ПРИМЕНЕНИЕ НЕЙРОННЫХ СЕТЕЙ ДЛЯ ВЫБОРА АЛГОРИТМА СЖАТИЯ ДЛЯ БОЛЬШИХ ДАННЫХ НА СУПЕРКОМПЬЮТЕРАХ

С.В. Коробков

Крупные вычислительные многопроцессорные системы сейчас устроены так, что их узлы можно объединить в несколько групп и в этих группах узлы ничем не отличаются друг от друга. В том числе, такие узлы не содержат носителей данных доступных пользователю. Если же такие носители и присутствуют, то для их использования сначала необходимо загрузить на них данные, и только потом они будут доступны для обработки. Также бывает, что на таких узлах хранятся файлы операционной системы. Для хранения пользовательских данных чаще всего применяется централизованное хранилище. Часто узлы вычислителя связаны с таким хранилищем менее скоростными каналами, чем между собой. Иногда даже не все узлы имеют физическую связь с таким хранилищем [1]. В связи с этим, скорость обработки данных может быть больше, чем скорость обмена данными между вычислительными узлами и хранилищем данных. В таких случаях вычислительные узлы простаивают во время обмена данных с хранилищем.

В данной работе рассмотрим применение сжатия данных, предназначенного для ввода и/или вывода [2] [3]. С одной стороны, алгоритмы сжатия позволяют уменьшить объем передаваемых данных, но с другой стороны, на сжатие данных требуется время. Для обеспечения выигрыша во времени требуется подобрать оптимальный алгоритм. Этот алгоритм должен работать достаточно быстро. С другой стороны, он должен обеспечивать достаточно большой коэффициент сжатия, но не обязательно максимальный. Для программиста, который пишет параллельные программы, сжатие данных – это специфическая область, в которой ему потребуется разбираться некоторое время из-за сложности интерфейса библиотек сжатия, и отвлечет от решения первоочередной задачи. Поэтому программисты реализуют сжатие данных в своих параллельных программах только по острой необходимости. Но даже в случае, если они и реализуют сжатие данных, то чаще всего применяется один единственный метод, который наиболее известен. Скорее всего такой метод будет не оптимальным.

В данной работе представлена система для массивно-параллельных вычислителей, обеспечивающая автоматический выбор оптимального алгоритма сжатия данных для уменьшения времени затрачиваемом на ввод/вывод, основанная на применении нейросетевых классификаторах.

Общая структура предлагаемой системы представлена на рисунке 1.

Рис. 1 Потоки данных в системе автоматического выбора алгоритма сжатия.

На данном рисунке видно, что один и тот же набор данных сначала поступает на вход и по этому набору определяются параметры этих данных. После эти параметры поступают на вход нейронной сети. В нейронной сети определяются вероятности оптимальности всех имеющихся алгоритмов сжатия. Далее выбирается тот алгоритм который имеет наибольшую вероятность оптимальности и он используется для сжатия тех самых данных, которые подавались на вход системы автоматического выбора алгоритма сжатия.

Рассмотрим какие параметры более полно и качественно характеризуют данные. Параметры отправляемые на вход нейронной сети можно разделить на несколько классов по способу их расчета и по тому какие характеристики данных они собой представляют. В данной работе предлагается использовать 3 класса параметров. Эти классы представлены на следующем рисунке.

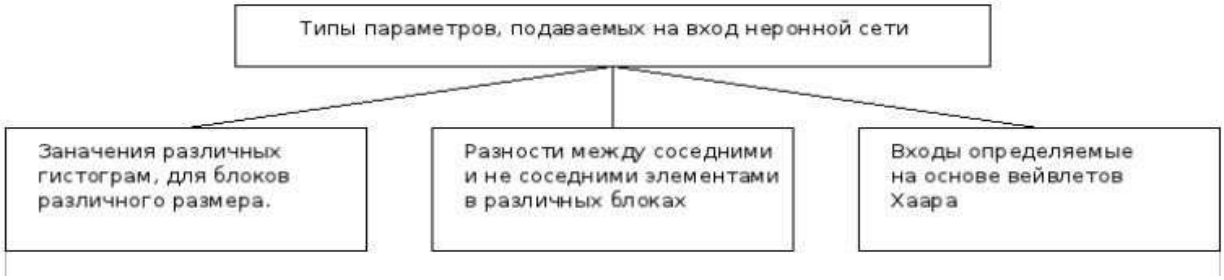

Рис. 2 Классификация параметров данных, подаваемых на вход нейросети.

В качестве первого класса используются гистограммы данных определяемы по блокам данных различного размера. Будем рассматривать распределение элементов блока относительно среднего арифметического этих элементов в блоке. Эти гистограммы покажут каким распределением обладают данные в зависимости от размера блоков. При этом на небольших блоках необходимо брать маленькое количество диапазонов для построения гистограммы, и следовательно эти диапазоны необходимо распределить неравномерно. Для использовании в качестве параметров классификаторе предлагается брать следующие размеры блоков и соответствующие им диапазоны:

1. Блок размером 16 элементов распределяется по 5 диапазонам. диапазонов не велико. Пусть среднее арифметическое всех значений в этой выборке равно x. В этом наборе используются следующие значения: от минимального значения до $10/16 \cdot x$, от $10/16 \cdot x$ до $14/16 \cdot x$, от $14/16 \cdot x$ до $18/16 \cdot x$, от $18/16 \cdot x$ до $22/16 \cdot x$, от $22/16 \cdot x$ до максимального значения.
2. Блок размером 64 элемента распределяется по 9 диапазонам, распределенным аналогично.
3. Блок размером 1024 элемента распределяется по 17 диапазонам, распределенным аналогично.
4. Блок из 16 миллионов элементов(размер всего сжимаемого блока) распределяется по 33 диапазонам, которые покрывают равномерно весь диапазон значений элементов симметрично относительно среднего арифметического.

В качестве второго класса входных параметров нейросетевого классификатора используются разности между элементами данных. Такие разности берутся для блоков различных размеров. Также такие разности берутся не только для непосредственно соседних элементов, но и для элементов стоящих через 1, 2 или 3 элемента. Такие разности показывают насколько равномерно изменяются данные, и могут показать специфическую структуру данных. В данной работе предлагается использовать разности между элементами следующего вида.

1. Набор знаковых разностей между соседними элементами. Блок из 8 элементов.
2. Усредненная разница между соседними элементами. 4 средних значения по блокам в 8 элементов. Вычисляются по 32 элементам.
3. По 4 разницы между элементами, находящимися через 1, 2, 3 элемента.
4. Средняя разница между соседними элементами на всем блоке.

Третьим классом входных параметров являются данные получаемые при помощи дискретного вейвлет преобразования. Такие параметры показывают частотные характеристики в блоках данных различного размера. Здесь предлагается использовать дискретное вейвлет преобразование Хаара. В конкретных значений для передачи в нейросетевой классификатор будем использовать следующие значения:

1. Для блока из 16 элементов применяется дискретное вейвлет преобразование Хаара 3 раза. После этого берутся 4 значения высоко частотных характеристик, и 4 значения низкочастотных.
2. Для блока из 32 элементов применяется дискретное вейвлет преобразование Хаара 5 раза. После этого берутся 4 значения высоко частотных характеристик, и 4 значения низкочастотных.
3. Для блока из 64 элементов применяется дискретное вейвлет преобразование Хаара 5 раза. После этого берутся 8 значения высоко частотных характеристик, и 8 значения низкочастотных.

Одним из основных параметров, влияющими на выбор алгоритма сжатия являются характеристики вычислителя. Среди этих параметров достаточно выделить производительность 1 вычислительного узла и производительность коммуникаций между вычислительными узлами и хранилищем. Посмотрим, какое влияние оказывают данные параметры используемого вычислителя на выбор алгоритма сжатия. В качестве элемента данных как единицы измерения будем рассматривать число с плавающей точкой. Рассмотрим различные параметры сжатия: S – длина сжимаемого блока данных, в элементах, V – скорость процессора, в элементах в секунду, R – коэффициент сжатия алгоритмом для заданных данных, W – скорость связи с хранилищем, в элементах в секунду, k – количество операций на обработку алгоритмом сжатия одного элемента. Тогда: V / k –

количество обрабатываемых элементов за секунду, $\frac{S \cdot k}{V}$ – время сжатия блока данных. $\frac{S}{R}$ – количество получившихся данных в результате сжатия, $\frac{S}{W}$ – время записи не сжатых данных, $\frac{S}{R \cdot W}$ – время записи сжатых данных. В результате получаем формулу:

$$(\frac{S \cdot k}{V} + \frac{S}{R \cdot W}) = \frac{S}{W} \cdot \omega$$

Здесь ω – параметр, показывающий преимущество сжатия, назовем его оптимальностью алгоритма, поскольку он показывает насколько применение этого алгоритма лучше простой записи на диск. Соответственно, чем меньше данный параметр, тем лучше используемый алгоритм сжатия. В случае, если данный параметр оказывается больше единицы, то такое сжатие применять невыгодно. После преобразования получим получим:

$$\omega = \frac{W}{V} \cdot k + \frac{1}{R}$$

Получаем параметр вычислителя $P = W / V$, который характеризует вычислитель с точки зрения алгоритмов сжатия. Но, в случае передачи такого параметра в качестве входного сигнала в каждый из подклассификаторов, этот параметр теряется среди других. В связи с этим необходимо рассмотреть другие методы подстройки классификатора под вычислитель.

Для классификатора наиболее разумно использовать различные нейросети для различных алгоритмов сжатия, поскольку именно такие нейросети лучше всего настраиваются. Такая нейронная сеть будет показывать вероятность получения лучшего алгоритма при применении предлагаемого алгоритма сжатия.

Для каждого алгоритма сжатия нейросеть будет состоять из нескольких нейросетевых подклассификаторов. Каждый из подклассификаторов отвечает за определенный вид входов: за гистограммные входы, за "разностные"входы, за вейвлетные входы. В результате нейросеть состоит из 3 нейросетевых подклассификаторов и одного нейрона их объединяющих. Схематически нейросетевой классификатор для каждого из алгоритмов сжатия изображен на рисунке 3.

Рис. 2 Классификация параметров данных, подаваемых на вход нейросети.

Рассмотрим подробно структуру подклассификаторов. Каждый из подклассификаторов представляет собой трехслойный нейросетевой классификатор. Количество нейронов в первом слое нейросети равно количеству входов. В третьем слое количество нейронов совпадает с количеством выходов, в нашем случае 1. В среднем слое используется количество нейронов равное половине количества нейронов в 1 слое. Схематически структура подклассификатора изображена на рисунке 4.

Рассмотрим операции, происходящие в нейронах подклассификаторов. В каждом нейроне сначала берется скалярное произведение вектора входов нейрона на вектор весов, назначенных входам. После происходит модификация полученного значения путем применения функции активации. В различных подклассификаторах применяются разные функции активации. Наиболее используемыми функциями являются сигмоида , обратная сигмоида , и функция . На выходе подклассификатора получается значение, которое является вероятностью применения соответствующего алгоритма сжатия, исходя из предложенных параметров.

Для гистограмм количество входов в подклассификаторе равно 64. Во втором слое используется 32 нейрона. В нейронах данного подклассификатора используется функция активации . Для разностей количество входов равно 25. Во втором слое используется 16 нейронов. В нейронах данного подклассификатора используется функция активации сигмоида. Для вейвлетов количество входов равно 32. Во втором слое используется 16 нейронов. В нейронах данного подклассификатора используется функция активации сигмоида.

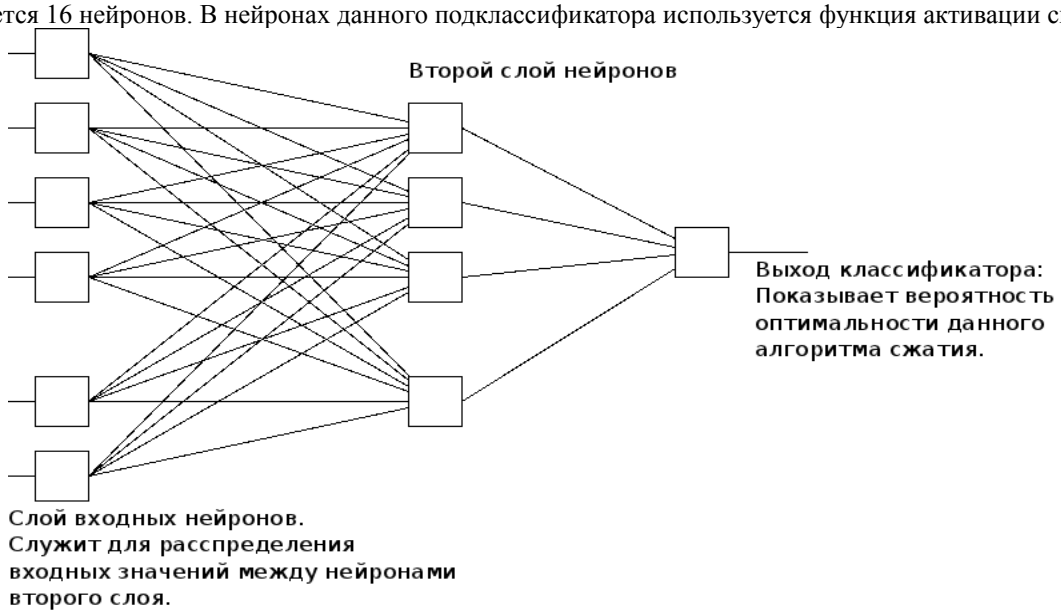

Рис. 4 Структура подклассификаторов.

Для осуществления сжатия в данной работе предлагается 6 алгоритмов сжатия данных реализованные в популярных библиотеках сжатия. В данной работе применяются 6 алгоритмов сжатия:

1. Алгоритм LZO реализованный в одноименной библиотке.
2. Алгоритм RLE реализация которого была взята из системы dact.
3. Алгоритм LZMA реализация которого была взята из системы clzip.
4. Алгоритм DEFLATE реализация которого была взята из библиотеки zlib.
5. Набор алгоритмов BWT+MTF+Huffman которые используются в bzip2.
6. Алгоритм описанный в статье [4]. Этот алгоритм показывает хорошие показатели для специфических данных.

Интерфейсы всех приведенных выше библиотек были унифицированы и из них была создана библиотека алгоритмов сжатия.

Нейронная сеть предлагаемая в данной работе необходимо обучить на большом количестве данных. Часто для обучения нейронных сетей необходимо применять большие наборы обучающих тестов, которые содержат от 1000 до 10000 обучающих примеров. Поэтому для применения обучающих алгоритмов к нейронным сетям необходимо сгененрировать обучающую выборку. Данную выборку надо генерировать случайным образом, но в соответствии с заданными различными параметрами данных. Выборка состоит из блока данных размером 16 миллионов элементов, и последовательности алгоритмов сжатия в порядке убывания их оптимальности для данной вычислительной системы. Тестирование необходимо провести на всем наборе обучающих данных. Для проверки применимости полученной системы необходимо показать, что система одинаково хорошо обучается и показывает хорошие результаты на различных вычислительных системах. Также необходимо провести тестирование на данных предназначенных для реальных задач.

В работе предложена концепция библиотеки предлагающей метод автоматического выбора алгоритма сжатия для суперкомпьютеров на основе нейросетевых классификаторов. Данная система может предполагает применение на различных вычислительных системах. Данная система может применяться для различных задач предполагающих обработку данных большого объема, характеристики которых могут меняется во время выполнения задачи. Дальнейшее направление работ предполагает развитие набора алгоритмов сжатия. Также необходимо предусмотреть возможность задавать параметр вычислителя и таким образом расширять применение данной системы на другие вычислительные системы.

ЛИТЕРАТУРА:

1. Sosa Carlos, Knudson Brant. IBM System Blue Gene Solution: Blue Gene/P Application Development. International Technical Support Organization, 2009. — August.
2. Filgueira Rosa, Singh David E., Pichel Juan C., ’s Carretero Jes. Exploiting data compression in collective I/O techniques.
3. Ke Jian, Burtscher Martin, Speight Evan. Runtime Compression of MPI Messages to Improve the

Performance and Scalability of Parallel Applications // Proceedings of the 2004 ACM/IEEE conference on Supercomputing. 2004. — November. P. 59.
4. С.В. Коробков. Алгоритмы сжатия данных при обработке изображений и визуализации // Всероссийская конференция молодых ученых «Теория и практика параллельного программирования». г.Новороссийск. Россия: Изд-во МГУ, 2010, 2010. — сентябрь. с. 541–545.