Операции типа «точка-точка» интерфейса MPI для СКСН «Ангара»

Д.Л. Аверичева, М.В. Кудрявцев

ОАО «НИЦЭВТ», Москва

avericheva@nicevt.ru, mkudryavtsev@nicevt.ru

Введение

В ОАО «НИЦЭВТ» ведется проект создания суперкомпьютера с мультитредово-потоковой архитектурой и аппаратной поддержкой распределенной общей памяти (СКСН «Ангара»)[1], предназначенного для эффективного решения задач с интенсивной нерегулярной работой с памятью и не уступающего традиционным суперкомпьютерам при решении задач другого типа. В данной работе исследуется возможность эффективной реализации на этом суперкомпьютере широко используемой библиотеки передачи сообщений MPI [2].

Реализация библиотеки MPI на распределенной памяти, как правило, подразумевает использование большого объема оперативной памяти $O(N^2)$, где N- количество MPI-процессов, сложные схемы синхронизации чтения и записи данных. Далее мы покажем, что с помощью специальной команды RemoteProcedureCall (RPC) запуска с узла-отправителя на заданном узле-получателе треда с передачей ему нескольких аргументов можно избежать некоторых недостатков традиционных реализаций библиотек MPI на распределенных системах [4]. Кроме того, показывается, что при эффективном использовании достаточно простых возможностей мультитредового процессора и простой по реализации сети можно получить результаты сравнимые с результатами, полученными на распространенных реализациях библиотеки MPI с использованием сложных сетевых интерфейсов.

Архитектура СКСН «Ангара»

В рамках российского проекта создания СКСН «Ангара» разрабатывается оригинальный мультитредово-потоковый микропроцессор и маршрутизатор коммуникационной сети. Суперкомпьютер содержит множество вычислительных узлов, соединенных коммуникационной сетью с большой суммарной пропускной способностью при передаче коротких пакетов. Вычислительный узел (далее просто узел) содержит: сетевой адаптер/маршрутизатор; многоядерный мультитредово-потоковый микропроцессор с аппаратной поддержкой трансляции адресов глобально адресуемой памяти и передачи коротких системных пакетов, реализующих такие обращения; локальную память с большим расслоением на базе стандартных DRAM-модулей. В исследованиях рассматривалась базовая конфигурации мультитредового микропроцессора J7-2[1], его параметры представлены в таблице 1.

В частности, для понимания излагаемого далее материала необходимо пояснить некоторые особенности организации памяти СКСН «Ангара», отображения (распределения) адресов сегментов глобально адресуемой виртуальной памяти на локальную физическую память каждого из узлов [3].

Во-первых, возможно использование отображения виртуальных адресов на физические, когда номер узла задается старшими битами виртуального адреса. Такое отображение называется блочным. В данном случае сегмент распределяется равными порциями (блоками) по узлам, на которые этот сегмент распределен: в физической памяти первого узла располагается первый блок подряд идущих виртуальных адресов, в физической памяти второго узла – второй блок, на последнем узле – последний блок.

Во-вторых, каждая 64-разрядная ячейка памяти СКСН «Ангара» имеет дополнительный теговый f/e-бит состояния, а адреса ячеек имеют указатель режима доступа к памяти. Выполнение операций с памятью зависит от режима доступа и битов состояния адресуемой ячейки. В обычном режиме f/e-бит не влияет на работу. В режиме

synchronize чтение из ячейки со значением f/e-бита empty не производится, аппаратура обеспечивает ожидание треда без выдачи им команд, до тех пор, пока ждущая команда не увидит значение f/e-бита full . В этом случае она меняет состояние обратно на empty и возвращает значение, а тред продолжает работать. Этот механизм обеспечивает удобную и эффективную синхронизацию тредов. Перед началом работы f/e-биты инициализируется значением empty.

Таблица 1. Основные характеристики базовых конфигураций многоядерных мультитредово-потоковых микропроцессоров, используемых в исследованиях

Параметр	J7-2
Частота, ГГц	0.5
Число ядер/тредов	2/64
Количество команд, выдаваемых в ядре на выполнение за такт	4
Размер кэша, МБ/way	1/4
Пропускная способность кэш-памяти, ГБ/сек	64
Пропускная способность DRAM-памяти, ГБ/сек	25.6
Дуплексная пропускная способность линка сети 4D-тор, ГБ/сек	4

В системе команд СКСН «Ангара» имеется команда RemoteProcedureCall (RPC) запуска с узла-отправителя на заданном узле-получателе треда с передачей ему нескольких аргументов. Запускаемый тред на узле-получателе выбирается из нескольких заранее выделенных тредов для обслуживания RPC. Если тред невозможно по какой-то причине запустить, то на узел-отправитель в регистр результата записывается соответствующее значение кода возврата.

Для оценки производительности СКСН «Ангара» на тестовых оценочных программах в ОАО «НИЦЭВТ» на языке Charm++ разработана параллельная потактовая имитационная модель, которая хорошо масштабируется по производительности при использовании до 512 узлов суперкомпьютера МВС-100k, имеющегося в МСЦ РАН. Отрабатываемые в модели временные диаграммы команд работы с памятью и работы сети максимально приближены к реальным.

Библиотека MPI для СКСН «Ангара»

Соглашения

В реализации MPI для СКСН «Ангара» MPI-процесс представляется доменом защиты микропроцессора J7, т.е. набором дескрипторов данных, команд и ресурсов задачи [3]. В одном мультитредовом ядре J7, например, доступны 4 домена защиты, один из которых обязательно занят операционной системой. Все треды домена защиты делятся на две категории. Треды, непосредственно выполняющие MPI-программу, будем называть пользовательскими. Треды, выполняющие специальные процедуры библиотеки MPI (очистку буфера, копирование очень длинного сообщения), будем называть служебными. Соотношение служебных и пользовательских тредов может быть изменено пользователем, по умолчанию, в одном MPI-процессе служебных тредов должно быть не более половины всех тредов. Служебные треды создаются один раз при вызове функции MPI_Init, затем ожидают вызова RPC. По умолчанию, в каждом MPI-процессе только один пользовательский тред. Для создания дополнительных пользовательских тредов можно пользоваться специальной библиотекой. Треды отображаются на аппаратные тредовые устройства мультитредового ядра посредством библиотеки времени выполнения – run-time системой. В одном мультитредовом ядре микропроцессора J7 имеется 64 тредовых устройства.

Библиотека MPI должна удовлетворять требованиям стандарта, поэтому сообщения одного треда одного MPI-процесса, отправленные одному и тому же MPI-процессу,

должны быть получены в том порядке, в котором были отправлены. Сообщения от разных тредов одного MPI-процесса могут приходить в любом порядке.

Структуры данных

В локальной памяти каждого MPI-процесса находится только один буфер входных сообщений и счетчик для него. Счетчик буфера - 64-разрядное слово, состоящее из двух 32-разрядных частей: количество записанных и прочитанных (head) и записанных-и-непрочитанных (tail) сообщений соответственно. Увеличение head и tail происходит с помощью атомарных операций. В результате её выполнения становятся известны также старые значения старые значения обоих счетчиков. Эти счетчики используются для вычисления индекса в буфере. Буфер сообщений является циклическим. Т.к. мы получаем одновременно значения обоих счетчиков, то оказывается известным, есть ли место для нового сообщения.

head	**tail**
32 бита	32 бита

Буфер состоит из одинаковых элементов. Элемент включает следующие поля: поле valid, действительности элемента (64 бита); header заголовок MPI сообщения (2x64 бит), указателя на специальную внутреннюю переменную MPI_Request (64 бита); тело сообщения либо указатель на сообщение data (3x64 бита). Если значение поля valid 0, то ячейка свободна, если 1 – занята. Сообщения последовательно добавляются в конец (tail) и считываются из начала в произвольном порядке. Из элемента буфера можно считать сообщение, только если f/e-бит поля valid имеет значение full. В элемент буфера можно записать новое сообщение, если f/e-бит поля valid имеет значение empty.

valid	**header**	**header**	**p_MPI_Request**	**data**	**data**	**data**

Виды сообщений

Сообщение будем называть коротким, если его длина не более трех слов (24 байта). Такое сообщение полностью копируется в буфер приема сообщений. Сообщение будем называть длинным, если его длина больше трех слов. В соответствующую ячейку буфера сообщений заносится указатель на тело сообщения. Тело сообщения копируется непосредственно из буфера процесса-отправителя в буфер процесса-получателя. Сообщение будем называть очень длинным, если его длина больше 28 слов. Очень длинные сообщения будем резать на куски размером по 28, 210 или 213 слов и копировать несколькими тредами параллельно непосредственно из буфера MPI-процесса отправителя в буфер процесса-получателя.

Механизм передачи короткого сообщения

При добавлении сообщения в буфер атомарно увеличиваем счетчик tail, в возвращаемом значении последние 32 бита будут равны его предыдущему значению. Таким образом, фактически получаем адрес свободной ячейки одновременно с увеличением значения счетчика. Для добавления записи в буфер на узле, в локальной памяти которого буфер находится, выполняем при помощи RPC функцию добавления записи на служебном треде. Заголовок передаваемого сообщения и само сообщение передаем треду, как параметры вызова RPC. Сообщения записываются в буфер в том порядке, в котором были отправлены. Запись в буфер производится в режиме synchronize, после записи у поля valid ячейки устанавливается значение f/e бита full.

Чтение сообщений из буфера производится в режиме synchronize. В том случае, если тред проверил ячейку буфера, он выставляет теговый f/e бит поля valid в значение full. Значение поля valid равно 1, если заголовки совпали, и 0, если не совпали. Затем, если сообщение не найдено, тред считывает значение из следующей ячейки буфера. Если тред полностью проверил буфер, но не нашел нужного ему сообщения, то он пы-

тается считать значение ещё не заполненной ячейки (значение f/e бита empty) в режиме synchronize, до тех пор пока не придет новое сообщение.

Кроме того, при чтении первой ячейки буфера мы будем блокировать счетчик сообщений. Если сообщение то, которое мы искали, то мы увеличиваем значение head и снимаем блокировку со счетчика. Если сообщение нам не подходит, то переходим к следующему сообщению и снимаем блокировку счетчика. При этом счетчик не заблокирован для записи новых сообщений в буфер, а только для получения.

Операции типа точка-точка

Функции MPI_Send и MPI_Recv в точности выполняют описанный выше алгоритм. MPI_Isend и MPI_Irecv выполняются аналогично, но отдельно создаваемыми служебными тредами. Для MPI_Isend создается дополнительный тред у MPI-процесса-получателя. Для MPI_Irecv создается дополнительный тред у того же MPI-процесса, который вызывал функцию MPI_Irecv.

Результаты

Библиотека MPI для СКСН «Ангара» реализована на языке Си с ассемблерными вставками. Оценочные тесты выполнялись на имитационной модели СКСН «Ангара». На рисунке 1 представлена пропускная способность на тесте PingPong исследуемой библиотеки в сравнении с пропускной способностью библиотеки HP-MPI версии 2.02 на сети Infiniband DDR (дуплексная пропускная способность 2Gbytes/sec), процессор AMD Opteron 8431 «Istanbul» 2,4GHz. Приведенные результаты показывают, что за счет высокой мультитредовости и высокой пропускной способности сети, достигаемой за счет передали пакетов с высокой конверизацией, возможно получение хороших характеристик передачи сообщений через функции MPI.

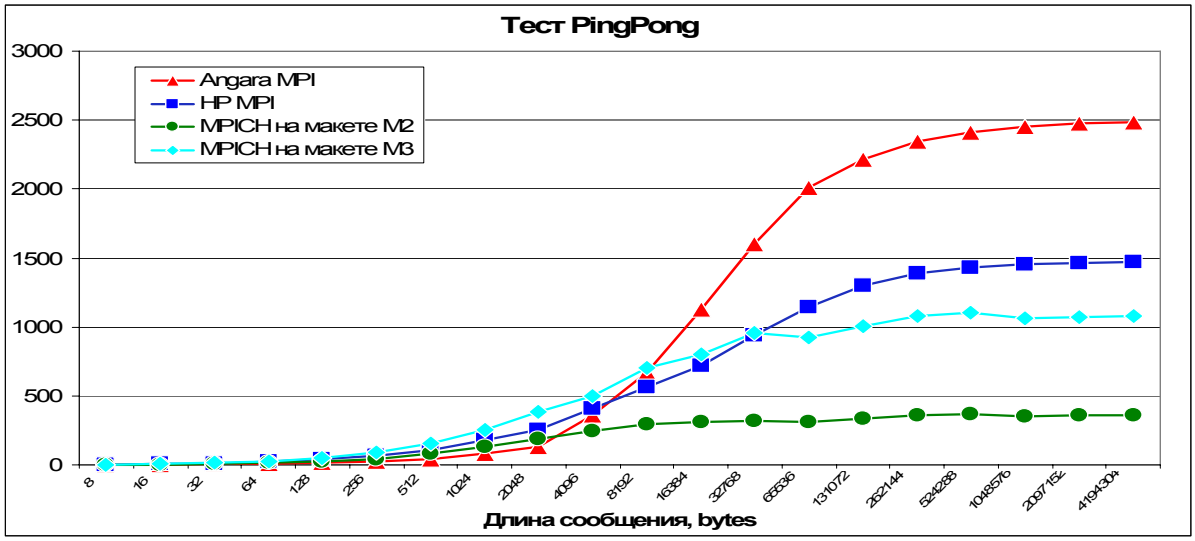

Рисунок 1. Пропускная способность на тесте PingPong.

Видно, что при длине сообщений от 4К исследуемая реализация MPI заметно выигрывает по производительности благодаря использованиию множества тредов, одновременно выполняемых в мультитредовом процессоре. До этого размера по описанному протоколу передачи пока используется однотредовая реализация.

Реализуемость сети с высокой конверизацией доказывается на макетах M2 и M3, разработанных в ОАО «НИЦЭВТ». На рисунке 1 приведены данные по библиотеке MPICH-2 версии 1.1.1 на этих макетах. Макет M2 состоит из шести узлов, соединенных сетью 2D-тор (размерность 3x2), линк 6 Gbits/sec, процессор Pentium 4 3.0 GHz, PCI-Express 4x, реализован в ПЛИС Virtex4. Макет M3 состоит из двух узлов, соединенных сетью 2D-тор, линк 13 Gbits/sec, PCI-Express 8x, реализован в ПЛИС Virtex5.

Заключение

Рассмотрена реализация библиотеки MPI на основе удаленного вызова функций для СКСН «Ангара» с глобально-адресуемой памятью. В данной реализации промежуточные буферы для хранения сообщений занимают не более чем O(N) места оперативной памяти (N-количество взаимодействующих MPI-процессов); передача длинных сообщений с помощью операций Send и Recv выполняется без излишних копирований, если этого не требует стандарт; используются возможности параллелизма мультитредового процессора для эффективной передачи сообщений; обеспечено совмещение передач данных с вычислениями, за счет выполнения операций неблокирующего приема/передачи данных в специально создаваемых тредах.

Литература

1. Слуцкин А., Эйсымонт Л. Российский суперкомпьютер с глобально адресуемой памятью // Открытые системы. – 2007. – №9. – С. 42–51.
2. “MPI: A Message-Passing Interface Standard,” Proc. Int’l J. Supercomputer Applications and High Performance Computing, MIT Press, 1994, pp. 159-416.
3. Семенов А. С., Соколов А.А., Эйсымонт Л. К. Архитектурные особенности и реализация глобально адресуемой памяти мультитредово-потокового суперкомпьютера. // Электроника: Наука, Технология, Бизнес. – 2009. – №1. – С. 50-61.
4. Patrick Geoffray, “A Critique of RDMA”, Myricom Inc, August 18, 2006.