MPI Library and Low-Level Communication on the K computer

•N. Shida •S. Sumimoto •A. Uno
FUJITSU Sci. Tech. J., Vol. 48, No. 3, pp. 324–330 (July2012),

Ключ к повышению производительности приложений для массивных параллельных систем таких, как K компьютер, является повышение скорости обмена данными между вычислительными узлами. В K компьютере, этот межузловой обмен определяется библиотекой коммуникации и связи низкого уровня Message Passing Interface (MPI). Данная статья описывает реализацию и производительность библиотеки коммуникации MPI, которая использует новую архитектуру Tofu-интерконнекта, введенного в состав K компьютера, чтобы повысить производительность приложений Пета-масштаба, а также механизм связи низкого уровня, который выполняет управление тонкими особенностями интерконнекта Tofu.

1. Введение

Коммуникационная библиотека MPI (интерфейса обмена сообщениями) является хорошо устоявшимся стандартом для обмена сообщениями в системах параллельной обработки, однако не будет преувеличением сказать, что качество этой библиотеки коммуникации может существенным образом влиять на производительность системы в целом.

Это остается справедливым для K компьютера[note)] —для достижения производительности мирового уровня на 80 000 узлах крупномасштабной системы параллельной обработки должны применяться различные творческие мероприятия при использовании этой библиотеки. В частности, для повышения производительности, важно свести к минимуму время связи между узлами. Таким образом, при реализации коммуникационной библиотеки MPI в дополнение к обычно используемой коммуникационной библиотеке MPI, важно также предусмотреть механизм низкоуровневой коммутации для управления интерконнектом Tofu.

Данная статья описывает библиотеку коммуникации MPI и механизм низкоуровневого обмена данными для K компьютера.

2. Цели развития и проблемы реализации библиотек коммуникации MPI

Ниже приводятся цели развития и проблемы реализации библиотек коммуникации MPI. Эти цели не ограничиваются K компьютером - они также могут применяться при проектировании суперкомпьютеров класса exa-Flops, которые в 100 раз производительнее K компьютера.

1) MPI с высокой основной производительностью

Основные коммуникационные операции в суперкомпьютере включают коммуникации точка-точка, групповые коммуникации и операции ввода-вывода MPI, производительность которых может иметь влияние на производительность системы в целом. В частности, производительность обмена данными точка-точка, которая входит в производительность группового обмена данными, а также операций ввода-вывода MPI, требует оптимальной системы обмена данными, которая могла бы "выжать" наивысшую производительность из интерконнекта Tofu с точки зрения латентности и пропускной способности.

2) Оптимальный обмен данными для среды больших масштабов.

Производительность системы, как правило, пропорциональна количеству используемой при коммуникации памяти, что означает, существование компромисса между минимизацией объема памяти, используемой в системе, и достижением высокой производительности. Кроме того, использование памяти увеличивается пропорционально количеству процессов, с которыми конкретный процесс должен общаться, что означает высокую вероятность нехватки доступной памяти в системе уровня 80 000-узлов. Важным вопросом становится следующий: как

note) "K computer" - английское название, которое RIKEN использовал для суперкомпьютера в данном проекте начиная с июля 2010. "K" пришло из японского слова "Kei", которое обозначает 10 пета или 10 в 16й степени.

достичь минимизации использования памяти с учетом вышеизложенных ограничений. Короче говоря, существует потребность в системе обмена данными, которая каким-то образом минимизирует объем используемой памяти.

3) Дружественное пользователю окружение.

Интрконнект Tofu, используемый в K компьютере, имеет топологию 6D тора.[1)] Для обычного пользователя понимание его физической связности не является простой задачей. Поэтому необходимо провести исследования: как обойти неисправные узлы при работе масштабной системы с 80 000 узлами.

3. Обзор реализации библиотеки коммуникации MPI

Политика, которую мы взяли на вооружение для реализации библиотеки коммуникации MPI при решении задач, описанных в предыдущем разделе, заключается в предоставлении библиотеки коммуникации низкого уровня с учетом, насколько это возможно, стандартного API (интерфейса прикладного программирования) на основе библиотеки коммуникации MPI с открытым исходным кодом и достижением посредством этой библиотеки коммуникации низкого уровня максимально специфических для K компьютера функций настолько глубоких, насколько это возможно. Это будет означать эффект минимизации изменений по отношению к стандартной библиотеке коммуникаций MPI.

Для данной цели мы адаптировали Open MPI, являющийся открытой реализацией MPI. В первую очередь мы выбрали Open MPI, поскольку он имеет добрый послужной список для SPARC процессоров, используемых в K компьютере. К тому же он поддерживает InfiniBand- главный интерконнект для кластеров ПК, что, таким образом, упрощает процесс разработки.

Библиотека коммуникации для K компьютера на основе вышеуказанной политики решает сформулированные проблемы в следующих случаях:

1) MPI с высокой основной производительностью

Во-первых, для улучшения производительности точка- точка мы предоставляем API коммуникации, сосредоточенные вокруг обмена данных RDMA (удаленного прямого доступа к памяти), который может в полной мере использовать характеристики интерконнекта Tofu на уровне библиотеки коммуникации нижнего уровня, а также мы используем этот API для реализации библиотеки коммуникации MPI.

Во-вторых, для повышения производительности группового обмена данными, мы используем несколько сетевых интерфейсов архитектуры интерконнекта Tofu на основе обмена данными точка-точка сосредоточенном вокруг коммуникации RDMA и адаптируем алгоритм групповой обмена данными применительно к топологии 6D тора.

2) Оптимальная коммуникация для крупномасштабной среды.

Мы предприняли два мероприятия для достижения высокой производительности обмена данными и максимально возможного использования наименьшего уровня памяти. Первое мероприятие заключалось в сведении к минимуму количества памяти, необходимого для коммуникационного буфера с использованием коммуникации, сосредоточенной вокруг RDMA, и второе заключалось в фиксации количества поддерживаемых процессов, с которым отдельно взятый процесс может осуществлять одновременный обмен данными.

3) Дружеский пользователю интерфейс.

Для K компьютера топология 6D тора представляется виртуальным 3D тором, для того, чтобы сделать его более легким для работы пользователя с 6D топологией. Это достигается путем объединения шести осей топологии 6D таким образом, чтобы организовать 3D конфигурацию. В результате, становится легче переносить приложения, разработанные ранее для существующих систем 3D-тора, и существует больше настраиваемых форм 3D-торов, из которых можно осуществлять выбор. Такая 3D конфигурация имеет результатом ряд полезных свойств. Например, система может использоваться как система 3D-тора, даже если она разделена на множество рабочих мест, к тому же приложения могут выполняться без беспокойства о присутствующих дефектных узлах в системе посредством соответствующих установок в каналах связи.

4. Реализация Open MPI на K компьютере

Структура MPI, разработанного для реализации на K компьютере представлена на рисунке 1. Для поддержки связи с низкой латентностью и групповым доступом на основе RDMA в базовую структуру Open MPI были внесены некоторые изменения. Для данной реализации Open MPI на K компьютере были установлены следующие требования:

1) Напоминание об обновлении на новую версию.

Библиотека с открытым исходным кодом MPI постоянно обновляется для размещения новых функции, исправления ошибок и т.д. Действительно, существует высокая вероятность того, что новые функции, которые будут добавлены в MPI версию 3.0, которая в настоящее время изучается на форуме MPI, приведет к серьезным изменениям в библиотеке. По этой причине уделяется большое внимание реализациям новых версий Open MPI, которые могут быть осуществлены путем применения патчей без изменения структуры Open MPI, если это возможно.

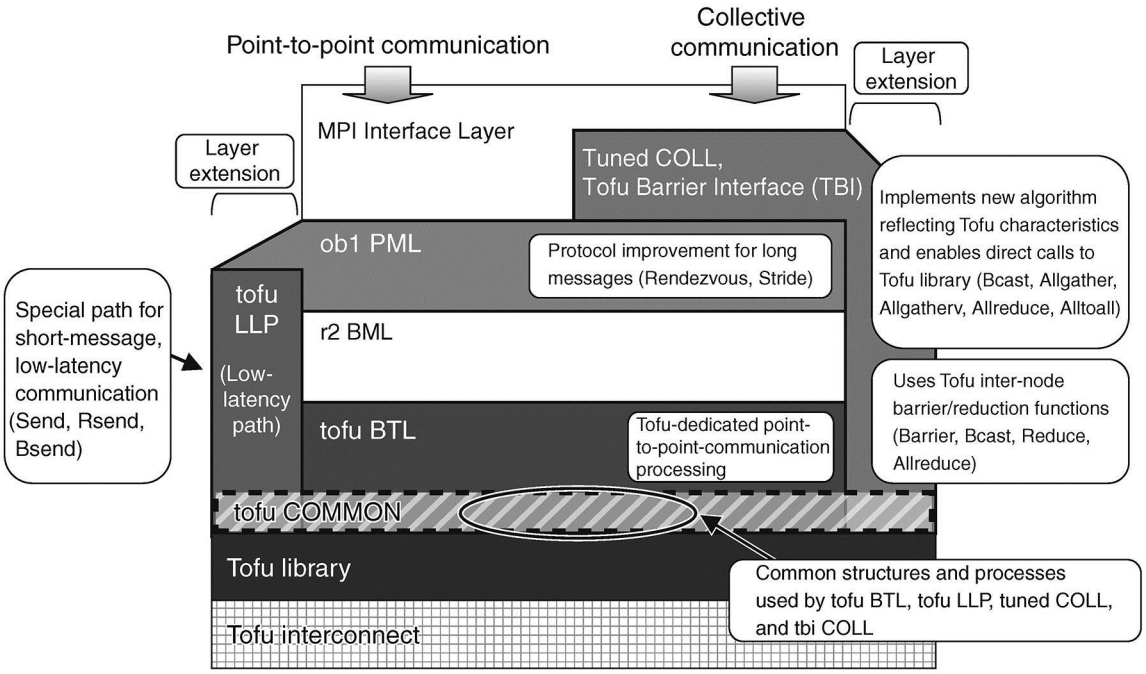

Рисунок 1
Структура MPI К компьютера

2) Концентрация на обмене данными с низкой латентностью.

Текущая реализация Open MPI для осуществления связи точка-точка представлена тремя уровнями коммуникационных библиотек (COLLs) - обмена сообщениями точка-точка (PML), уровень управления BTL (below-the-line, BML) и уровень перемещения байт (BTL). Это означает, что для выполнения такой связи по крайней мере три вызова функций должны быть сделаны, что, тем самым, увеличивает латентность. Чтобы достичь аппаратной производительности, необходимой для библиотеки обмена данными MPI К компьютера, для связи точка-точка был добавлен путь с низкой латентностью (LLP) в качестве выделенного ярлыка.

3) Групповой доступ, основанный на прямом обращении к памяти (RDMA).

Часть алгоритма групповой коммуникации использует недавно разработанную структуру групповой коммуникации вместо стандартной структуры Open MPI для того, чтобы:

- Осуществлять аппаратную поддержку Tofu (Barrier, Bcast, Reduce, Allreduce)
- Поддерживать основанную на RDMA групповую коммуникацию
- Маркировать и управлять множественными сетевыми интерфейсами.

Короче говоря, мы разработали алгоритм группового обмена данными, позволяющий нескольким сетевым интерфейсам, которые будут использоваться в частности для часто выполняемых коммуникационных операций Bcast, Allgather, Allreduce и Alltoall, а также минимизирует столкновения пакетов в сети.

Благодаря установлению двух режимов связи MPI был снижен объем памяти, необходимой для коммуникации: высокоскоростной (high-speed) и экономящий память (memory-saving). Коммуникация начинается в режиме экономии памяти, который поддерживает небольшой размер буфера памяти для коммуникации. Однако, если количество коммуникаций с другим конкретным процессом превышает определенное пороговое значение, то режим переключается на высокоскоростной, использующий буфер коммуникации большего размера. Существует также механизм, предотвращающий чрезмерное потребление памяти путем ограничения количества переключений на высокоскоростной режим.

5. Реализация низкоуровневого обмена данными на К компьютере

Механизм низкоуровневой коммуникации для К компьютера получает максимальную производительность интерконнекта Tofu путем использования библиотек низкого уровня интерконнекта Tofu. Этот механизм обеспечивает два вида функций.

1) Низкоуровневая коммуникация.
Это основанные на RDMA функции коммуникации для получения полной аппаратной производительности интерконнекта Tofu.

2) Составление карт категорий.
Эта функция представляет 3D тор после удаления неисправных узлов на основании информации, получаемой от интерфейса работ.

6. Оценка основных характеристик производительности коммуникации

Для оценки основных характеристик коммуникации К компьютера мы вначале оценили производительность обмена данными точка-точка для ближайших соседей связи как на уровне библиотеки Tofu, так и на уровне библиотеки MPI. Затем мы оценили производительность группового обмена данными как аппаратных, так и программных средств, используя операции коммуникации Allreduce и Barrier.

6.1 Производительность обмена данными точка-точка на уровне библиотек Tofu

Латентность одностороннего обмена данными для связи ближайших соседей на уровне библиотеки Tofu была 0,92мкс для сообщения 8байт, а максимальная пропускная способность 4,76ГБ/с для сообщения 16 МБ. Учитывая, что аппаратная задержка 0,91 мкс, можно видеть, что задержка обработки связи на уровне библиотеки Tofu была довольно низкой- 0,01мкс. Производительность коммуникации для пропускной способности 1-4 сетевых интефейсов Tofu (TNIs) показана на рисунке 2. Эти результаты показывают, что производительность может быть улучшена на 3TNI масштабируемым образом до 14,26Гбайт/с. Ограничение производительности в 15,03ГБ/с для 4TNI объясняется эффектом бутылочного горлышка в интерфейсе со стороны CPU.

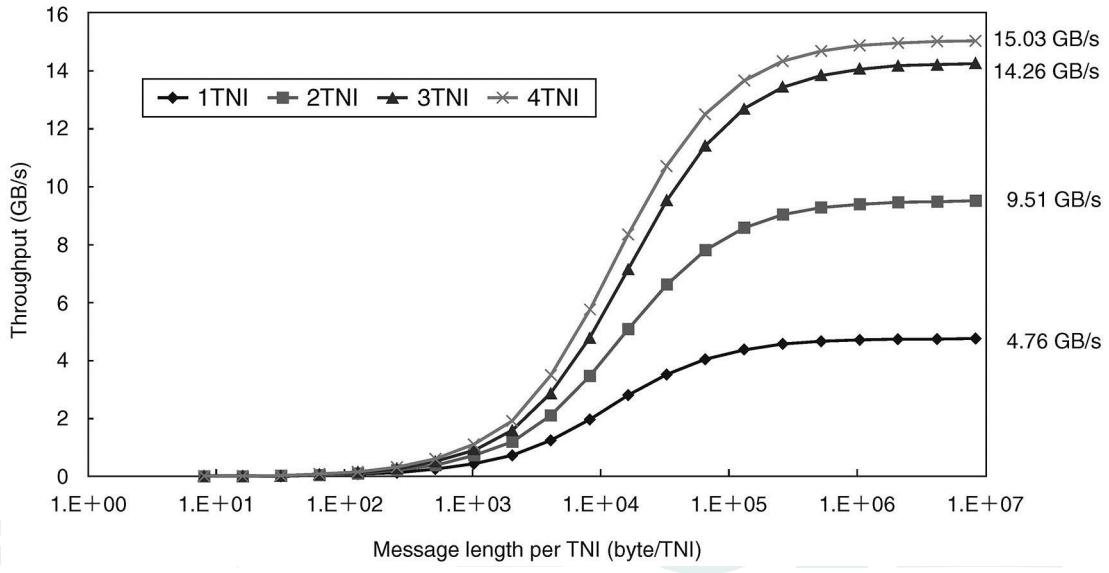

Рисунок 2
Значение пропускной способности обмена данными 1–4 TNIs на уровне библиотеки Tofu

6.2 Производительность обмена данными точка-точка на уровне библиотек MPI

Односторонняя латентность на уровне библиотеки MPI показана на рисунке 3, а производительность полосы пропускания обмена данными показана на рисунке 4. По сравнению с предыдущим рисунком латентность обмена данными на уровне библиотеки MPI составляет 1,27мкс для сообщения 8байт в случае ближайших соседей по обмену и, как следует из последнего рисунка, максимальная пропускная способность обмена данными составила 4,7ГБ/с для сообщения 16МБ. Из приводимых результатов видно, что аппаратная производительность интерконнекта Tofu загружается до максимума.

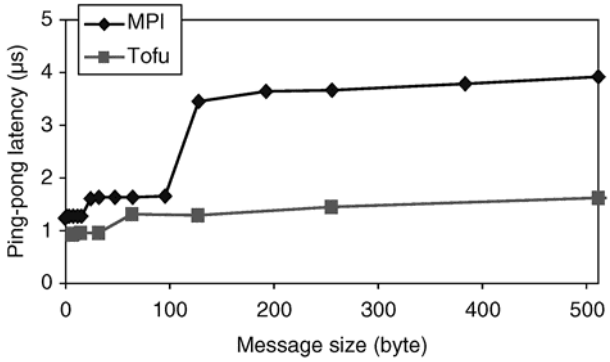

Рисунок 3
Латентность для одностороннего обмена на уровне библиотеки MPI

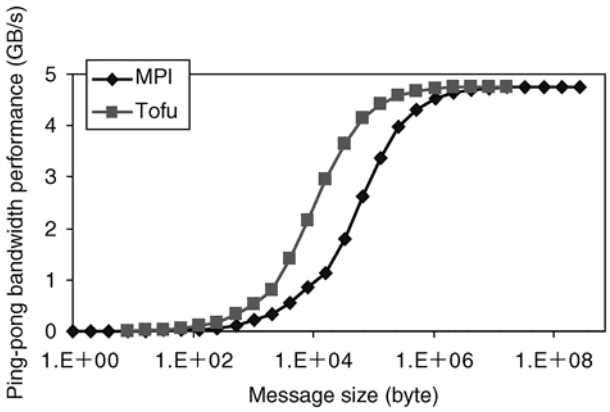

Рисунок 4
Значение пропускной способности на уровне библиотеки MPI

6.3 Производительность группового обмена данными на уровне библиотек MPI

Групповые характеристики обмена данными для аппаратной реализации с использованием барьерных интерфейсов Tofu (TBIs) на уровне библиотеки MPI показано на рисунке 5. В частности, эта цифра показывает результаты для операций Allreduce и Barrier до 9216 узлов и сравнивает полученные результаты с аналогичными, полученными программной реализацией. Рассматривая результаты аппаратной реализации, можно увидеть, что практически нет никакого ухудшения производительности даже при обмене данными между 9216 узлами, демонстрируя, что производительность была стабильной.

Значение пропускной способности операции Allreduce на уровне библиотеки MPI показана на рисунке 6. В частности, на рисунке показаны результаты для алгоритма коллективного обмена данными (Trinaryx3), который использует несколько сетевых интерфейсов, что раскрывает полные возможности производительности интерконнекта Tofu и который предотвращает совмещение каналов связи. Алгоритм обмена данными Trinaryx3 Allreduce, разработанный для K компьютера использует три сетевых интерфейса и достигает производительности обмена данных 7,1ГБ/с, что примерно в 5 раз выше, чем для двух существующих алгоритмов группового обмена данными (Ring и Recursive Doubling), которые также показаны для сравнения.

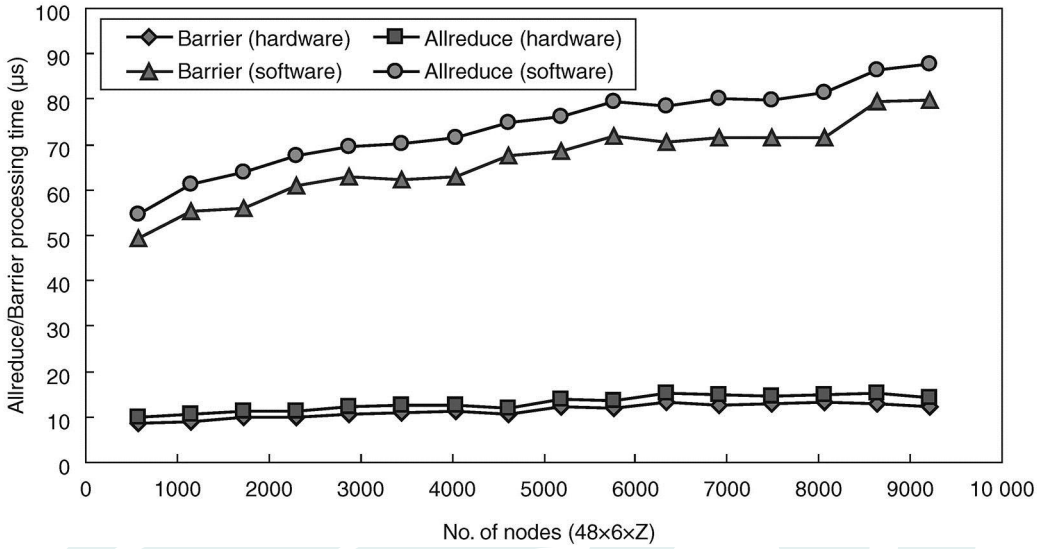

Рисунок 5
Аппаратная производительность коллективного доступа на уровне библиотеки MPI

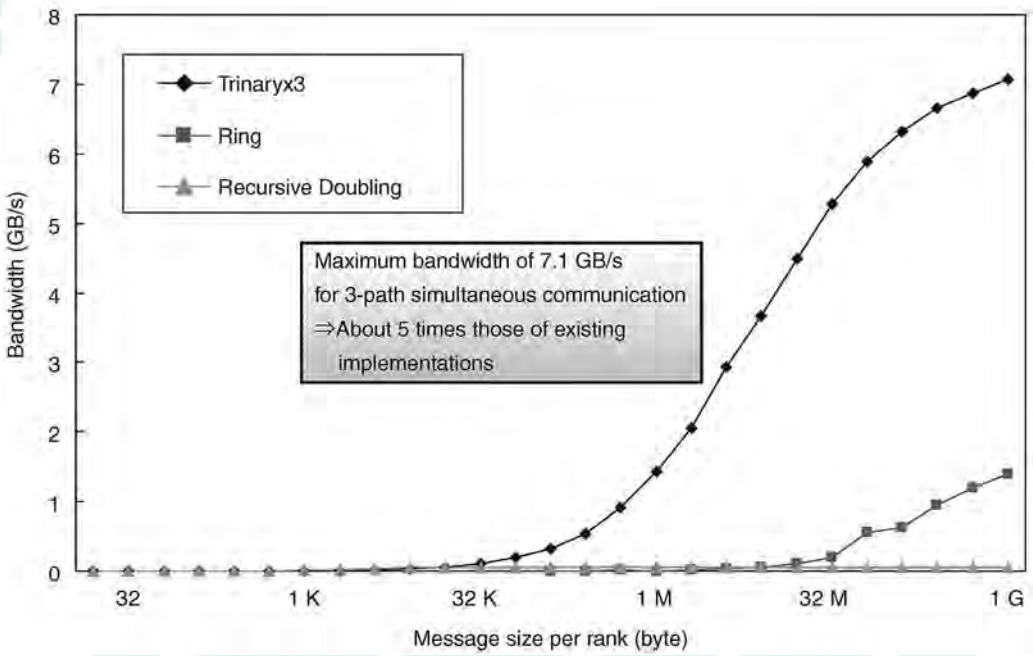

Рисунок 6
Значение пропускной способности Allreduce на уровне библиотеки MPI

7. Выводы

Данная статья описывает реализацию и производительность MPI и механизма низкоуровневого обмена данными, представляемого K компьютером. Были установлены три цели разработки для данной реализации MPI: MPI с высокой основной производительностью, оптимальный обмен данными для крупномасштабной среды и дружественное пользователю окружение. Для достижения этих целей были приняты разнообразные творческие мероприятия, и результатом стали высокопроизводительный обмен данными и простота в использовании. Тем не менее, существует еще много возможностей для совершенствования на ультра-крупномасштабной системе, подобной K компьютеру. В будущих исследованиях мы ожидаем дальнейшее увеличение производительности и внесения своего вклада в сообщество Open MPI.

Ссылки

1) Y. Ajima et al.: A 6D Mesh/Torus Interconnect for Exascale Computers. IEEE Computer, Vol. 42, No. 11, pp. 36–40 (2009).