



солидации ресурсов, обеспечение непрерывности работы и обеспечение безопасности деятельности предприятия.

Используя отказоустойчивые эффективные ЦОД на базе адаптивной сети, предприятия могут перераспределять свои ресурсы, выделенные на развитие организации, адекватно реагируя на действия конкурентов, расширяя рынок своих сервисов, а также ускоряя процесс разработки новых сервисов.

Работа выполнена при финансовой поддержке гранта Президента РФ для молодых ученых - кандидатов наук МК-6016.2016.9, стипендии Президента РФ СП-505.2016.5, программы УМНИК Фонда содействия развитию малых форм предприятий в научно-технической сфере.

Литература

1. Горшков С.Г., Никитин Е.В., Саксонов Е.А. Задача формирования структуры базовой сети // Вестник Тихоокеанского государственного университета. 2010. № 2. С. 59-66.
2. Корячко В.П., Перепелкин Д.А., Иванчикова М.А. Алгоритм адаптивной маршрутизации в корпоративных сетях нескольких провайдеров связи // Вестник Рязанского государственного радиотехнического университета. 2013. № 2 (44). С. 52-56.
3. Корячко В.П., Перепелкин Д.А., Иванчикова М.А. Алгоритм парных переходов каналов связи при динамическом изменении нагрузки в корпоративных сетях нескольких провайдеров связи с различными зонами покрытия // Вестник Рязанского государственного радиотехнического университета. 2014. № 47. С. 72-80.
4. Перепелкин Д.А. Алгоритм парных перестановок маршрутов на базе протокола OSPF при динамическом отказе узлов и линий связи корпоративной сети // Вестник Рязанского государственного радиотехнического университета. 2014. № 47. С. 72-80.
5. A. Jarry, "Fast reroute paths algorithms", Telecommunication Systems. 2013. Vol. 52, Issue 2. pp. 881-888.

В.А. Кузьмин, Д.В. Глухов

ОПТИМИЗАЦИЯ ВЫЧИСЛИТЕЛЬНОГО ПРОЦЕССА В СИСТЕМЕ КЛАСТЕРНОГО ТИПА

(Саратовский государственный технический университет имени Гагарина Ю.А.)

Рост вычислительных мощностей персональных компьютеров и совершенствование технологий компьютерных коммуникаций в последние годы послужили стимулом к созданию и развитию высокопроизводительных компьютерных систем по стоимости значительно меньшей стоимости суперкомпьютеров, которые долгое время были единственным инструментом для решения



сложных задач (в частности, при имитационном моделировании) [1]. Такие системы (называемые кластерами) получили большое распространение и широко используются в различных научно-исследовательских центрах по всему миру [2]. Вообще, определение, кластера, согласно компании DEC (США), которая впервые ввела данное понятие, следующее: кластер есть группа машин, которые связаны между собой и функционируют как единое устройство. Как правило, в таких системах применяются принципы параллельного программирования, т.е., фактически, весь кластер рассматривается в качестве многопроцессорного компьютера. С другой стороны, возможен подход, в котором применяется ряд подзадач, моделирующих отдельные элементы изучаемой системы. В этом случае возникает дополнительный ряд проблем: вопросы синхронизации отдельных элементов, допустимость рассогласования по модельному времени, оптимальное распределение этих элементов внутри кластера.

Под кластерами понимается группа компьютеров (ЭВМ) объединённых в одну вычислительную сеть и работающих как единое целое. Вообще говоря, каких-то строгих требований к машинам, входящих в кластер, не предъявляется. Они могут иметь как различную аппаратную платформу (Intel, Sun, Mac и т.д., в силу большей доступности, чаще используется платформа Intel), так и различную программную платформу (т.е., на машинах кластера могут быть установлены различные операционные системы – Unix/Linux, MS Windows, SUN Solaris и т.д.) [3]. Более того, кластер может состоять из нескольких сегментов LAN, использующих различные сетевые технологии. В случае применения в пределах кластера различных аппаратно-программных решений, такой кластер будет называться гетерогенным, в случае же, если все машины представлены одинаковыми платформами, а локальная сеть построена на одной технологии, то такой кластер будет называться гомогенным [3].

Часть описанных проблем, в принципе, может быть решена при помощи теории расписаний, однако, из неё же известно, что общего решения для всех типов задач не существует, более того, может оказаться, что оптимальное решение найти вообще невозможно. С какой стороны подходить к проблеме распределения задач внутри кластера, можно решить лишь при наличии некоторых эмпирических данных, лишь после чего можно предлагать те или иные варианты. В случае успешного решения проблемы по оптимизации такого вычислительного процесса появляется возможность ускорения вычислений (что позволит приблизить модельное время к реальному, или даже ускорить его, что может быть особо значимо при моделировании длительных процессов), либо ускорения модели при некоторых приемлемых сроках получения конечного результата.

Сложности при распараллеливании возникают с циклами, в которых присутствуют различные зависимости данных. Преобразование циклических конструкций с неявной зависимостью данных, например вычисление рекуррентных последовательностей, существующими средствами либо невозможно, либо малоэффективно даже при использовании специальной аппаратной поддержки.



Применение специализированных вычислительных комплексов позволяет ускорить решение некоторых классов задач за счет аппаратной реализации операций. Однако специализация резко ограничивает область применения таких систем. Напротив, простая в организации архитектура вычислительной системы кластерного типа оказывает влияние на время решения задач. Накладные расходы, связанные с пересылкой сообщений, могут перекрыть эффект от использования множества вычислительных устройств.

Применение кластерных вычислительных систем связано с рядом проблем. Известно, что на время решения задачи в кластере влияет количество используемых процессоров [1]. При проектировании вычислительного кластера для конкретной прикладной задачи требуется предварительный анализ ресурсной базы и реализации алгоритма: необходимо оценить объем вычислительной работы и определить количество вычислительных устройств, при котором время решения будет наименьшим, что позволит максимально эффективно использовать имеющиеся ресурсы.

Определение оптимального количества вычислительных устройств нетривиально, так как время решения задачи зависит от множества факторов: количества оперативной памяти на узлах кластера, производительности дисков и коммуникационной среды и, наконец, программной реализации алгоритма решения задачи [2]. Таким образом, актуальными являются исследования, связанные с разработкой способов преобразования циклических конструкций вычисления рекуррентных последовательностей и циклов с использованием индексных множеств для многопроцессорных систем кластерного типа с учетом количества задействованных вычислительных устройств.

В работе авторы используют методики оптимизации определения минимально достаточного числа необходимых вычислительных ресурсов с помощью разработанных алгоритмов, описанных в работах, где потребовалось находить оценки параметров нелинейных систем с помощью полного метода наименьших квадратов, что обусловило использование больших вычислительных мощностей нескольких компьютеров, ввиду большой вычислительной сложности задачи.

Применение указанных алгоритмов позволило создать эффективные программы решения поставленной задачи, которые работают в реальном времени.

Литература

1. Вишневецкий, В.М. Теоретические основы построения компьютерных сетей: учеб. - метод. пособие / В.М. Вишневецкий. - Изд. м.: Техносфера, 2003. - 219 с.
2. Мяслицин, В.В. Методика определения оптимального количества процессоров кластерной системы : учеб. для вузов / В.В. Мяслицин. - Изд. Томск, ТПУ, 2007. - 468 с.
3. Максимов, Н.В. Архитектура ЭВМ и вычислительных систем : учеб. для вузов / Н.В. Максимов, Т.Л. Партыка, И.И. Попов. - М.: Инфра - М, 2005.- 512 с./



4. Львов, А.А. Алгоритм оценки параметров математических моделей линейных и нелинейных систем / А.А. Львов, А.А. Северов // Вестник СГТУ, № 4(43), 2009. – С.77-81.

5. Львов, А.А. Анализ моделей метода наименьших квадратов и методов получения оценок / А.А. Львов, М.В. Мусаев // Вестник СГТУ, № 4(43), 2009. – С.137-141.

М.С. Мутагаров

ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ МОДЕЛИРОВАНИЯ ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ МАТРИЧНОГО УМНОЖЕНИЯ

(Самарский государственный университет путей сообщения)

Многие алгоритмы управления, идентификации систем, цифровой обработки сигналов широко используют матричные вычисления, одной из основных задач в которых является умножение матриц.

Классический алгоритм матричного умножения является итеративным и ориентирован на последовательное вычисление строк результирующей матрицы C . Известно [1], если исходные матрицы A и B квадратные размера $n \times n$, каждый элемент матрицы C произведения есть скалярное произведение строки и столбца исходных матриц A и B , т.е. при умножении квадратных матриц размера $n \times n$ количество выполненных операций имеет порядок $O(n^3)$. Очевидно, что такой большой объем вычислений, которые необходимо производить в системах управления в реальном масштабе времени, существенно ограничивает сложность решаемых такими системами задач.

В настоящее время разработаны и продолжают интенсивно разрабатываться различные параллельные алгоритмы умножения матриц. В параллельных алгоритмах, исходя определения операции матричного умножения, вычисление всех элементов матрицы C может быть выполнено независимо друг от друга, т.е. используется свойство параллелизма по данным. Поэтому основной принцип организации параллельных вычислений заключается в использовании в качестве базового потока вычисления одного элемента результирующей матрицы C , т.е. вычисления над элементами одной строки матрицы A и одного столбца матрицы B . Очевидно, общее количество потоков получаемых при таком подходе равно n^2 [1], что определяет число требуемых вычислительных устройств.

Актуальной задачей параллельных вычислений является определение показателей эффективности параллелизма: оценка получаемого ускорения вычислений, степень загрузки процессоров и ее балансировка, потери производительности для организации параллелизма, отношение производительности вычислительной системы к ее стоимости и др. Эффективность параллелизма в значительной мере зависит от того, в какой мере параллельный алгоритм учитывает характерные свойства параллельной ВС.