільки вони є рівноправними;

4) формується структурована множина елементів попарного розбиття $\{\Re_m^{|2|} \mid m = \overline{1, M}\}$, яка задає план навчання.

Крок 3. Оптимізація кодової відстані d_m може відбуватися як за прямою процедурою перебору

$$d_m(k) = [d_m(k-1) + h \mid d_m(k) \in G_m^d],$$

так і за іншими схемами пошуку глобального максимуму КФЕ в робочій (допустимій) області визначення його функції (дихотомічні методи, методи випадкового пошуку, популяційні методи тощо). При цьому береться $E_m(0) = 0$.

Крок 5. Процедура закінчується при знаходженні максимального КФЕ навчання в робочій області визначення його функції та визначення оптимального радіуса контейнера класу X_m^o:

$$d_m^* = \arg \max_{G_E \cap \{d\}} E_m,$$

де на радіуси контейнерів класів розпізнавання накладається обмеження $\{d\} = \{d_1, ..., d_{\max}\} \in [0; d(x_m \oplus x_c) - 1]$, тобто контейнери не можуть поглинати ядро сусіднього класу X_c^o, центром якого є двійковий еталонний (усереднений) вектор x_c.

Як КФЕ інформаційно-екстремального машинного навчання системи керування можуть розглядатися різні інформаційні міри, наприклад модифікації ентропійного критерію (3.2.1, 3.2.7) або міри Кульбака (3.2.10, 3.2.12).

У режимі екзамену рішення про належність вектора-реалізації $x^{(j)}$ одному з класів алфавіту $\{X_m^o\}$ приймається шляхом обчислення геометричної функції належності, на основі якої будуються в просторі ознак розпізнавання вирішальні правила. Для гіперсферичного класифікатора функція належності має простий вигляд [56] :

$$\mu_m = 1 - \frac{d(x_m^* \oplus x^{(j)})}{d_m^*}, \qquad (3.4.5)$$

де $d(x_m^* \oplus x^{(j)})$ – кодова відстань від центра контейнера класу X_m^o до вектора $x^{(j)}$, що розпізнається.

Клас, до якого належить вектор, що розпізнається, визначається за максимальним значенням функції (3.4.5).

При синтезі вирішальних правил на основі методів машинного навчання часто необхідно використовувати на-

вчальні вибірки великого розміру, що обумовлює потребу у використанні великих обсягів оперативної пам'яті. Крім того, дані, одержані в процесі моніторингу компонентів інфокомунікаційного середовища, як правило, характеризуються значною незбалансованістю класів. У такій ситуації використання стандартних методів класифікації призводить до збільшення помилки класифікації аномального функціонального стану порівняно з класифікацією нормального стану. Одним із простих способів урахування незбалансованості є збільшення «премій» за правильні класифікації реалізацій міноритарних класів [65]. Такий підхід дозволяє підвищити чутливість до рідкісних подій, але одночасно призводить до зменшення сумарної точності розпізнавання. У зв'язку з цим останнім часом набули популярності методи семплювання [68, 69], суть яких полягає у балансуванні обсягу вибірок класів шляхом генерації нових чи видалення дублювальних (неінформативних) векторів-реалізацій образів (рис. 3.11).

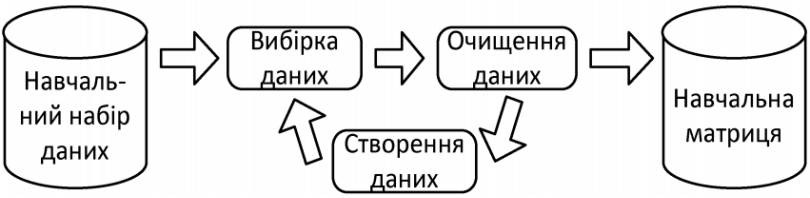


Рисунок 3.11 – Схема балансування класів методами семплювання

Базовими методами семплювання є undersampling та oversampling, які можуть використовуватися як окремо, так

і комбіновано в одній ітераційній процедурі формування навчальної вибірки [68]. Застосування undersampling дозволяє збалансувати кількість реалізацій різних класів випадковою вибіркою з мажоритарного класу, однак це може призвести до втрати важливої інформації. Методи oversampling штучно збільшують кількість реалізацій міноритарного класу шляхом дублювання чи шляхом їх генерації за допомогою малих відхилень від реальних реалізацій міноритарного класу.

У задачах оброблення великих даних, зокрема незбалансованих, дістали поширення методи паралельної (bagging) і послідовної (boosting) композиції класифікаторів, що навчаються на різних підвибірках [70].

В алгоритмі Bagging вхідні дані випадковим чином розбиваються на одинакові за розміром підмножини, кожна з яких використовується для навчання одного базового класифікатора. При цьому кожен вектор-реалізація із вхідного набору даних має однакову ймовірність бути обраним для навчання базових класифікаторів. Прогноз паралельної композиції з K класифікаторів визначається більшістю голосів або їх усередненим значенням.

В алгоритмі Boosting базові класифікатори навчаються послідовно, а набір даних, на яких навчається кожен наступний базовий алгоритм, залежить від точності прогнозування попереднього базового класифікатора (рис. 3.12).

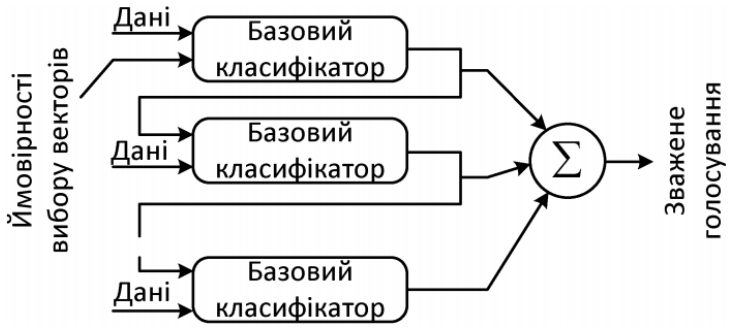


Рисунок 3.12 – Схема послідовної композиції класифікаторів

Важливим поняттям у Boosting є вага вектора-реалізації в наборі даних, яка оновлюється на кожному кроці навчання базового класифікатора). Значення ваги вектора-реалізації описує її важливість для навчання наступного базового класифікатора і обчислюється на основі помилки прогнозу попереднього класифікатора на цьому векторі-реалізації. Вагу вектора-реалізації можна інтерпретувати як імовірність її вибору для навчання наступного класифікатора. Прогноз послідовної композиції класифікаторів здійснюється шляхом зваженого голосування :

$$a(x) = round\left(\sum_{k=1}^{K} \omega_k a_k(x)\right),$$

де $round$ – функція округлення до найближчого цілого числа; ω_k – нормований ваговий коефіцієнт k-го базового класифікатора, для якого виконується умова

$$\sum_{k=1}^{K} \omega_k = 1.$$

Розмір одержуваної композиції класифікаторів залежить від використовуваного методу машинного навчання та особливостей вхідних даних.

У багатьох випадках функціонування класифікатора відбувається за умов невизначеності, що виникає через перекриття границь класів внаслідок складної конфігурації багатомодального розподілу векторів-реалізацій класів у просторі ознак (рис. 3.13).

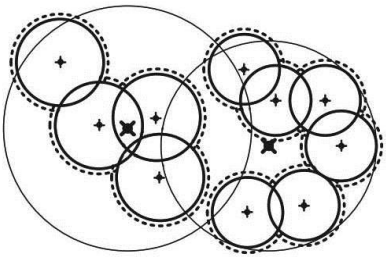


Рисунок 3.13 – Перетин гіперсферичних роздільних гіперповерхонь двох багатомодальних класів

Як показано на рисунку 3.13, для підвищення точності класифікації за умов нерівномірного та багатомодального розподілу векторів-реалізацій у класах варто розглянути подання кожного класу декількома кластерами. Тому для автоматичного формування навчальних матриць для систем класифікаційного керування дістало поширення вико-

ристання ідей і методів кластер-аналізу архівних даних [71–73].

В основу кластер-аналізу покладена гіпотеза компактності реалізацій образу, згідно з якою вектори-реалізації образів задаються у просторі ознак у вигляді згущень, які мають відповідні центри їх розсіювання. Реалізації, що відносять до одного згущення, яке часто називають кластером, таксоном, або просто класом розпізнавання, є найближчими до еталонного (усередненого) вектора-реалізації, що визначає центр розсіювання реалізацій. Класичним представником таких методів є алгоритм *k*-середніх. Розглянемо основні кроки цього алгоритму.

Крок1. Задається кількість кластерів *k*, які необхідно утворити.

Крок 2. Випадково з вибірки обирається k векторів-реалізацій, що на цьому кроці вважаються центрами кластерів.

Крок 3. Кожний вектор-реалізація «приписується» до одного із *k* кластерів, відстань до якого найкоротша.

Крок 4. Розраховується новий центр кожного кластера, ознаки вектора-реалізації якого розраховуються як середнє арифметичне ознак об'єктів, що входять до цього кластера.

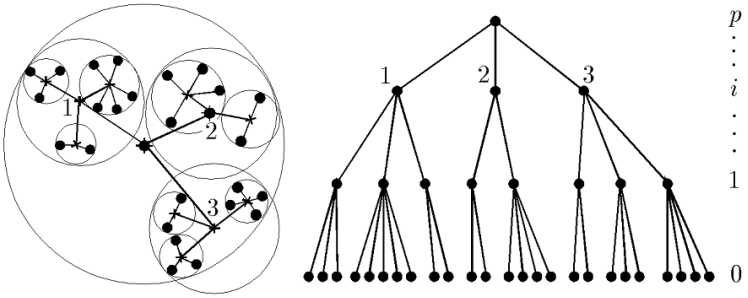
Крок 5. Ітераційне повторення кроків 3–4 до стабілізації складу кластерів.

Ще одним популярним алгоритмом ітераційного кластер-аналізу є FOREL (від «формальний елемент»), де використовуються кластери гіперсферичної форми [71]. На початку алгоритму будується гіперсфера мінімального радіуса. Кожне збільшення радіуса супроводжується обчислен-

ням центра ваги точок, що знаходяться в межах гіперсфери, і перенесенням центра гіперсфери до одержаного центра ваги. Умовою зупинення процедури центрування є стабілізація складу внутрішніх об'єктів кластера і незмінність координат центра. Внутрішні об'єкти кластера вилучаються з розгляду під час побудови гіперсфери наступного кластера.

Недоліком алгоритмів *k*-середніх FOREL є їх чутливість до вибору початкової точки пошуку, що може призвести до нестійкої кластеризації.

На рисунку 3.14 показано структури даних, побудованих у процесі ієрархічного кластер-аналізу [71].



а *б*

Рисунок 3.14 – Схеми ієрархічного кластер-аналізу:
а – геометричне розбиття простору ознак;
б – дендрограма

Серед алгоритмів ієрархічної класифікації виділяють два основні типи: висхідні (агломеративні) і спадні (дивізимні) алгоритми. В агломеративних алгоритмах на початку роботи кожний вектор-реалізацію розміщують в окре-

мому кластері, а потім об'єднують кластери в більші, поки всі вектори-реалізації розподілу не будуть міститися в одном кластері. Дивізимні алгоритми працюють за принципом «зверху вниз»: на початку всі реалізації розміщують в один клас, який потім розбивають на менші кластери.

Вектори-реалізації функціонального стану компонентів та сервісів інформаційно-телекомунікаційного середовища, як правило, мають високу розмірність. До словника ознак можуть входити дані про трафік, канали зв'язку, мережеві пристрої та обчислювальні ресурси вузлів оброблення. Такі дані збираються стандартними системами керування з різноманітних сенсорів та одержуються від програмного забезпечення мережевого рівня на клієнтському та мережевому обладнанні. Однак для вирішення поточних завдань керування доступністю, конфігураціями, продуктивністю та безпекою ІТ-інфраструктури корисною є лише частина доступної інформації. Тому для підвищення оперативності машинного навчання та достовірності вирішальних правил можуть бути застосовані різні методи редукції словника ознак.

У загальному випадку процес редукції словника ознак є ітераційним, кожна ітерація якого вміщує етапи генерації варіантів словника, оцінювання їх ефективності та перевірки критерію зупину [74]. Етап генерації словників може починатися як з варіанта словника, що містить всі ознаки, так і з пустого або випадково сформованого. Наступний словник створюється шляхом додавання/видалення ознак із сформованого на попередній ітерації варіанта або повторного випадкового формування. Оцінювання ефективності

словника ознак може здійснюватися за дистанційними, інформаційними, кореляційними, сумісними та точнісними мірами якості. Умовами зупинення можуть бути досягнення граничного значення міри якості, граничної кількості ітерацій чи відсутність зростання міри якості після виконання ітерації алгоритму.

Таким чином, дані, що були накопичені в процесі моніторингу процесу функціонування інформаційно-телекомунікаційного середовища, можуть бути використані для формування за допомогою алгоритмів машинного навчання та кластер-аналізу оптимальних в інформаційному сенсі вирішальних правил для передбачення ситуацій, які призводять до матеріальних витрат чи втрати прибутку.

3.5. Методи оптимізації параметрів функціонування системи керування розподіленим обчислювальним середовищем

Зведення частинних критеріїв ефективності навчання системи керування інфокомунікаційним середовищем до єдиного інтегрального критерію J дозволяє перейти до розгляду однокритеріальної задачі багатопараметричної оптимізації. При цьому вектор просторово-часових параметрів функціонування системи керування можна подати у вигляді

$$g =< g_1,...,g_{\Xi_1},\ f_1,...,f_{\Xi_2},\psi_1,...,\psi_{\Xi_3} >, \qquad (3.5.1)$$

де $< g_1,...,g_{\xi_1},...,g_{\Xi_1} >$ – параметри, які впливають на вартість експлуатації та обслуговування системи керування;

$< f_1,..., f_{\xi_2},..., f_{\Xi_2} >$ – параметри, що впливають на кодування контрольованих характеристик;

$<\psi_1,...,\psi_{\xi_3},...,\psi_{\Xi_3}>$ – параметри, які впливають на кодування функціональних станів.

Обмеження на відповідні параметри функціонування можуть бути задані у вигляді систем нерівностей :

$$R_{\xi_1}(g_1,...,g_{\xi_1},...,g_{\Xi_1}) \le 0,$$
$$R_{\xi_2}(f_1,..., f_{\xi_2},..., f_{\Xi_2}) \le 0,$$
$$R_{\xi_3}(\psi_1,...,\psi_{\xi_3},...,\psi_{\Xi_3}) \le 0.$$

Згідно з принципами ієрархічної впорядкованості та відкладених рішень процес синтезу прогностичних вирішальних правил системи керування можна подати як ієрархічну ітераційну процедуру оптимізації структурованих параметрів її навчання :

$$g^* = <\arg\left\{\max_G\left\{\max_F\left\{\max_\Psi J\right\}\right\}\right\}>, \qquad (3.5.2)$$

де G – області допустимих значень множини параметрів $< g_1,...,g_{\xi_1},...,g_{\Xi_1} >$; F – області допустимих значень

множини параметрів $< f_1,..., f_{\xi_2},..., f_{\Xi_2} >$; Ψ – області допустимих значень множини параметрів $<\psi_1,..., \psi_{\xi_3},..., \psi_{\Xi_3}>$.

У робочому режимі система керування повинна прийняти оптимальне оперативне рішення s^* щодо розподілу ресурсів інформаційно-телекомунікаційного середовища. Це рішення знаходиться в процесі мінімізації комплексного критерію (3.3.5), що оцінює рівень енергоспоживання та прогнозованих порушень SLA-угод :

$$s^* = \arg\min_{s \in S} \sum_{i=1}^{I} \lambda_i k_i(s),$$

де λ_i – коефіцієнт важливості (пріоритетності) i-го частинного критерію; $k_i(s)$ – нормалізовані значення частинних критеріїв ефективності рішення $s \in S$ щодо розподілу ресурсів; S – область допустимих рішень.

Кожне рішення s щодо розподілу ресурсів за визначеною схемою кодується вектором дійсних або двійкових чисел :

$$s = < s_1,..., s_r,..., s_R > . \qquad (3.5.3)$$

Вибір схеми кодування рішення s залежить як від виду ресурсів і типу задач, які їх використовують, так і вибору частинних критеріїв. Для розподілу обчислювальних задач між віртуальними обчислювальними вузлами вектор s може складатися з числових значень моментів запуску

множини задач на заданій множині вузлів. Крім того, розподіл задач можна задати у вигляді бітового рядка, що являє двійкову матрицю впорядкування завдань, де елементи одного її рядка подають послідовність виконання задач на одному обчислювальному вузлі. Для розподілу мережевих ресурсів між потоками пакетів, що надходять на порти маршрутизатора, вектор s може складатися з вагових коефіцієнтів дистанційних метрик для кожного каналу зв'язку.

З огляду на вищевикладене задачі синтезу прогностичних вирішальних правил і планування використання ресурсів інформаційно-телекомунікаційної системи зводяться до задачі однокритеріальної багатопараметричної оптимізації. При цьому гетерогенність, ієрархічність інформаційно-телекомунікаційного середовища, багатофакторність та нестаціонарність процесів споживання ресурсів обумовлюють нелінійність, багатоекстремальність та високу розмірність оптимізаційної задачі. Для ефективного розв'язання оптимізаційних задач з подібними характеристиками у 80-х роках минулого століття почали інтенсивно розроблятися стохастичні пошукові алгоритми оптимізації [64, 70]. Останнім часом у цьому напрямку спостерігається стрімке зростання інтересу до розроблення алгоритмів, в основу яких покладено ідеї, запозичені з природи, а також базові постулати універсальності, фундаментальності, властиві самоорганізації природних систем. Одним із таких напрямків є мультиагентні методи інтелектуальної оптимізації, що базуються на моделюванні колективного (ройового) інтелекту (Swarm intelligence) [75].

Алгоритми ройового інтелекту відносять до класу евристичних алгоритмів. Евристичний алгоритм – це алгоритм, призначений для більш швидкого вирішення проблеми порівняно з класичними методами, або для знаходження наближеного ефективного розв’язку, коли інші методи не в змозі знайти точний розв’язок. Це досягається за рахунок оптимальності, повноти, точності або швидкості виконання алгоритму.

У рамках «колективного» інтелекту глобальна поведінка всієї системи розглядається як результат взаємодій ряду простих агентів. Прихильники цього напряму спираються на такі положення [75] :

1) багатоагентна система – це популяція простих і залежних один від одного агентів;

2) кожен агент самостійно визначає свої реакції на події в локальному середовищі та взаємодії з іншими агентами;

3) зв'язки між агентами є горизонтальними, тобто не існує агента-супервізора, що керує взаємодією інших агентів;

4) немає точних правил, що дозволяють визначити глобальну поведінку агентів;

5) поведінка, властивості та структура на колективному рівні породжуються лише локальними взаємодіями агентів.

Великого поширення на практиці дістали такі популяційні алгоритми пошуку глобального максимуму багатоекстремальної функції критерію оптимізації, побудовані на інтелекті рою: алгоритми мурашиних колоній, рою бджіл, косяків риб, летючих мишей, зграї птахів, рою частинок

тощо. Загальна схема ройових алгоритмів передбачає такі етапи :

1) ініціалізацію рою шляхом створення, як правило, випадковим чином у просторі пошуку певної кількості початкових наближень до шуканого рішення;

2) міграцію агентів рою шляхом застосування міграційних операторів (специфічних для кожного ройового алгоритму), які переміщують агентів в області пошуку, наближаючи їх до екстремуму критерію оптимізації;

3) перевірку умови зупинення алгоритму, невиконання якої означає повернення до другого етапу.

Умовою зупинення ройового алгоритму оптимізації може бути або досягнення заданого числа ітерацій, або стан стагнації, коли краще досягнуте значення критерію оптимізації не змінюється впродовж заданого числа ітерацій.

Розглянемо один із найбільш популярних алгоритмів ройової оптимізації – алгоритм рою частинок (Particle Swarm Optimization, PSO), оскільки він дозволяє знайти глобальний максимум критерію, не потребує початкових наближень і відрізняється простотою реалізації [70]. Завдяки випадковості розподілу частинок та їх хаотичності в русі з’являється дуже велика імовірність знайти оптимальний розв’язок за декілька ітерацій. Елемент випадковості в процесі пошуку забезпечується параметрами алгоритму, значення яких генеруються випадково із заданого діапазону $(0, 1)$ у відповідності до нормального закону розподілу $U(0, 1)$.

Ефективність кожної частинки, тобто її близькість до глобального оптимуму, вимірюється за допомогою наперед визначеної фітнес-функції, роль якої виконує критерій оптимізаційної задачі. Кожна частинка зберігає таку інформацію: P_j – поточна позиція j-ї частинки; V_j – поточна швидкість частинки; $Pbest_j$ – краща персональна позиція частинки. Краща персональна позиція j-ї частинки – це позиція j-ї частинки, в якій значення фітнес-функції для частинки було максимальним на поточний момент часу. Крім того, з метою пошуку глобального екстремуму фітнес-функції найкраща частинка шукається в усьому рої, а її позиція позначається як $Gbest$. Розглянемо основні кроки реалізації алгоритму рою частинок для оптимізації вектора параметрів рішення (3.5.1).

1. Ініціалізацію рою:

1) ініціалізацію кількості частинок ;

2) ініціалізацію розмірності кожної частинки R та ініціалізацію меж зміни параметра s_r ;

3) ініціалізацію початкових позицій частинок

$$P_j(0) := s_{\max} U(0,1);$$

4) ініціалізацію початкових швидкостей частинок $V_j(0) := 0$;

5) ініціалізацію максимальної швидкості частинок у $V_{\max,\, r}$;

6) ініціалізацію вагових коефіцієнтів для формули швидкості, тобто ваги інерції w та констант прискорення c_1 і c_2.

2. Інкремент номера ітерації: $k := k+1$.

3. Інкремент номера частинки: $j := j+1$.

4. Інкремент номера координати в позиції: $r := r+1$.

5. Розрахунок нового стану частинки:

1) розрахунок r-ї компоненти швидкості для j-ї частинки за правилами

$$V_{j,r}(k+1) := wV_{j,r}(k) + c_1 a_{1,r}(k) \times$$
$$\times [Pbest_{j,r}(k) - P_{j,r}(k)] + c_2 a_2(k) \cdot [Gbest_j - P_{j,r}(k)],$$
$$V_{j,r}(k+1) := \begin{cases} V_{j,r}(k+1) & якщо \quad V_{j,r}(k+1) < V_{\max,r}, \\ V_{\max,r} & якщо \quad V_{j,r}(k+1) \geq V_{\max,r}, \end{cases}$$

де $a_1(k) = U(0,1)$, $a_2(k) = U(0,1)$;

2) оновлення позиції частинки

$$P_j(k+1) := P_j(k) + V_j(k+1);$$

3) зміна швидкості у разі виходу за межі простору за правилом

$$V_{j,r}(k+1) := V_{j,r}(k+1)\alpha,$$

де α – параметр типу межі простору

($\alpha = 0$ – поглинальна межа; $\alpha = 1$ – прозора;

$\alpha = -1$ – відзеркалювальна; $\alpha = -U(0,1)$ – демпфірувальна);

4) обчислення цільової функції $J_j(k+1)$;

5) оновлення значень найкращої персональної Pbest та глобальної Gbest -позицій

$$Pbest_j(k+1) := \begin{cases} Pbest_j(k), \text{якщо } J(P_j(k+1)) \le J(Pbest_j(k)); \\ P_j(k+1), \text{якщо інакше}; \end{cases}$$

$$Gbest(k+1) := \arg\max_j \{J(Pbest_j(k+1))\}.$$

6. Перевірка умови зупинення: якщо

$$(k < k\_\max) \wedge (J(Gbest(k+1) < J\_\max),$$

де $k\_\max$ – задана максимальна кількість ітерацій; $J\_\max$ – граничне значення фітнес-функції, то перехід до кроку 2, інакше – до кроку 7.

7. ЗУПИНЕННЯ.

Ще одним популярним представником ройових алгоритмів оптимізації є алгоритм пошуку косяком риб (Fish School Search, FSS), що відрізняється простотою реалізації, інтерпретабельністю та високою швидкістю збіжності.

В алгоритмі FSS кожна риба зберігає одне з рішень задачі. При цьому косяк риб є агрегацією агентів рою, які рухаються приблизно з однією й тією самою швидкістю та орієнтацією, підтримуючи приблизно однакову відстань між собою. Індивідуальний успіх кожної риби в пошуці

рішення характеризується її вагою, що відіграє роль пам'яті. Кожна ітерація пошуку виконує дві групи операторів – оператори годування та оператори плавання.

Оператор годування формалізує успішність дослідження агентами тих чи інших областей «акваріума» і полягає в обчисленні ваги z-го агента, яка пропорційна нормалізованій різниці значень фітнес-функції на наступній та поточній ітераціях,

$$w_z[k+1] = w_z[k] + \frac{J(P_z[k+1]) - J(P_z[k])}{\max(J(P_z[k+1]), J(P_z[k]))}, \; z = \overline{1, Z},$$

де $P_z[k+1]$, $P_z[k]$ – позиція z-го агента в багатовимірному просторі рішень на k-й та $(k+1)$-й ітерації алгоритму FSS.

Максимально можливе значення ваги агента w_z в алгоритмі FSS обмежується значенням $w_{\max} > 0$. При цьому під час ініціалізації популяції всім агентам присвоюється вага, що дорівнює $w_{\max} \cdot 0{,}5$.

В алгоритмі FSS розрізняють три види плавання – індивідуальне, інстинктивно-колективне та колективно-вольове. Ці види плавання здійснюються послідовно один за одним в окремі інтервали часу $(t, \tau], (\tau, \theta], (\theta, t')$, $t < \tau < \theta < t', t' = t+1$.

Під час індивідуального плавання агентів відбувається їх переміщення, що має рівноймовірний випадковий ха-

рактер. При цьому за одну ітерацію алгоритму FSS крок індивідуального плавання виконується фіксовану кількість разів. Компоненти кроку переміщення V_z^{ind} рівномірно розподілені в заданому інтервалі $v_{\max}^{ind}$:

$$V_z^{ind} = U(0;1)v_{\max}^{ind},\ z = \overline{1,Z},$$

де $U(0;1)$ – випадкове число із заданого діапазону $(0;1)$.

У процесі інстинктивно-колективного плавання на кожного з агентів чинять вплив всі інші агенти популяції і цей вплив пропорційний індивідуальним успіхам агентів. При цьому позиції агентів обчислюються за формулою

$$P_z^{\theta} = P_z^{\tau} + \frac{\sum_j V_j^{ind}(\tau)(J(P_j^{\tau}) - J(P_j^{t}))}{\sum_j J(P_j^{\tau}) - J(P_j^{t})},\ z = \overline{1,Z}.$$

Колективно-вольове плавання полягає у зміщенні всіх агентів у напрямку поточного центра тяжіння популяції за умови збільшення сумарної ваги косяка риб у результаті індивідуального та інстинктивно-колективного плавання. Якщо сумарна вага зменшилася, то зміщення відбувається в протилежному напрямку. Колективно-вольове плавання виконується за правилами

$$P_z^{'} = P_z^{\theta} \pm v^{vol}(P_z^{\theta} - P_c^{\theta}),\ z = \overline{1,Z}, \qquad (3.5.4)$$

де P_c^{θ} – координати центра тяжіння косяка риб, що визначаються за формулою

$$P_c^{\theta} = \frac{\sum_z w_z^{\theta} P_z^{\theta}}{\sum_z w_z^{\theta}}.$$

У формулі (3.5.4) знак плюс використовується за умови

$$\sum_z w_z^{\theta} > \sum_z w_z^{\theta-1},$$

а знак мінус – у протилежному разі.

При цьому розмір кроку переміщення агентів v^{vol} є випадковою величиною :

$$v^{vol} = v_{\max}^{vol} U(0;1),$$

де $v_{\max}^{vol}$ – значення максимально допустимої довжини кроку переміщення при колективно-вольовому плаванні.

Однією з основних проблем конструювання ройових алгоритмів є проблема забезпечення балансу між швидкістю збіжності алгоритму, яка характеризує швидкість зменшення різноманітності в рої, і диверсифікацією пошуку, що характеризує широту пошуку і відповідає за збереження різноманітності в рої. Найбільш розвиненими механіз-

мами вирішення проблеми балансу є адаптація та самоадаптація ройових алгоритмів. Реалізація цих механізмів передбачає використання допоміжного фонового алгоритму пошуку, що здійснює настроювання параметрів базового ройового алгоритму і забезпечує максимально можливу швидкість його збіжності [75].

Таким чином, синтез прогностичних вирішальних правил системи керування інформаційно-телекомунікаційним середовищем та безпосереднє прийняття рішень передбачають пошук екстремуму функції критерію багатьох параметрів. При цьому одним із найбільш ефективних підходів до пошуку глобального оптимуму оптимізаційної задачі даного типу є використання ідей і методів ройового інтелекту.

3.6. Контрольні запитання та завдання для самопідготовки

1. Що називається інформацією в широкому розумінні?
2. Що називається інформацією в логіко-гносеологічному аспекті?
3. Що називається кібернетикою в логіко-гносеологічному аспекті?
4. Що називається одиницею кількості інформації?
5. Що називається інформаційною мірою Хартлі? Наведіть формулу міри Хартлі.
6. Що називається безумовною (апріорною) ентропією? Наведіть формулу безумовної ентропії.

7. Який вигляд має формула умовної (апостеріорної) ентропії? Що вона характеризує під час прийняття рішень?

8. Який вигляд має формула нормованого ентропійного критерію Шеннона?

9. Що називається першою достовірністю рішень?

10. Що називається помилкою першого роду рішень?

11. Що називається помилкою другого роду рішень?

12. Що називається другою достовірністю рішень?

13. Виразіть умовну (апостеріорну) ймовірність $p(\mu_1 / \gamma_1)$ через точнісні характеристики двохальтернативного рішення.

14. Виразіть умовну (апостеріорну) ймовірність $p(\mu_1 / \gamma_2)$ через точнісні характеристики двохальтернативного рішення.

15. Виразіть умовну (апостеріорну) ймовірність $p(\mu_2 / \gamma_1)$ через точнісні характеристики двохальтернативного рішення.

16. Виразіть умовну (апостеріорну) ймовірність $p(\mu_2 / \gamma_2)$ через точнісні характеристики двохальтернативного рішення.

17. Який вигляд має формула критерію Шеннона як функція точнісних характеристик?

18. Як обчислюються на практиці оцінювання точнісних характеристик?

19. Який вигляд має робоча формула нормованого критерію Шеннона?

20. Яким умовам повинна задовольняти робоча область визначення функції інформаційного критерію?

21. Яку конструкцію має критерій Кульбака?

22. Виразіть через точнісні характеристики повну ймовірність P_t правильного прийняття рішень.

23. Виразіть через точнісні характеристики повну ймовірність P_f неправильного прийняття рішень.

24. Наведіть аналітичну формулу критерію Кульбака як функцію від точнісних характеристик.

25. Наведіть робочу формулу критерію Кульбака.

26. Який вигляд має нормований критерій Кульбака?

27. За яких умов значення критерію Кульбака, що обчислюється за формулою (3.2.13), буде максимальним?

28. Яке максимальне значення має критерій Кульбака, що обчислюється за формулою (3.2.13), при $n=100$ і $r=2$?

29. Назовіть основні частинні критерії оптимізації інформаційно-телекомунікаційної системи.

30. Яке основне завдання загального синтезу системи керування інформаційно-телекомунікаційного сервісу?

31. Наведіть формулу узагальненого критерію ефективності системи керування інформаційно-телекомунікаційного сервісу.

32. Наведіть формулу узагальненого критерію економічної ефективності системи керування інформаційно-телекомунікаційного сервісу з точки зору отримуваного прибутку.

33. Який вигляд має узагальнений критерій ефективності І. В. Кузьміна?

34. Який вигляд має узагальнений критерій ефективно-

сті здатної навчатися системи керування інформаційно-телекомунікаційного сервісу?

35. Яка ідея принципу зваженої суми частинних критеріїв оптимізації?

36. Яка ідея методу найближчого сусіда?

37. Яка ідея методу *k*-найближчих сусідів?

38. Коли доцільно використовувати як міру близькості квадрат евклідової відстані?

39. Коли доцільно використовувати як міру близькості степеневу евклідову відстань?

40. Що називається кодовою відстанню Хеммінга?

41. Що називається гетерогенною дистанційною мірою? Наведіть формулу.

42. Що називається нормалізованою відстанню між кількісними ознаками? Наведіть формулу.

43. Яка суть статистичних методів класифікації?

44. Наведіть формулу Байєса.

45. Яке правило прийняття рішень в статистичному класифікаторі Байєса?

46. Які загальні недоліки статистичних методів класифікації?

47. Яка ідея методу опорних векторів?

48. Наведіть вирішальне правило методу опорних векторів.

49. Які основні недоліки методу опорних векторів?

50. Які основні концептуальні положення інформаційно-екстремальної інтелектуальної технології ІЕІ-технології) аналізу даних?

51. Що називається контрольним полем допусків на

ознаку розпізнавання?

52. Що називається нормованим полем допусків на ознаку розпізнавання?

53. Наведіть категорійну модель інформаційно-екстремального машинного навчання з оптимізацією системи контрольних допусків на ознаки розпізнавання.

54. Що називається параметром машинного навчання?

55. Яка ідея алгоритму прийняття оперативного рішення щодо розподілу ресурсів інфокомунікаційної системи?

56. Для розв'язання яких задач доцільно використовувати алгоритми ройового інтелекту?

57. Які основні концептуальні положення ройових алгоритмів?

58. Яка загальна схема ройового алгоритму?

59. Які умови зупину ройового алгоритму?

60. Наведіть приклади ройової оптимізації параметрів функціонування системи керування.

СПИСОК ЛІТЕРАТУРИ

1. Recommendation ITU-T E. 800. Overall network operation telephone service, service operation and human factors / Telecommunication standardization sector of ITU. – Switzerland, Geneva : WTSA, 2009. – 22 p.

2. Recommendation ITU-T Y.1540. Internet protocol data communication service – IP packet transfer and availability performance parameters / Telecommunication standardization sector of ITU. – Switzerland, Geneva : WTSA, 2011. – 8 p.

3. Бычков Е. Д. Модели управления и мониторинга состояниями сетевых элементов телекоммуникационной сети с использованием теории нечетких множеств [Текст] : автореф. дис. … д-ра техн. наук : 05.13.01 / Е. Д. Бычков. – Новосибирск, 2016. – 409 с.

4. Recommendation ITU-T P.10 / G.100. Terminals and subjective and objective assessment methods / Telecommunication standardization sector of ITU. – Switzerland, Geneva : WTSA, 2012. – 8 p.

5. Toward Total Quality of Experience : A QoE Model in a Communication Ecosystem [Text] / V. A. Rojas-Mendizabala, R. Conte-Galvana, A. Serrano-Santoyoa, A. Gomez-Gonzalez // Conference on enterprise information systems. – USA, NJ : IEEE Press, 2013. – Vol. 50, I. 4. – P. 58–65.

6. Recommendation ITU-T G. 1080. Transmission systems and media, digital systems and networks / Telecommunication standardization sector of ITU. – Switzerland, Geneva : WTSA, 2009. – 44 p.

7. Alreshoodi M. Survey on QoE/QoS correlation models for multimedia services [Text] / M. Alreshoodi, J. Woods // International journal of distributed and parallel systems. – Bristol, PA, Taylor & Francis Inc.– 2013. – Vol. 4, №. 3. – P. 53–72.

8. ETSI TS 103 294 V1.1.1. Speech and multimedia Transmission Quality (STQ). Quality of Experience. A Monitoring Architecture / Technical specification from European Telecommunications Standards Institute. – Valbonne, France : Sophia Antipolis Cedex, 2014. – 24 p.

9. Meta-Modeling QoE – Towards a Generic Methodology for Building QoE Models [Text] / M. Varela, L. Skorin-Kapov, F. Guyard, M. Fiedler // PIK – Praxis der Informationsverarbeitung und Kommunikation – 2014. – № 37 (4). – P. 265–274.

10. Approaches for Future Internet architecture design and Quality of Experience (QoE) Control [Text] / S. Battilotti, F. D. Priscoli, C. G. Giorgi, et al. // WSEAS transaction on communications. – Wisconsin, USA : World Scientific and Engineering Academy and Society – 2015. – Vol. 14 – P. 62 – 73.

11. SLA Management Handbook / TM Forum Publication. – Morristown, NJ, USA : TeleManagement Forum, 2001. – 141 p.

12. From Service Level Agreements (SLA) to Experience Level Agreements (ELA): The challenges of selling QoE to the user [Text] / M. Varela, P. Zwickl, M. Xie et al. // Communication Workshop (ICCW), 2015 IEEE International

Conference on 8–12 June 2015. – London, UK : IEEE Press, 2015. – P. 1741–1746.

13. Зведення метрик оцінювання рівня обслуговування користувачів на основі експертних оцінок [Текст] / С. Ф. Теленик, О. І. Ролік, О. М. Моргаль, О. С. Квітко // Вісник Вінницького політехнічного інституту. – 2011. – № 1. – С. 112–123.

14. Балькин Г. Ф. Системный анализ в инфокоммуникациях : учебное пособие / Г. Ф. Балькин, Ю. Г. Балькин, Л. А. Крапивянская // Государственный университет телекоммуникаций. – Киев : ГУТ МОН Украины, 2014. – 97 с.

15. Підвищення ефективності розподілу ресурсів телекомунікаційної мережі шляхом зміни марш-рутів передавання даних : електронне видання / Б. А. Бугиль, М. М. Климаш, О. А. Лаврів, І. В. Демидов // Проблеми телекомунікацій. – Харків : ХНУРЕ, 2012. – № 4 (9). – С. 33–44.

16. Zhao W. Internet Quality of Service: an Overview / W. Zhao, D. Olshefski, H. Schulzrinne // Columbia University Research Report CUCS-003-00. – Manhattan, New York, USA : IBM && Columbia, 2003. – 11 p.

17. Аветисян Р. А. Оптимизация маршрутизации в IP-сетях путем задания соответствующих метрик каналов / Р. А. Аветисян, Р. А. Геворкян // Автоматизация и системы управления – Изв. НАН РА и ГИУА. – Ереван, Республика Армения : ГИУА, 2004. – Т. LVII, № 3. – С. 500–505.

18. Богданова Н. В. Способ повышения эффективности системы управления телекоммуникационными сетями /

Н. В. Богданова // АСАУ. – Киев : НТУ КПИ, 2006. – № 9 (29). – С. 23–32.

19. Управління високопродуктивними ІТ-інфраструктурами / С. Ф. Теленик, Ю. В. Бойко, М. М. Глибовець та ін. // Вісник НТУ КПІ. – Київ : НТУ КПІ, 2015. – Т. 61 – С. 120–141.

20. P. M. Santiago del Rio. Internet Traffic Classification for High-Performance and Off-The-Shelf Systems / P. M. Santiago del Rio // Ph.D. Thesis. – Madrid, Spain : Technical University of Madrid, 2013. – 217 p.

21. Katal S. A Survey of Machine Learning Algorithm in Network Traffic Classification / S. Katal, H. Singh // International Journal of Computer Trends and Technology (IJCTT). – Madurai, India : Seventh Sense Research Group, 2014. – Vol. 9, № 6. – P. 301–304.

22. Volunteer-Based System for classification of traffic in computer networks / T. Bujlow, K. Balachandran, M. T. Riaz, J. M. Pedersen // In Proceedings of 19th Telecom-munications Forum TELFOR 2011. – Sydney : IEEE Press, 2011. – P. 210–213.

23. Iacovazzi A. Network Communication Privacy: Traffic Masking against Traffic Analysis /Alfonso Iacovazzi // Ph.D. Thesis. – Rome, Italy : Sapienza University of Rome, 2013. – 119 p.

24. Козак Р. О. Аналіз засобів забезпечення анонімності в мережі інтернет / Р. О. Козак // Інформаційно-вимірювальні та обчислювальні системи і комплекси в технологічних процесах. – Хмельницьк : Хмельницький національний університет, 2014. – № 1.– Р. 100–105.

25. Chakravarty S. Traffic Analysis Attacks and Defenses in Low Latency Anonymous Communication / S. Chakravarty // PhD thesis Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Graduate School of Arts and Sciences. – Manhattan, New York, USA : Columbia University, 2014. – 138 p.

26. Alshammari R. Machine Learning Based Encrypted Traffic Classification : Identifying SSH and Skype / R. Alshammari, A. N. Zincir-Heywood // IEEE Symposium on Computational Intelligence for Security and Defense Applications. – Ottawa, Canada : IEEE Press. – 2009. – 8 p.

27. Модели обнаружения сетевых вторжений : учеб. пособие / Г. А. Остапенко, Н. М. Радько, М. П. Иванкин, Г. А. Савенков. – Воронеж : ФГБОУ ВПО «Воронежский государственный технический университет», 2013. – 111 с.

28. Большев А. К. Алгоритмы преобразования и классификации трафика для обнаружения вторжений в компьютерные сети : авторефер. дис. … канд. техн. наук : 05.13.19 / А. К. Большев. – Санкт-Петербург, 2011. – 18 с.

29. Elich M. Flow-based Network Anomaly Detection in the Context of IPv6 / M. Elich // PH.D. THESIS PROPOSAL. – Brno : Masarykova Univerzita, 2012. – 27 p.

30. Мухин В. Е. Мониторинг состояний информационных ресурсов для реализации адаптивного управления защищенностью компьютерных систем / В. Е. Мухин, А. Н. Волокита, Е. Н. Павленко // Штучний інтелект. – Донецьк : ІПШІ МОН України і НАН України, 2005. – № 3. – С. 670–680.

31. Introduction to Grid Computing / B. Jacob, M. Brown, K. Fukui, N. Trivedi. – Armonk, NY, USA : IBM Corp., 2005. – 268 p.

32. Burrows E. Network Design Considerations for Grid Computing / E. Burrows // Engineering Systems. – San Jose, California, USA : Broadcom Corporation, 2012. – 13 p.

33. Cloud Computing Synopsis and Recommendations / L. Badger, T. Grance, R. Patt-Corner, J. Voas // NIST Special Publication. – Gaithersburg, Meryland, USA : NIST, 2012. – 81 p.

34. Tsai W.-T. Service-Oriented Cloud Computing Architecture / W.-T. Tsai, X. Sun, J. Balasooriya // Seventh International Conference on Information Technology. – Washington, USA : IEEE Computer Society, 2010. – P. 684 – 689.

35. Ameen R. Y. Survey of Server Virtualization / R. Y. Ameen, A. Y. Hamo // International Journal of Computer Science and Information Security. – Piitsburgh, Pennsylvania, USA : IJCSIS, 2013. – Vol. 11, № 3. – 10 p.

36. Шаповалов Т. С. Планирование выполнения заданий в распределенных вычислительных системах с применением генетических алгоритмов : дис. … канд. техн. наук : 05.13.11 / Т. С. Шаповалов. – Хабаровск, 2010. – 146 с.

37. Кальпеева Ж. Б. Модели и методы организации вычислительных процессов в распределенной облачной среде : дис. … доктора философии (PhD) : 6D070400 / Ж. Б. Кальпеева. – Алматы, Республика Казахстан, 2014. – 136 с.

38. Singh P. Comparative Study of Parallel Scheduling Algorithm for Parallel Job / P. Singh, Z. Quadri, A. Kumar // International Journal of Computer Applications. – Geneva, Switzerland : Inderscience Enterprises Ltd, 2016. – Vol. 134, №. 10. – P. 10–14.

39. Multitarget Heuristic Algorithm f or Virtual Machine Placement / L. Chen, J. Zhang, L. Cai et al. // International Journal of Distributed Sensor Networks. – New York, NY, USA : Hindawi Publishing Corporation, 2015. – Vol. 2015 – 14 p.

40. Грушин Д. А. Энергоэффективные вычисления для группы кластеров /Д. А. Грушин, Н. Н. Кузюрин // Труды Института системного программирования РАН. – Москва : ИСП РАН, 2012. – Т. 23 – С. 433–445.

41. Garcıa J. L. B. Improved Self-management of DataCenter Systems Applying Machine Learning // J. L. B. Garcıa / Ph. D. Thesis. – Barcelona, Catalunya (Spain) : Polytechnic University of Catalonia, 2013. – 155 p.

42. Veni T. Dynamic Energy Management in Cloud Data Centers : A Survey / T. Veni, S. Mary, S. Bhanu // International Journal on Cloud Computing: Services and Architecture (IJCCSA). – New York, NY, USA : ACM, 2013. – Vol. 3, №. 4. – P. 13–26.

43. Bu X. Interference and Locality-Aware Task Scheduling for MapReduce Applications in Virtual Clusters / X. Bu, J. Rao, C.-Z. Xu // Proceedings of the 22nd international symposium on High-performance parallel and distributed computing. – New York, NY, USA : ACM, 2013. – P. 227–238.

44. A study on using uncertain time series matching algorithms for MapReduce applications // N. B. Rizvandi, J. Taheri, R. Moraveji, A. Y. Zomaya / Journal of Concurrency and Computation: Practice and Experience – Special Issue in Cloud Computing Scalability. – John Wiley Publisher. – 2012. – 19 p.

45. Tune K. K. Energy and SLA aware VM Scheduling / K. K. Tune, V. Varma // MS by Research in Computer Science and Engineering. – Hyderabad, India : International Institute of Information Technology – 2014. – 18 p.

46. Job Aware Scheduling Algorithm for MapReduce Framework / R. Nanduri, N. Maheshwari, R. Raja, V. Varma // 3rd IEEE International Conference on Cloud Computing Technology and Science. – Athens, Greece : IEEE Press, 2011. – P. 724–729.

47. Caglar F. A Performance Interference-aware Virtual Machine Placement Strategy for Supporting Soft Real-time Applications in the Cloud / F. Caglar, S. Shekhar, A. Gokhale // 3rd International Workshop on Real-time and Distributed Computing in Emerging Applications. – Madrid, Spain : Universidad Carlos III de Madrid, 2014. – 6 p.

48. Sampaio A. M. S. Energy-efficient and SLA-based Management of IaaS Cloud Data Centers / A. M. S. Sampaio // PhD thesis. – Porto, Portugal : University of Porto, 2015. – 109 p.

49. Salfner F. Using Hidden Semi-Markov Models for Effective Online Failure Prediction / F. Salfner // In Proceedings of the 26th IEEE International Symposium on

Reliable Distributed Systems. – Washington, DC, USA, IEEE Computer Society, 2007. – P. 161–174.

50. Watanabe Y. Online Failure Prediction in Cloud Datacenters / Y. Watanabe, Y. Matsumoto // FUJITSU Scientific & Technical Journal. –Tokyo, Japan : Fujitsu Ltd, 2014 – Vol. 50, № 1. – P. 66–71.

51. Кузьмин И. В. Оценка эффективности и оптимизация автоматизированных систем контроля и управления / И. В. Кузьмин. – Москва : Сов. радио, 1971. – 296 с.

52. Цымбал В. П. Основы теории информации и кодирования / В. П. Цымбал. – Киев : Вища школа, 1977. – 288 с.

53. Краснопоясовський А. С. Класифікаційний аналіз даних : навчальний посібник / А. С. Краснопоясовський. – Суми : СумДУ, 2003. – 162 с.

54. Краснопоясовський А. С. Інформаційний синтез інтелектуальних систем керування / А. С. Краснопоясовський. – Суми : СумДУ, 2004. – 261 с.

55. Довбиш А. С. Основи проектування інтелектуальних систем / А. С. Довбиш. – Суми : СумДУ, 2009. – 171 с.

56. Довбиш А. С. Інтелектуальні інформаційні технології в електронному навчанні / А. С. Довбиш, А. В. Васильєв, В. О. Любчак. – Суми : СумДУ, 2013. – 172 с.

57. Moskalenko V. V. Information-Extreme Algorithm for Optimizing Parameters of Hyperellipsoidal Containers of Recognition Classes / A. S. Dovbysh, N. N. Budnyk, V. V. Moskalenko // Journal of automation and information scien-ces. − New York : Begell House Inc., 2012. – Vol. 44, I. 10. − P. 35–44.

58. Критеріальне оцінювання ефективності інформаційних пристроїв та систем / М. А. Філинюк, В. О. Багацький, Л. Б. Ліщинський, О. В. Войцеховський. – Вінниця : ВНТУ, 2014. – 143 с.

59. Chang X. S. Data Analytics for Optimising Cyber and Data Centre Operation / X. S. Chang, S. L. Sim // DSTA Horizont. – Onward, Singapore : DSTA, 2015. – С. 54–58.

60. Полосинов С. А. Синтез интегральных оценочных критериев в задачах принятия решений / С. А. Полосинов // XII Всероссийское совещание по проблемам управления. – Москва : ВСПУ, 2014. – С. 7943–7954.

61. Технологии анализа данных: Data Mining, Visual Mining, Text Mining, OLAP. – 2-е изд. / А. А. Барсегян, М. С. Куприянов, В. В. Степаненко, И. И. Холод. – Санкт-Петербург : БХВ-Петербург, 2007. – 384 с.

62. Cerioli A. Robust classification with categorical variables / A. Cerioli, M. Riani, A. C. Atkinson // Proceedings in Computational Statistics. − Heidelberg : Physica-Verlag HD, 2006. − P. 507–519.

63. Alkharusi H. Categorical Variables in Regression Analysis: A Comparison of Dummy and Effect Coding / H. Alkharusi // International Journal of Education. – Las-Vegas : Macrothink Institute, 2012. – Vol. 4, № 2. – P.202–210.

64. Москаленко В. В. Информационно-экстремальный метод классификации наблюдений с категориальными признаками / А. С. Довбыш, В. В. Москаленко, А. С. Рыжова // Кибернетика и системный анализ. – Киев, Украи-

на : Институт кибернетики им. В. М. Глушкова НАН Украины, 2016. – Вып. 52, № 2. – С. 4–13.

65. Мерков А. Б. Распознавание образов. Введение в методы статистического обучения / А. Б. Мерков. – Москва : URSS, 2011. – 254 с.

66. Воронцов К. В. Обзор современных исследований по проблеме качества обучения алгоритмов / К. В. Воронцов // Таврический вестник информатики и математики, 2004. – №1. – С. 5–22.

67. Singh R. Issue related to sampling techniques for network traffic dataset / R. Sigh, H. Kumar, R. K. Singla // International Journal of Mobile Network Communications & Telematics. – Sydney, Australla : WSP, 2013– Vol. 3, № 4. – P. 75–85.

68. An Application of Oversampling, Undersampling, Bagging and Boosting in Handling Imbalanced Datasets / B. W. Yap, K. A. Rani, H. A. A. Rahman et al. // Proceedings of the First International Conference on Advanced Data and Information Engineering. Lecture Notes in Electrical Engineering. – Singapore : Springer Science, 2014. – Vol. 285 – P. 13–22.

69. Загоруйко Н. Г. Прикладные методы анализа данных и знаний / Н. Г. Загоруйко. − Новосибирск : ИМ СО РАН, 1999. – 270 с.

70. Intelligent Decision Support System for Medical Radioisotope Diagnostics with Gamma-camera / V. V. Moskalenko, A. S. Dovbysh, A. S. Rizhova, O. V. Dyo-min // Journal of Nano- and Electronic Physics. – Sumy, Ukraine : Sumy State University, 2015. – Vol. 7, № 4. – P. 04036-1–04036-7.

71. Местецкий Л. М. Математические методы распознавания образов : курс лекций / Л. М. Местецкий. – Москва : МГУ, ВМиК, 2004. – 85 с.

72. Довбиш А. С. Оптимізація словника ознак розпізнавання системи керування, що навчається / А. С. Довбиш, І. В. Шелехов, О. В. Коробченко // Адаптивні системи автоматичного управління. – 2015. – № 2 (27). – С. 44–50.

73. Карпенко А. П. Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой / А. П. Карпенко. – Москва : МГТУ им. Н. Э. Баумана, 2014. – 446 с.

Навчальне видання

Москаленко В’ячеслав Васильович,
Довбиш Анатолій Степанович

ВСТУП ДО ІНФОРМАЦІЙНОГО АНАЛІЗУ І СИНТЕЗУ ІНФОКОМУНІКАЦІЙНИХ СИСТЕМ

Навчальний посібник

Художнє оформлення обкладинки І. В. Шелехова
Редактор М. Я. Сагун
Комп’ютерне верстання В. В. Москаленка

Формат 60×84/16. Ум. друк. арк 13,25. Обл.-вид арк. 8,23. Тираж 300 пр. Зам. №

Видавець і виготовлювач
Сумський державний університет,
вул. Римського-Корсакова, 2, м. Суми, 40007
Свідоцтво суб’єкта видавничої діяльності ДК № 3062 від 17.12.2007.