

где $H(k)$ - эстиматор оценивания, m_0 - математическое ожидание начального состояния системы.

Таким образом, решение рассмотренной задачи сводится к решению двух независимых задач, а именно задачи оптимального детерминированного управления и задачи оптимального стохастического оценивания состояний.

Программная реализация алгоритма была осуществлена на алгоритмическом языке C++.

Построенная математическая модель позволяет не только исследовать перспективы развития предприятия, но и осуществлять оптимальное управление им. Оно предоставит возможность не только достигать высоких финансовых показателей, но и обеспечит надежное стабильное функционирование предприятия.

РАЗРАБОТКА ПРИЛОЖЕНИЙ ДЛЯ УПРАВЛЕНИЯ МЕТАВЫЧИСЛЕНИЯМИ

Дегтяренко Б.В.

"Метакомпьютер - это сеть географически распределенных гетерогенных вычислительных ресурсов, объединенных программным обеспечением таким образом, что все эти ресурсы могут использоваться столь же легко, как и в одном ПК". Метакомпьютинг дает неограниченные вычислительные мощности, столь необходимые для работы систем, обеспечивающих национальную безопасность, моделирующих экологические задачи, природные катаклизмы, предсказывающие погоду.

Вот лишь некоторые области, где возникают задачи подобного рода:

- Предсказания погоды, климата и глобальных изменений в атмосфере
- Науки о материалах
- Построение полупроводниковых приборов
- Сверхпроводимость

СЕКЦИЯ ИНФОРМАТИКИ

- Структурная биология
- Разработка фармацевтических препаратов
- Генетика человека
- Квантовая хромодинамика
- Астрономия
- Гидро- и газодинамика
- Эффективность систем сгорания топлива
- Разведка нефти и газа
- Вычислительные задачи наук о мировом океане
- Распознавание и синтез речи

Еще одна область применения метакомпьютерных технологий - выполнение параллельных программ на процессорах, установленных в различные компьютеры. Обмен данными между процессорами обеспечивается не по специально разработанным каналам связи, как в суперкомпьютерных архитектурах, а по локальным и даже глобальным сетям с серийно выпускаемым и потому недорогим оборудованием. Именно поддержка параллельных распределенных процессов отличает метакомпьютинг от модели клиент-сервер, в которой вызовы процедур происходят по синхронной схеме. Этот вариант применения метакомпьютинга самый интересный: не делая миллионов вложений в суперкомпьютеры, можно получить эквивалентные вычислительные мощности, например, объединив ПК сетью под управлением операционной системы Linux.

Назначение разработки:

Данная разработка предназначена для выполнения программ (проведения вычислений) в тех случаях, когда среди вычислительных систем доступных пользователю нет необходимых ресурсов (специализированных процессоров, объемов оперативной памяти или дискового пространства, программных модулей) или время вычисления на этих вычислительных системах превышает критические пределы (или превышает «разумные пределы»).

Структура разработки:

В данной работе будут употребляться 3 термина, характеризующие программное обеспечение на вычислительных системах :

1. «Клиент» — программное обеспечение на вычислительной системе (из доступных пользователю), на которой находится исходное задание, куда будет возвращено решение (если оно вообще будет найдено).
2. «Сервер вычислений» — программное обеспечение на вычислительной системе, на которой находятся свободные ресурсы, куда возможно будет передано задание и с которой в этом случае будет возвращено решение.
3. «Сервер системы» — программное обеспечение на вычислительной системе, которое в случае поступления нового задания от клиента произведет отбор из всех готовых серверов вычислений тот или те, которые имеют достаточные ресурсы и, предположительно, с как можно меньшим временем передачи информации на эти сервера ресурсов от пользователя и назад.

Структура клиента: предварительная подготовка задания к выполнению включает в себя формирование информационного блока, который содержит необходимую для сервера системы информацию о задании (необходимые процессоры с минимальной тактовой частотой, специализированные процессоры, объемы оперативной памяти и дискового пространства, программные модули и т.д.) и, возможно, о топологическом местонахождении данной вычислительной системе. Далее следует отправка этого информационного блока на сервер системы (адрес которого должен быть известен клиенту), который вернет список серверов вычислений, готовых к началу вычислений, имеющих все необходимые ресурсы, и время передачи информации с и на которые от клиента предположительно будет быть минимальным.

Структура сервера вычислений: при запуске сервер вычислений отправляет на сервер системы информационный

СЕКЦИЯ ИНФОРМАТИКИ

блок, который содержит необходимую информацию о свободных ресурсах вычислительной системы, на которой находится сервер вычислений (характеристики процессоров, объем оперативной памяти и дискового пространства, программные модули и т.д.), и, возможно, о топологическом местонахождении данной вычислительной системе.

С сервера системы получает подтверждение регистрации в базе данных и в этом случае ожидает «опросов» скорости передачи информации с и на клиентов и передачи задания на этот сервер. В противном случае сервер вычислений откладывает отсылку информационного блока на сервер системы или завершает свою работу.

Структура сервера системы: сервер системы содержит базу данных. Поля таблицы в базе данных: уникальное имя сервера вычислений, адрес сервера вычислений, запуск сервера вычислений, готовность сервера вычислений, множество полей, содержащее информацию о топологическом местонахождении вычислительной системы сервера вычислений, множество полей содержащих информацию о свободных ресурсах вычислительной системы сервера вычислений.

Сервер системы все время ожидает информационных блоков от серверов вычислений для внесения изменений в базе данных:

- для регистрации в базе данных;
- для включения в группу готовых серверов вычислений;
- для исключения из группы готовых серверов вычислений;
- для исключения из группы запущенных серверов вычислений.

Сервер системы так же периодически проверяет готовность серверов вычислений из группы готовых серверов и после нескольких подряд неудачных проверок одного и того же сервера вычислений удаляет его из группы запущенных серверов.

СЕКЦІЯ ІНФОРМАТИКИ

Вывод: данная разработка может быть реализована, и использоваться как на современном вычислительном оборудовании, так и на устаревшем, повышая его функциональные возможности.

Литература:

1. I.Foster, C.Kesselman. Globus: A metacomputing infrastructure toolkit. Int. J. Supercomput. Appl., 1997. <http://www.globus.org>
2. Корягин Д.А., Коваленко В.Н. Вычислительная инфраструктура будущего. «Открытые системы», 1999, № 11-12
3. В. Коваленко, Е. Коваленко. Пакетная обработка заданий в компьютерных сетях. «Открытые системы», 2000, № 7-8

ОБЧИСЛЕННЯ ТОЧНИХ ЗНАЧЕНЬ ЕЛЕМЕНТАРНИХ ФУНКЦІЙ

Борисенко В. Л., СумДУ

Постановка задачі

Задається елементарна функція і її аргумент. Необхідно обчислити її значення. Для цього можна використати ряд Тейлора – його реалізацію за допомогою формули Маклорена.

Програма повинна передбачати завдання потрібної точності обчислень – довжина ряду буде відповідно змінюватись.

Математична модель

Формула Маклорена:

$$f(x) = f(0) + \frac{f'(0)}{1!} + \frac{f''(0)}{2!} + \dots + \frac{f^{(n)}(0)}{n!} + \dots$$

В програмі будуть обчислюватись декілька функцій, наприклад така:

$$f(x) = e^x$$

$$f^{(n)}(x) = e^x$$

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots + \frac{x^n}{n!} + \dots$$