ПРОГНОЗИРОВАНИЕ МАСШТАБИРУЕМОСТИ ЗАДАЧИ УМНОЖЕНИЯ РАЗРЕЖЕННОЙ МАТРИЦЫ НА ВЕКТОР ПРИ ПОМОЩИ МОДЕЛИ КОММУНИКАЦИОННОЙ СЕТИ

И.А. Пожилов, А.С. Семенов, Д.В. Макагон

1. Введение

В ОАО «НИЦЭВТ» ведется проект по созданию высокоскоростной коммуникационной сети [1, 2] и суперкомпьютера на базе этой сети и коммерческих микропроцессоров.

Сеть проектировалась таким образом, чтобы эффективно поддерживать задачи с мелкозернистым параллелизмом, в которых основными требованиями являются высокий темп выдачи коротких сообщений и низкая коммуникационная задержка.

Такие задачи обычно связаны с построением и обработкой графов, больших нерегулярных массивов данных, сильно разреженных матриц. Концепция сети и главные архитектурные принципы были сформулированы после проведения подробного имитационного моделирования [3] в рамках проекта по разработке отечественного суперкомпьютера «Ангара».

Сеть поддерживает детерминированную и адаптивную передачу пакетов, отказоустойчивость на канальном уровне и обход отказавших каналов и узлов. Однако основной отличительной чертой сети является эффективная поддержка односторонних коммуникаций (библиотека SHMEM, MPI2). В частности, поддерживаются неблокирующие записи, чтения, атомарные операции, а также асинхронные коллективные операции (по виртуальной древовидной подсети).

Основная модель программирования — MPI и/или SHMEM. Библиотека SHMEM является простым и удобным средством для увеличения продуктивности программирования и оптимизации критически важных мест кода в MPI программах (благодаря переходу на односторонние коммуникации).

Реализованная библиотека SHMEM соответствует стандартной библиотеке Cray SHMEM с дополнениями, позволяющими более эффективно выполнять типовые операции. Библиотека MPI реализована на основе MPICH2 поверх SHMEM.

В настоящее время разработан прототип сети на ПЛИС, состоящий из 9 узлов, соединённых в двумерный тор 3x3. В каждом узле находятся два процессора Intel Xeon E5620 2,4 ГГц (Westmere), доступно 2 ГБ RAM, сервисная сеть 1 Гбит/с Ethernet. Прототип сетевого адаптера использует ПЛИС Xilinx Virtex 5 FX100 speed grade 2, интерфейс PCI-Express gen1 x8 (20 Гбит/с) и каналы с пропускной способностью 6,25 Гбит/с в каждую сторону. К концу 2012 года планируется подготовить и выпустить маршрутизатор на кристалле (ASIC).

Текущие характеристики производительности прототипа следующие: темп выдачи сообщений — 14,62 миллионов в секунду, полная коммуникационная задержка удалённой записи между соседними узлами — 2,6 мкс (задержка на один хоп — 0,7 мкс), достигаемая чистая пропускная способность канала — 523 МБ/с, достигаемая чистая пропускная способность при выдаче из сети — 1524 МБ/с.

Для кристалла маршрутизатора (ASIC) планируется использовать интерфейс PCI-Express gen2 x16 (80 Гбит/с) и каналы с пропускной способностью 75 Гбит/с. Достигаемая чистая пропускная способность канала предположительно увеличится до 6,3 ГБ/с, достигаемая чистая пропускная способность при выдаче из сети как минимум до 6 ГБ/с, ожидаемая коммуникационная задержка между соседними узлами не превысит 500 нс. Сетью ASIC предполагается объединять процессоры, по вычислительной мощности сопоставимые с процессорами Intel Westmere, установленными на прототипе.

2. Задача умножения разреженной матрицы на вектор

Задача умножения разреженной матрицы на плотный вектор является DIS-задачей с интенсивным нерегулярным доступом к памяти, ее производительность определяется возможностями коммуникационной сети. В приложениях обычно требуется выполнение нескольких итераций умножения фиксированной матрицы на вектор, что приводит к необходимости производить обмен полученными на очередной итерации частями полного результата между вычислительными узлами для начала следующей итерации. Данную задачу в силу ее простоты довольно часто используют для оценки производительности суперкомпьютеров, на ее основе построен метод сопряженных градиентов, включенный в пакет оценочных тестов NPB.

Существует несколько стандартных способов параллельной реализации умножения разреженной матрицы на вектор [4]. В первом алгоритме «Allgather» матрицу разделяют горизонтально по вычислительным узлам. В этом случае каждому узлу необходимо иметь в своем распоряжении весь умножаемый вектор, в результате умножения на узле получается часть нового вектора, которая должна рассылаться всем узлам при помощи операции, соответствующей операции MPI_Allgather. Во втором алгоритме матрицу разделяют вертикально по узлам, тогда каждый узел получает частичную сумму для каждого элемента вектора, итоговая сумма получается для каждого узла при помощи операции, аналогичной MPI_Reduce_scatter.

Недостатком указанных алгоритмов является неэффективное использование коммуникационной сети: в силу разреженности матрицы большинство элементов плотного вектора не используются при вычислении. Причем этот эффект увеличивается с ростом числа узлов, так как при этом на каждый узел в среднем приходится все меньшее количество ненулевых элементов матрицы.

В третьем алгоритме «Alltoall» матрица делится горизонтально по вычислительным узлам, но каждый узел после окончания счета рассылает всем узлам только требуемые каждому из этих узлов элементы. Соответственно, каждый узел получает из сети только нужные ему для счета элементы, а не весь вектор.

Приведем оценку объема коммуникаций для первого и третьего алгоритмов. В первом случае каждый узел получает в результате вычислений n/p элементов, которые рассылаются остальным узлам, откуда получаем оценку n(p-1) элементов результирующего вектора, где n — размер матрицы, p — количество вычислительных узлов. Видно, что общий объем коммуникаций растет пропорционально количеству вычислительных узлов, что накладывает ограничения на масштабируемость.

Для третьего алгоритма получаем, что в худшем случае каждому узлу необходимо nd/p элементов, где d — среднее количество ненулевых элементов в строке. Эта оценка получена из замечания, что каждый узел обладает n/p строками исходной матрицы, в каждой из которых в среднем имеется d ненулевых элементов, каждый из которых в худшем случае (если их горизонтальные индексы различны) требует один элемент результата предыдущей итерации. Далее, от каждого из оставшихся узлов в среднем требуется nd/p^2 элементов, если же не считать сам получающий узел (он уже обладает необходимой информацией), то от каждого узла будет получено $nd(p-1)/p^2$, а суммарный объем будет $nd(p-1)/p$. Для большого числа узлов $(p-1)/p$ близко к единице, откуда окончательно получаем оценку общего объема пересылок по сети nd. В итоге получаем, что общий объем пересылок в случае третьего алгоритма не зависит от количества вычислительных узлов.

Нетрудно видеть, что при p>d алгоритм «Alltoall» будет требовать меньше пересылок, чем первый алгоритм, причем с ростом числа узлов количество элементов, требуемое каждому узлу, уменьшается, что увеличивает возможности алгоритма по масштабированию.

3. Имитационная модель

Для оценки производительности на тестовых программах на языке Charm++ создана параллельная потактовая имитационная модель разрабатываемой коммуникационной сети 3D-тор [3]. Модель хорошо масштабируется при использовании большого количества узлов вычислительного кластера.

В модели реализована параметризованная обобщенная схема маршрутизатора разрабатываемой в ОАО «НИЦЭВТ» сети с топологией многомерный тор. Модель позволяет варьировать размеры буферов, число виртуальных каналов, инжекторов и эжекторов (входов/выходов из кроссбара для выдачи и приёма пакетов в/из сети), изменять алгоритмы работы блоков маршрутизации и арбитража. Входные виртуальные каналы одного направления могут подключаться к кроссбару либо напрямую, либо через входной арбитр.

При прохождении пакета через маршрутизатор используется конвейер, включающий стадии анализа заголовка, выбора дальнейшего маршрута, выставления запроса входному/выходному арбитру, ожидания ответа от арбитра, перемаршрутизацию/переход в другую виртуальную подсеть при выполнении определённых условий (таймаута ожидания, например). Задержки на выполнение каждой стадии могут быть явно заданы, так же как и параметры конвейеризации (предварительный арбитраж) и используемая стратегия арбитража.

К каждому маршрутизатору подключена упрощенная модель процессора (условно — ShmemPE), которая реализует как функции сетевого адаптера по формированию и интерпретации пришедших пакетов (для выполнения, например, чтений из памяти моделируемого узла), так и функции собственно вычислительного процессора, исполняющего некоторую вычислительную задачу. Интерфейс PCI-Express, связывающий процессор с адаптером, моделируется приближенно, учитывается практически только пропускная способность. Реальный процессор и память не моделируются вообще.

Для удобства программирования вычислительных задач для процессора ShmemPE имитируемая программа выполняет коммуникации при помощи функций интерфейса SHMEM. Эти функции разбивают сообщения на один или несколько сетевых пакетов (с учетом максимального размера сетевого пакета) и не возвращают управление в имитируемую программу до тех пор, пока не дождутся отправки сообщения в сеть в случае shmem_put или окончания выполнения барьера в случае shmem_barrier.

4. Реализация

Для имитации выполнения итерационного метода в реализациях выполняется подряд несколько итераций умножения разреженной матрицы на вектор.

На модели разрабатываемой коммуникационной сети были реализованы два алгоритма: первый, основанный на коллективной операции Allgather (реализованной программно, т. е. без аппаратной поддержки со стороны маршрутизатора), а также третий, основанный на операции Alltoall, с отправкой только необходимых участков результирующего вектора.

Для моделирования выполнения алгоритма «Alltoall» для случайно сгенерированной матрицы с параметрами n = 10240000, d = 10 и d = 50 был произведен расчет необходимого объема пересылок между каждой парой узлов (квадратная матрица размером $p \times p$, где p — количество моделируемых

вычислительных узлов (от 2 до 2048). Далее, производился запуск модели и выполнялись указанные отправки в режиме «по кольцу»: узел с номером j отправляет сообщение необходимого объема узлам с номерами j + 1, j + 2, …, np, 0, 1, …, j – 2, j – 1 в указанном порядке. Рассылка завершается барьерной синхронизацией, которая гарантирует получение каждым узлом данных, необходимых для начала очередной итерации.

Далее из результатов выполнения OpenMP-реализации теста на одном узле прототипа вычисляется среднее время, необходимое для выполнения пары операций (вещественное сложение и умножение во внутреннем цикле алгоритма) при фиксированном значении параметров n и d. Это упрощение связано с отсутствием модели процессора и памяти, что вынуждает считать зависимость времени, затрачиваемого на вычисление результата на одном узле, линейной относительно числа строк матрицы (при фиксированном размере плотного вектора). В итоге в предположении последовательного выполнения вычислений, а затем — коммуникаций, вычисляется итоговая производительность.

Для проверки адекватности модели производилось сравнение результатов моделирования прототипа коммуникационной сети с результатами на прототипе. Максимальное отклонение модели прототипа от прототипа составило 32%, среднее отклонение — 13% при моделировании первого алгоритма рассматриваемой задачи для разного набора параметров, что позволяет с большей надежностью относиться к результатам моделирования будущей системы.

Для сравнения результатов моделирования с результатами на существующих суперкомпьютерах первый и третий алгоритмы были реализованы при помощи MPI и OpenMP. В реализации первого алгоритма использовалась операция MPI_Allgather, третьего алгоритма — MPI_Alltoall.

5. Результаты моделирования и выполнения алгоритмов на суперкомпьютерах

На рис. 1 представлено ускорение работы алгоритмов на разных суперкомпьютерах и на модели, а также достигаемая максимальная абсолютная производительность. Выполнение тестов производилось на суперкомпьютере «Ломоносов» в НИВЦ МГУ (процессоры Intel Xeon X5570, интерконнект InfiniBand 4x QDR), а также на кластере IBM BlueGene/P, установленном на факультете ВМК МГУ. Параметры модели соответствуют характеристикам сети на основе кристалла маршрутизатора (ASIC).

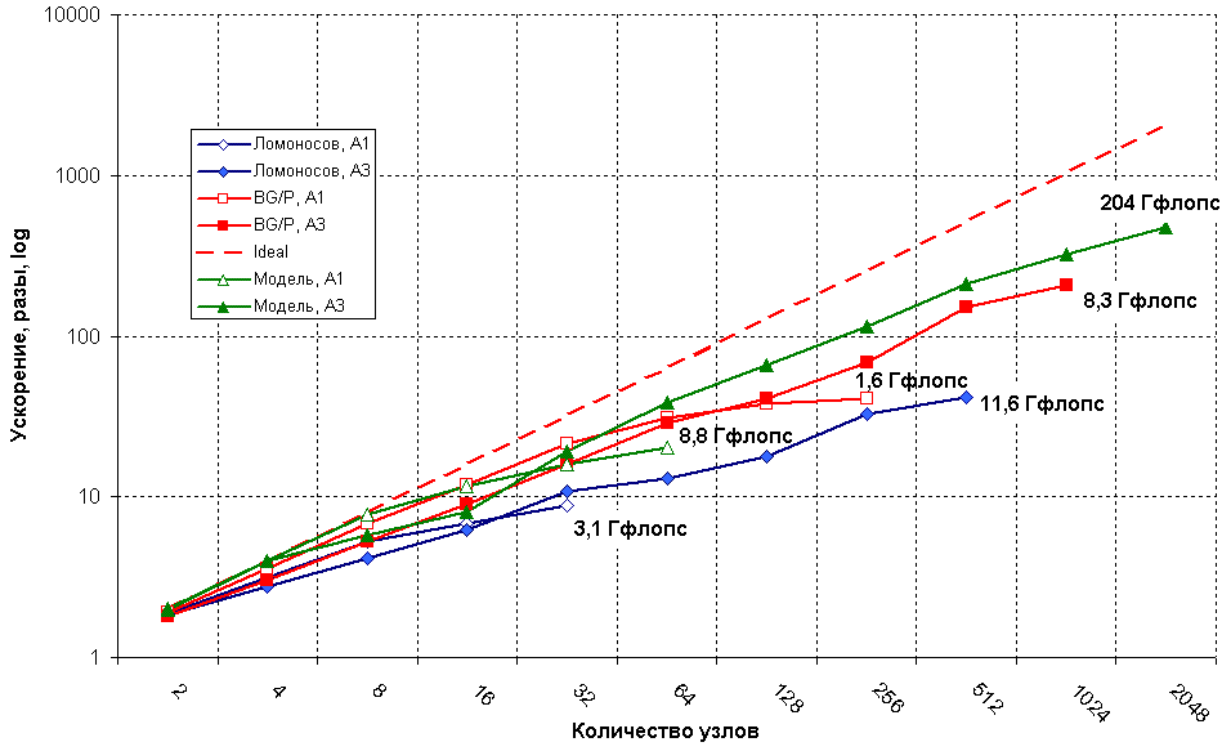

Рис. 1. Ускорение производительности (в логарифмической шкале) различных алгоритмов умножения разреженной матрицы на вектор (n=10240000, d=10) на суперкомпьютерах «Ломоносов» и IBM BlueGene/P, а также на модели разрабатываемой коммуникационной сети. A1 — первый алгоритм «Allgather», A3 — третий алгоритм «Alltoall».

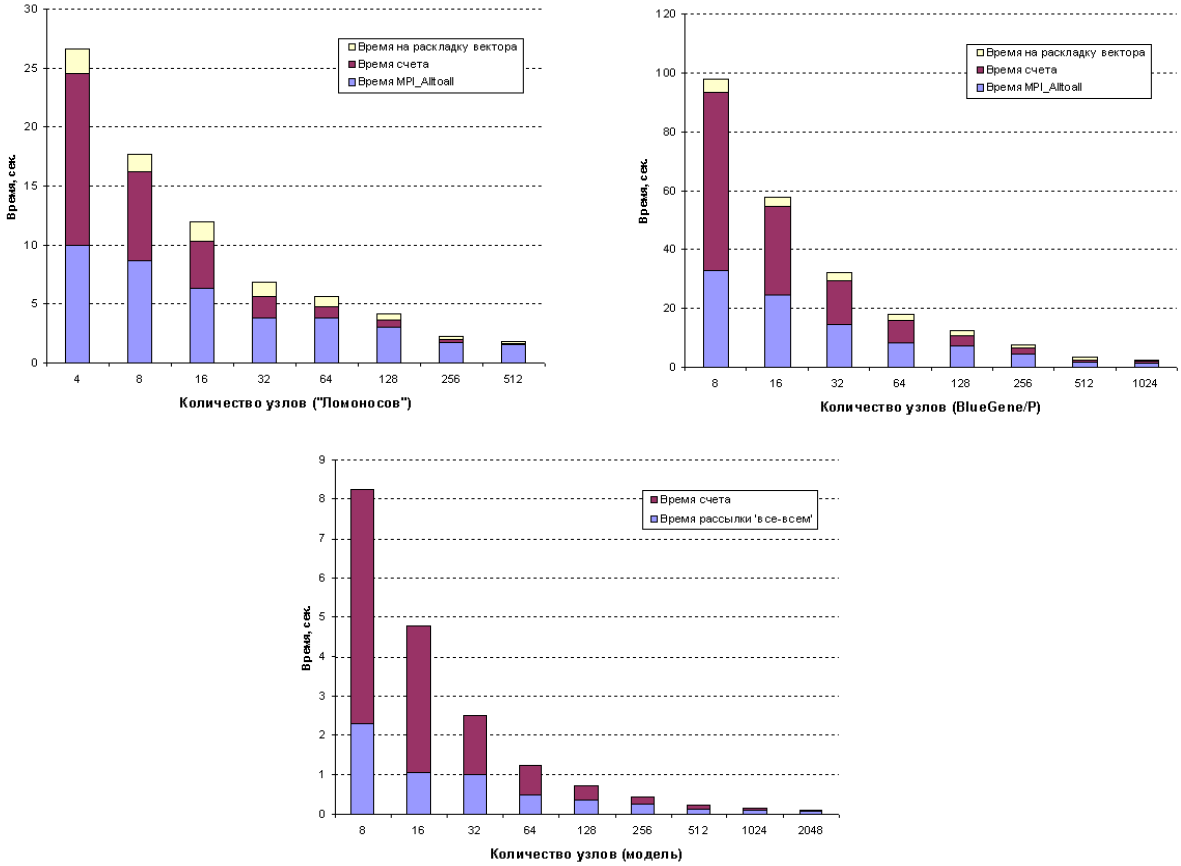

Рис. 2. Профиль выполнения 100 итераций третьего алгоритма «Alltoall» умножения разреженной матрицы на вектор (n=10240000, d=10) на суперкомпьютерах «Ломоносов», IBM BlueGene/P и на модели разрабатываемой коммуникационной сети.

На первом алгоритме все суперкомпьютеры и модель показывают плохую производительность и масштабируемость. Первый алгоритм опережает третий только на относительно небольшом числе узлов.

На третьем алгоритме для большого числа узлов ускорение на модели опережает ускорение на BlueGene/P, для 1024 узлов ускорение на модели составляет 323 раза, на BlueGene/P — 209 раз. Рост производительности на «Ломоносове» останавливается для 512 узлов, ускорение при этом составляет 41 раз.

На реальных суперкомпьютерах время выполнения вычислений в 100 итерациях третьего алгоритма (см. рис. 2) масштабируется почти линейно с ростом числа узлов, такое же допущение сделано и в реализации данной задачи для модели сети. В реализации с использованием MPI пришедшие данные раскладываются по своим местам в умножаемом векторе, однако на рисунке видно, что эта составляющая также хорошо масштабируется. Поэтому в третьем алгоритме все зависит от масштабирования рассылки «все-всем», которая на «Ломоносове» и BlueGene/P реализована при помощи коллективной операции MPI_Alltoall. Напомним, что суммарный объем сообщений по всем узлам в третьем алгоритме остается постоянным с ростом числа узлов. На «Ломоносове» время выполнения 100 операций MPI_Alltoall в рамках задачи падает от 10 секунд на 4 узлах до 1,5 секунд на 512 узлах, на BlueGene/P те же операции выполняются 43 секунды на 4 узлах, на 1024 узлах — 1,5 секунды. На модели сети на 4 узлах рассылка «все-всем» занимает 0,9 секунды, на 8 узлах — 2,3 секунды, на 1024 узлах — 0,1 секунды. Результаты показывают, что именно рассылка «все-всем» приводит к падению ускорения производительности алгоритма.

Результаты рассылки «все-всем» на модели ограничены теоретической пиковой пропускной способностью, которая равна пиковой бисекционной пропускной способности сети, что косвенно подтверждает правильность результатов.

Среди возможных причин лучшей масштабируемости результатов на модели по сравнению с BlueGene/P, в котором также используется сеть 3D-тор, можно назвать две. Во-первых, если на BlueGene/P, число узлов меньше 512, то используется не тор, а решетка, у которой бисекционная пропускная способность в два раза ниже, чем у тора. В модели для заданного числа узлов всегда используется тор. Во-вторых, инжекция и эжекция пакетов из сети в модели реализованы идеализированно, не моделируется память и процессор, что может повлечь завышенность оценки.

Тем не менее, полученные результаты показывают возможность получения высокой реальной производительности на задаче умножения разреженной матрицы на вектор на модели разрабатываемой коммуникационной сети.

6. Заключение

В статье при помощи имитационного моделирования показана возможность хорошего масштабирования производительности задачи умножения разреженной матрицы на вектор на модели коммуникационной сети, разрабатываемой в ОАО «НИЦЭВТ».

В ближайшем будущем планируется реализация в модели сети поддержки коллективных операций, исследование с их помощью различных вариантов реализации задачи умножения разреженной матрицы на вектор. Также планируется увеличить точность моделирования выдачи и приема сообщений через PCI-Express, рассматривается возможность использования моделей реальных процессоров с моделью коммуникационной сети.

ЛИТЕРАТУРА:

1. А.С. Симонов, И.А. Жабин, Е.Р. Куштанов, И.А. Аверин и др. Эскизно-технический проект ОКР «Разработка базовой технологии создания высокоскоростной отказоустойчивой коммуникационной сети с поддержкой глобально адресуемой памяти для перспективных суперкомпьютеров», 2010.
2. А.А. Корж, Д.В. Макагон, И.А. Жабин, Е.Л. Сыромятников и др. «Отечественная коммуникационная сеть 3D-тор с поддержкой глобально адресуемой памяти для суперкомпьютеров транспетафлопсного уровня производительности». // Параллельные вычислительные технологии (ПаВТ’2010): Труды международной научной конференции (Уфа, 29 марта – 2 апреля 2010 г.) [Электронный ресурс] – Челябинск: Издательский центр ЮУрГУ, 2010. — С. 227–237.
3. Л.К. Эйсымонт, А.С. Семенов, А.А. Соколов, А.С. Фролов, А.Б. Шворин «Моделирование российского суперкомпьютера «Ангара» на суперкомпьютере» // Суперкомпьютерные технологии в науке, образовании и промышленности: Сб. под редакцией академика В.А. Садовничего, академика Г.И. Савина, чл.-корр. РАН Вл.В. Воеводина. М.: Издательство Московского университета, 2009. — С. 145–150.
4. А.А. Корж «Распараллеливание задачи умножения разреженной матрицы на вектор на вычислительных кластерах с минимальной аппаратной поддержкой PGAS» // Параллельные вычислительные технологии (ПаВТ’2009): Труды международной научной конференции (Нижний Новгород, 30 марта – 3 апреля 2009 г.). – Челябинск: Изд. ЮУрГУ, 2009. — С. 813.