

## ДИНАМИЧЕСКОЕ СЖАТИЕ ДВОИЧНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

доц. Кулик И.А., студ. Харченко С.Н.

На современном этапе развития научно-технического прогресса, который неразрывно связан с развитием и внедрением информационных технологий, широкое развитие информационных систем привело к существенному росту объемов передаваемых данных. Для повышения эффективности использования коммуникационных и информационно-вычислительных ресурсов, применяются различные методы и средства. Среди них важную роль играет сжатие, которое стало неотъемлемой частью систем хранения и передачи данных, обеспечивая необходимую скорость их обработки путем выявления скрытых резервов в производительности отдельных элементов и системы в целом, неучтенных на этапе проектирования.

Достойное место среди методов сжимающего кодирования занимает метод адресно-векторного сжатия, отличающийся высокой скоростью преобразования данных и простотой реализации. За последние годы широкое распространение получили статические методы, однако с развитием коммуникационных систем, передаваемые по каналам связи данные становились все более разнообразными. В связи с этим актуальным является проведение исследований в области динамических методов сжатия, которые были бы адаптивными к изменению типов передаваемых данных.

В настоящей работе предлагается алгоритм адресно-векторного сжатия данных, который позволяет динамически изменять длину обрабатываемой комбинации от 8 до 32 бит, в зависимости от типа передаваемых данных с целью получения максимального коэффициента сжатия.

Входная 32 битная комбинация анализируется блоками по 8 бит. После этого рассматривается возможность объединения нескольких таких блоков последовательно расположенных комбинаций в пакеты длиной 8, 16, 24 или 32 бита с условием получения максимального сжатия при его дальнейшей обработке. Необработанные биты рассматриваются в совокупности с новопришедшим на обработку 32 битным пакетом.

Алгоритм динамического адресно-векторного сжатия состоит из следующих действий, где  $n$  - первое кодовое слово, которое формируется для дальнейшего сжатия, его длина может варьироваться от 8 до 32 бит,  $m$  - второе кодовое слово, которое подается на обработку, его длина фиксирована 8 битами:

1. Для начала цикла обработки данных получают первые  $n$  бит из входного потока данных.
2. На следующем этапе запускается цикл обработки, и получают следующие  $m$  бит.
3. Производится проверка на сжимаемость первых  $n$  бит, в случае, если они сжимаемы, осуществляется переход к п. 4, в противном случае - к п.5.
4. На этом этапе производится проверка сжимаемости следующих  $m$  бит, если они так же сжимаемы, то осуществляется переход к п.7, в противном случае - к п. 6.
5. Если следующие  $m$  бит являются сжимаемыми, то осуществляется переход к п.6, в другом случае - к п.8.
6. Производится проверка сжимаемости совокупности  $(n+m)$  бит. Если они сжимаемы, то осуществляется переход к п. 9, в противном же случае - к п. 8.
7. Если первые  $n$  бит и следующие за ними  $m$  бит являются сжимаемыми, то делается проверка. Является ли коэффициент сжатия совокупности этих комбинаций большим, чем коэффициент сжатия

- последовательности  $m$ ? Если больше, то осуществляется переход к п.8, в другом случае - к п.9.
8. Две последовательности бит объединяются в одну:  $n=n+m$ . И цикл обработки повторяется (переход к п.2).
  9. На этом этапе происходит обработка кодовой комбинации  $n$ .
  10. Выдача обработанной комбинации в канал связи.
  11. В качестве первой комбинации используется следующая полученная:  $n=m$ , и цикл обработки повторяется.

Каждая обработанная комбинация для ее декодирования на приемной стороне сопровождается заголовком, который содержит служебную информацию: размер обработанной кодовой комбинации, определяемый первыми двумя битами и данные о количестве единиц в обрабатываемой последовательности.

По результатам проведенного эксперимента и полученных практических данных, можно сделать следующие выводы. При рассматриваемых размерах обрабатываемых двоичных последовательностей, когда длина заголовка пакета соизмерима с длиной данных, более эффективным является применение статического сжатия, так как оно имеет минимальную длину служебной информации в блоке и просто в реализации. С дальнейшим ростом размера пакета, когда служебная часть становится намного меньше длины самого пакета, лучшие результаты показывает динамическое сжатие. Однако, при этом возрастают затраты, связанные с необходимостью хранения больших объемов данных для обработки. Применение быстродействующих микроконтроллеров позволит обойти эту проблему и эффективно использовать динамические алгоритмы для быстрого сжатия двоичных последовательностей.