НЕБЛОКИРУЮЩИЕ КОЛЛЕКТИВНЫЕ КОММУНИКАЦИОННЫЕ ОПЕРАЦИИ В NUMGRID

М.А. Городничев

1.Введение

В работе представлена реализация неблокирующих коллективных коммуникационных операций в проекте NumGRID.

1.1. NumGRID

Программный комплекс NumGRID [1,2] предназначен для объединения разнородных вычислительных кластеров в единый вычислительный ресурс на основе реализации общей коммуникационной среды для процессов параллельных программ, разработанных на основе стандартов MPI [3]. NumGRID позволяет исполнять MPI-приложение так, чтобы процессы приложения были распределены по рабочим узлам нескольких кластеров. Все распределенные процессы одного приложения объединяются в коммуникатор MPI_COMM_WORLD и имеют возможность обмениваться информацией между собой средствами MPI.

1.2. Неблокирующие коммуникационные операции MPI

В грядущем стандарте MPI-3.0 [4] предполагается введение неблокирующих коллективных операций. Известно [5], что во времени исполнения программ численного моделирования существенную долю занимают коллективные коммуникации такие как сбор данных со всех процессов в один (MPI_Reduce, MPI_Gather и др.), рассылка данных от одного процессора всем (MPI_Bcast, MPI_Scatter и др). Коллективные операции в действующих стандартах MPI являются блокирующими. Например, пока корневой процесс не примет данные от всех остальных процессов в ходе операции MPI_Reduce, функция не вернет управление на корневом процессе и исполнение каких-либо вычислений в ходе ожидания прихода данных невозможно. Так как с ростом числа используемых процессорных элементов время на завершение глобальных операций возрастает, использование коллективных операции является одной из причин плохой масштабируемости программ. Вместе с тем, применение коллективных операций естественно для многих задач. Таким образом, возникает необходимость в реализации неблокирующих коллективных операций, когда инициализация операции производится вызовом одной функции, а требование о завершении операции выставляется посредством вызова другой функции.

Экспериментальные исследования показывают значительный выигрыш в производительности программ при переходе от блокирующих к неблокирующим функциям [6].

1.3. Неблокирующие коммуникации в NumGRID

В неоднородной вычислительной среде, которую представляет собой объединение вычислительных кластеров на основе NumGRID, задача сокрытия коммуникаций на фоне счета еще более важна, чем в однородных вычислительных системах.

Система межкластерных коммуникаций в NumGRID изначально была спроектирована для поддержки неблокирующих коммуникационных операций point-to-point. В настоящей статье представлена реализация неблокирующих коллективных коммуникационных операций MPI в NumGRID и их экспериментальное исследование.

В разделе 2 дается обзор программного комплекса NumGRID, в разделе 3 рассматривается организация межкластерных коммуникаций в NumGRID и реализация неблокирующих коллективных операций, в разделе 4 – экспериментальное исследование реализации коллективных операций.

2. Обзор программного комплекса NumGRID

2.1. Цель проекта и основные требования

Цель проекта NumGRID заключается в разработке программного комплекса для поддержки исполнения параллельных программ на объединении вычислительных кластеров с распределением процессов параллельных программ между узлами вычислительных кластеров. В отличие от таких систем, как Globus Toolkit [7], ставящих задачу объединения вычислительных ресурсов в общем, проект NumGRID сконцентрирован на разработке простого инструмента пользовательского уровня для распределения процессов параллельной программы между кластерами, к которым пользователь имеет непосредственный доступ посредством личных учетных записей. Принципиальным является обеспечение единой коммуникационной среды для распределенных процессов на основе стандартов MPI. Выбор стандарта MPI в качестве коммуникационного протокола обосновывается доминирующим положением MPI среди средств разработки параллельных программ численного моделирования.

К разработке программного комплекса в проекте NumGRID предъявляются следующие неформальные требования, обоснованные сложившейся практикой организации вычислений и целью проекта:

1. Общая коммуникационная среда для процессов распределенных по нескольким кластерам должны быть реализована на основе стандартов MPI.

2. О структуре кластеров и сети связи между ними нужно предполагать следующее: каждый кластер состоит по крайней мере из головного/управляющего узла и вычислительных узлов; при этом вычислительные узлы предназначены для запуска на них вычислительных процессов и связаны между собой высокоскоростной сетью, а с головным узлом – сетью, как правило, меньшей пропускной способности, поддерживающей протокол TCP; головной узел имеет еще по крайней мере один сетевой интерфейс, поддерживающий протокол TCP, через который узел может общаться с головными узлами других кластеров; головной узел используется для компиляции задач, управления локальными очередями задач и мониторинга задач.
3. Процессы параллельной программы должны размещаться на вычислительных узлах кластеров, узлы разных кластеров не имеют прямого сообщения друг с другом.
4. Должен быть предоставлен удобный интерфейс для задания конфигурации объединения кластеров, управления ресурсами и задачами.
5. Должны быть созданы условия для обеспечения динамических свойств прикладных программ (динамическая настраиваемость на доступные ресурсы, динамическая балансировка нагрузки, системы мониторинга и т.д.).
6. Должна обеспечиваться безопасность вычислений.
7. Каждому пользователю для запуска распределенных задач на нескольких кластерах должно быть достаточно иметь учетные записи на этих кластерах и пакет NumGRID. Организация NumGRID не должна требовать изменений в практике и политиках администрирования кластеров.
8. Кластеры могут быть разнородными в аппаратном, системном программном обеспечении, линии связи могут быть разной пропускной способности, кластеры могут иметь различную административную подчиненность.

Организации распределенного исполнения программ MPI посвящены (или были в свое время посвящены) такие проекты как PACX-MPI (Universität Stuttgart, Германия), MPICH-G2 (Northern Illinois University и Argonne National Laboratory, США), GridMPI (National Institute of Advanced Industrial Science and Technology – AIST, Япония) и другие. Однако принятые в этих проектах решения не удовлетворяют полностью приведенному списку требований. Обзор задач, которые нужно решать для объединения кластеров, и подходов к решению этих задач в различных проектах дан в [2]. Насколько известно автору, в подобных проектах не осуществлялась реализация неблокирующих функций коллективного взаимодействия.

2.2 Актуальность NumGRID

NumGRID, обеспечивая возможность распределенного выполнения приложений MPI, позволяет

- решать задачи, для которых недостаточно ресурсов отдельных кластеров,
- продлевать жизнь устаревающего оборудования за счет объединения его с новым,
- распределять части комплексных задач между специализированными кластерами в соответствии с индивидуальными требованиями частей к оборудованию и программному обеспечению,
- повысить гибкость при планировании распределения задач между кластерами в грид (распределять можно не приложения целиком, а с точностью до процессов),
- планировать постепенное наращивание мощностей распределенной системы.

Низкая пропускная способность каналов связи между узлами различных кластеров, относительно высокоскоростных соединений внутри кластеров, обуславливала сложность достижения хорошей эффективности распределенных приложений.

Однако интерес к совместному использованию вычислительных ресурсов, неоднородных как в смысле процессоров, так и в смысле связей между процессорами, приводит к 1) развитию вычислительных алгоритмов, допускающих гибкость в организации коммуникаций между процессами и даже частичную потерю сообщений [8,9], 2) развитию методов организации вычислений, позволяющих выполнять коммуникации на фоне счета и динамически перераспределять вычисления с целью оптимизации загрузки вычислительных узлов и сетей связи [10]. В то же время, 3) характеристики (пропускная способность, латентность) каналов, связывающих кластеры, становится сопоставимыми с характеристиками сетей связи между узлами кластеров. Конечно, при оценке целесообразности объединения систем, нужно принимать во внимание расстояние между ними. Например, в работе [11] Института AIST, разработавшего GridMPI, показано, что эффективность распределенного исполнения программ из тестовых пакетов NAS Parallel benchmark остается приемлемой при расстоянии до 60 км между связываемыми вычислительными системами.

Эти три технологических прорыва открывают перспективу для применения распределенных вычислительных систем к решению крупных задач, требующих коммуникаций между параллельными процессами.

2.3. Модель функционирования NumGRID

Схема устройства объединенного вычислительного ресурса NumGRID представлена на Рис. 1.

В примере объединяются два кластера через Internet. Каждый кластер имеет рабочие узлы, объединенные высокоскоростной сетью (Myrinet, Infiniband, др.). Также узлы связаны с головным узлом сетью с

меньшей пропускной способностью, поддерживающей протокол TCP. Процессы приложения MPI, находящиеся на рабочих узлах одного кластера обмениваются между собой через высокоскоростную сеть средствами специфичной для кластера библиотеки MPI, позволяющей использовать возможности высокоскоростной сети. В примере это библиотеки MPICH и LAM-MPI. Для процессов, распределенных по рабочим узлам двух кластеров, поддерживается глобальная адресация в рамках одного коммуникатора MPI. Сообщения между процессами, расположенными на разных кластерах, проходят путь через головные узлы кластеров, где для этих целей перед стартом приложения запускаются шлюзы.

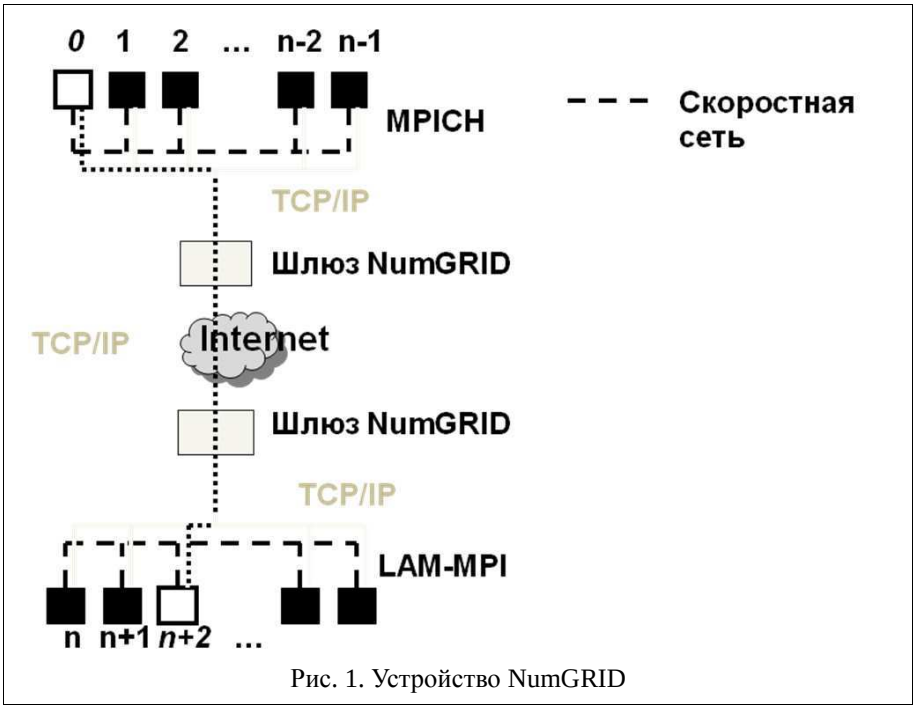

Рис. 1. Устройство NumGRID

Пользователь вначале запускает шлюзы на каждом кластере. Шлюзы устанавливают между собой связь. Пользователь загружает на головные узлы кластеров исходный код своего приложения и исходные данные, собирает приложение на каждом кластере с библиотекой NumGRID-MPI, запускает необходимое число процессов на каждом кластере как локальные задачи MPI. Локальная задача получает в параметрах информацию о месте данной группы процессов в общей распределенной задаче, что позволяет обеспечить глобальную адресацию процессов. В конце работы пользователь забирает с каждого кластера результаты работы программы.

Для обеспечения удобства работы пользователя разработано приложение для управления объединением кластеров и прохождением задач в NumGRID посредством графического интерфейса.

3.Реализация операций коллективных взаимодействий в NumGRID

Для эффективного использования высокоскоростных внутрикластерных сетей NumGRID обеспечивает передачу данных между узлами одного вычислительного кластера посредством обращения к установленным на кластерах библиотекам, реализующим MPI для соответствующих сетевых технологий. Между кластерами сообщение передается посредством шлюзов на головных узлах по алгоритмам, учитывающих топологию связей между кластерами. На сегодняшний день известны две реализации неблокирующих коммуникационных функций в соответствии с обсуждаемым стандартом MPI-3.0: LibNBC[12] и MPICH2-1.5b [13] (функции имеют префикс MPIX).

В основе реализации всех коммуникационных операций в NumGRID лежат функции асинхронной отправки и асинхронного приема данных между двумя процессами (point-to-point). Например, в реализации блокирующих функций (MPI_Send, MPI_Reduce) используется вызов асинхронной функции и функции ожидания завершения асинхронной операции.

Для организации коллективных коммуникаций в NumGRID рассматриваются две ситуации: когда все процессы, вызвавшие коллективную операцию находятся в пределах одного кластера, и когда они распределены между кластерами. В первом случае, применяются непосредственно функции установленной на кластере библиотеки MPI, реализующей соответствующую функцию.

Во втором случае группа процессов, составляющих коммуникатор коллективной операции, разбивается на подмножества в соответствии с принадлежностью процесса некоторому кластеру. Организация выполнения коллективной операции в таком случае распадается на два этапа: сбор или рассылка сведений между подмножествами и операции внутри подмножества.

Рассмотрим реализацию коллективных коммуникаций на примере операции MPI_Ireduce: неблокирующей операции редуцирования (например, собрать с каждого процесса данного коммуникатора по числу двойной точности и получить их сумму на процессе с номером 0). Для осуществления этой операции внутри каждого подмножества определяется локальный корневой процесс, который будет отвечать за локальную (в пределах кластера) сборку результата. Далее внутри каждого подмножества выполняется сборка, которая реализована в простейшем случае как набор асинхронных посылок в направлении локального корневого процесса. Для повышения эффективности операции, в случае большого количества процессов сбор результата может осуществляется по дереву. Локальные корневые процессы после асинхронной сборки локального результата, отправляют асинхронные сообщения глобальному корневому процессу операции MPI_Ireduce. Таким образом, возврат из функции MPI_Irecv происходит сразу после инициализации асинхронных коммуникационных операций. Функция MPI_Wait в NumGRID удостоверяется на отправляющих процессах в отправке сообщения их буфера-источника MPI_Ireduce, а на приемнике (корневом глобальном процессе) блокируется до фактического завершения приема всех асинхронных сообщений в рамках данного коммуникатора и данной операции.

В настоящий момент реализованы операции MPI_Ireduce, MPI_Iallreduce, MPI_Ibcast, MPI_Ibarrier для базовых типов данных MPI.

4.Эксперименты

Эксперименты по запуску приложений проводились на объединении кластеров Сибирского суперкомпьютерного центра (ССКЦ) и Новосибирского государственного университета (НГУ). Кластеры построены на основе двухпроцессорных узлов с процессорами Intel Xeon E5540 (4 ядра) и внутренней коммуникационной сетью InfiniBand. Пропускная способность канала между головными узлами кластеров – 10 Гб/с.

Исследование проведено на программе решения волнового уравнения в двумерной области явным методом. Распределение вычислений выполняется методом декомпозиции области. В конце итерации явного метода производится операция Allreduce для вычисления нормы разности между решениями соседних итераций и определения условия выхода.

В зависимости размера задачи и количества использованных процессорных элементов, наблюдалось ускорение программы до 3 раз относительно аналогичной программы, использующей блокирующие коммуникационные функции. В таблице представлено ускорение (количество раз) программы для заданного размера (верхняя строка) и количества процессорных элементов (левый столбец, процессорные элементы распределяются поровну между кластерами)

Таблица 1

	1000x1000	10000x10000
10	1,3	1
50	2,3	1,4
100	3,1	1,7

Очевидно, что в случае, когда коммуникации занимали существенную долю относительно вычислений и программа работала крайне неэффективно, сокрытие коммуникаций дало наиболее значительный прирост в производительности.

5. Заключение

Коммуникационная библиотека программного комплекса NumGRID, предназначенного для выполнения MPI-приложений на объединении кластеров, расширена набором неблокирующих функций коллективного взаимодействия (MPI_Ireduce, MPI_Iallreduce, MPI_Ibcast, MPI_Ibarrier) для базовых типов данных в соответствии с идеями и текущими спецификациями разрабатываемого стандарта MPI-3.0.

Экспериментальное исследование реализованных функций демонстрирует существенный выигрыш в производительности приложений, интенсивно использующих коллективные операции.

Дальнейшая работа в направлении темы исследования предполагает реализацию обработки пользовательских типов данных, оптимизацию алгоритмов коллективных операций для большого количества процессоров и расширение набора поддерживаемых коллективных операций MPI.

ЛИТЕРАТУРА:

1. D.Fougere, M.Gorodnichev, N.Malyshkin, V.Malyshkin, A. Merkulov, B.Roux. NumGrid Middleware: MPI Support for Computational Grids // Parallel Computing Technologies: 8th International Conference, PaCT 2005, Krasnoyarsk, Russia, September 5-9, 2005. Proceedings, Springer, 2005. - LNCS Vol. 3606, pp. 313-320.
2. М. А. Городничев. Объединение вычислительных кластеров для крупномасштабного численного моделирования в проекте NumGRID // Параллельные вычислительные технологии (ПаВТ’2012): труды международной научной конференции (Новосибирск, 26-30 марта 2012 г.). Челябинск: Издательский центр ЮУрГУ, 2012, стр. 432-443.

3. MPI Documents [электронный ресурс]. – Режим доступа: http://www.mpi-forum.org/docs. Дата обращения: 15.05.2012.
4. MPI-3.0 Standardization Effort [электронный ресурс]. http://meetings.mpi-forum.org/MPI_3.0_main_page.php. Дата обращения: 15.05.2012.
5. Rabenseifner R. Optimization of Collective Reduction Operations. In proc. of International Conference on Computational Science, June 7-9, Krakow, Poland, LNCS 3036, Springer-Verlag, 2004.
6. М. Г. Курносов, Е. С. Скворцов. Анализ и оптимизация выполнения неблокирующих операций коллективных обменов информацией на вычислительных кластерах // Новые информационные технологии в исследовании сложных структур: материалы Девятой российской конференции с международным участием. – Томск, Изд-во НТЛ, 2012, с.11.
7. Globus Toolkit Homepage. – URL: http://www.globus.org/toolkit. Дата обращения: 15.05.2012.
8. F. Oboril, M. B. Tahoori, V. Heuveline, D. Lukarski, J.-Ph. Weiss. Fault Tolerance Technique for Iterative Solvers / Karlsruhe Institute of Technology – URL: http://www.emcl.kit.edu/preprints/emcl-preprint-2011-10.pdf. Дата обращения: 15.05.2012.
9. Peng Du, Piotr Luszczek, Jack Dongarra: High Performance Dense Linear System Solver with Soft Error Resilience // In. proc. of International Conference on Cluster Computing (CLUSTER), Austin, TX, USA, September 26-30, 2011, pp. 272-280.
10. Malyshkin V.E., Perepelkin V.A. LuNA Fragmented Programming System, Main Functions and Peculiarities of Run-Time Subsystem - In: Proceedings of the 11th Conference on Parallel Computing Technologis, LNCS 6873 - pp. 53-61, Springer, 2011
11. Motohiko Matsuda, Tomohiro Kudoh, and Yutaka Ishikawa. Evaluation of MPI Implementations on Grid-connected Clusters using an EmulatedWAN Environment// Proceedings of the 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID.03), IEEE, 2003.
12. LibNBC - Nonblocking MPI Collective Operations [электронный ресурс]. http://www.unixer.de/research/nbcoll/libnbc/. Дата обращения: 15.05.2012.
13. MPICH-2 [электронный ресурс] http://www.mcs.anl.gov/research/projects/mpich2/. Дата обращения: 15.05.2012 г.