© 2012 г. М.А. ГОРОДНИЧЕВ
(Институт вычислительной математики
и математической геофизики СО РАН, Новосибирск)

РЕАЛИЗАЦИЯ MPI ДЛЯ РАСПРЕДЕЛЕННОГО ИСПОЛНЕНИЯ ПАРАЛЛЕЛЬНЫХ ПРОГРАММ НА ОБЪЕДИНЕНИИ ВЫЧИСЛИТЕЛЬНЫХ КЛАСТЕРОВ В ПРОЕКТЕ NUMGRID

В статье представлены результаты работы по реализации коммуникационной среды на основе стандартов MPI для организации выполнения параллельных программ на объединении вычислительных кластеров в проекте NumGRID. Реализация обеспечивает возможность общения в смысле MPI для процессов, расположенных на вычислительных узлах нескольких вычислительных кластеров. Обсуждаются требования к реализации, проектные решения и приводится анализ экспериментов по распределенному выполнению прикладных параллельных программ.

IMPLEMENTATION OF THE MPI FOR DISTRIBUTED COMPUTATIONS ON THE AGGREGATION OF COMPUTING CLUSTERS IN THE NUMGRID PROJECT / M.A. Gorodnichev (Institute of Computational Mathematics and Mathematical Geophysics, Pr. Ak. Lavrentjeva 6, Novosibirsk, 630090, Russia, E-mail: maxim@ssd.sscc.ru). We present the results of our work on the implementation of the MPI standards for distributed computations where processes of parallel programs are distributed over computational nodes of several computational clusters. With our implementation, such processes can communicate in the terms of MPI and legacy MPI-programs can be run distributed without major changes. We discuss requirements, taken design decisions and analyze the results of experiments on the distributed

1. Введение

Цель проекта NumGRID [1,2] 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 [3], 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. NumGRID позволяет исполнять MPI-приложение так, чтобы процессы приложения были распределены по рабочим узлам нескольких кластеров.

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.

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 [4,5],
2) развитию методов организации вычислений, позволяющих выполнять коммуникации на фоне счета и динамически перераспределять вычисления с целью оптимизации загрузки вычислительных узлов и сетей связи [6]. В то же время,
3) характеристики (пропускная способность, латентность) каналов, связывающих кластеры, становятся сопоставимыми с характеристиками сетей связи между узлами кластеров. Конечно, при оценке целесообразности объединения систем, нужно принимать во внимание расстояние между ними. Например, в работе [7] 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.

Достижения в этих трех областях открывают перспективу для применения распределенных вычислительных систем к решению крупных задач, требующих коммуникаций между параллельными процессами.

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, должна быть реализована на основе стандартов MPI.
2) О структуре кластеров и сети связи между ними нужно предполагать следующее: каждый кластер состоит 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 также 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.

2. Реализация коммуникационной среды в NumGRID

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

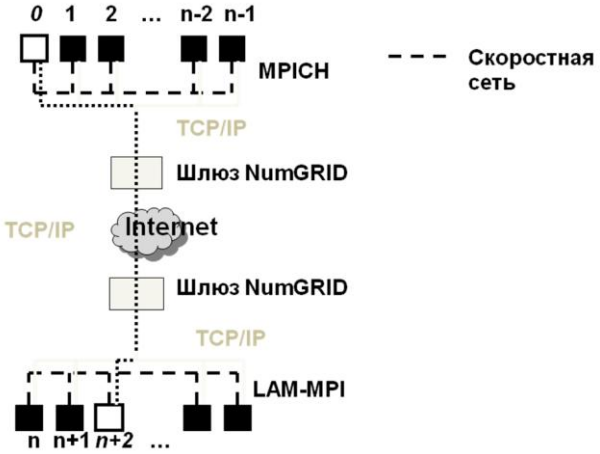

Рис. 1. Устройство NumGRID

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. Для эффективного использования высокоскоростных внутрикластерных сетей NumGRID обеспечивает передачу данных между узлами одного вычислительного кластера посредством обращения к установленной на данном кластере библиотеке, реализующей MPI для соответствующих сетевых технологий. Между кластерами сообщение передается посредством шлюзов на головных узлах по алгоритмам, учитывающих топологию связей между кластерами. Для процессов, распределенных по рабочим узлам объединяемых кластеров, поддерживается глобальная адресация в рамках одного коммуникатора MPI.

Реализация межкластерных коммуникаций в NumGRID опирается на следующие принципы:

– Применение пакетирования сообщений (как противоположность последовательному получению сообщения целиком и последующей передаче на каждом промежуточном узле) позволяет 1) обойти проблему ограниченного объема памяти на

промежуточных узлах, 2) осуществлять диспетчерирование пакетов нескольких сообщений параллельно и в чередовании (в результате появляется возможность доставлять короткие сообщения «на фоне» отсылки больших сообщений), 3) уменьшить задержку в доставке сообщения 4) увеличить скорость передачи сообщения за счет использования нескольких путей следования пакетов, 5) оптимизировать распределение памяти на промежуточных узлах.

- Пользователь (или автоматизирующая система запуска задач в NumGRID) должны иметь возможность описывать физическую топологию связей между кластерами с указанием характеристик связи. Это позволит использовать интеллектуальные алгоритмы маршрутизации.
- Должна быть обеспечена возможность задавать сообщениям приоритеты. Эта функциональность может быть использована высокоуровневыми системами программирования для ускорения доставки управляющих сообщений исполнительных подсистем. Нужно иметь в виду, что приоритеты сообщений не предусмотрены действующими стандартами MPI.
- NumGRID должен поддерживать динамическое подключение и отключение ресурсов в ходе работы программы, и динамическое управление MPI-процессами.
- Критическую важность имеет эффективная реализация неблокирующих коммуникационных функций.

В основе реализации всех коммуникационных операций в NumGRID лежат функции асинхронной отправки и асинхронного приема данных между двумя процессами (point-to-point).

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 с учетом структуры межкластерной сети и операции внутри подмножества.

В NumGRID реализованы экспериментальные функции неблокирующих коммуникационных операций, введение которых ожидается в стандарте MPI-3.0. Известно [8], что значительную часть времени вычислительные параллельные программы проводят в ожидании завершения блокирующих коллективных операций. Введение неблокирующих коммуникаций позволяет минимизировать эти непроизводительные расходы.

3. 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.

Накладные расходы. Тестовое приложение состоит из двух процессов. Процесс номер 0 посылает сообщение заданного размера процессу номер 1 и получает сообщение такого же размера назад. Тест показывает, как зависит время передачи сообщения от его

размера. На рис. 2 дано сравнение времени прохождения сообщений различных размеров (горизонтальная ось) при обмене между двумя узлами внутри кластера ССКЦ (линия MPI), и между кластерами при выборе различных размеров пакетов, на которые дробится сообщение (256, 50000). Размеры сообщений и пакетов заданы в элементах типа MPI_DOUBLE. Видно, что уменьшение размера пакета существенно увеличивает суммарные расходы на обработку пакетов.

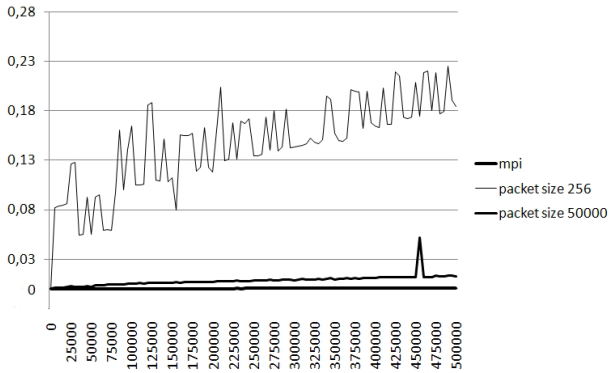

Рис. 2. Время прохождения кругового сообщения между двумя процессами в зависимости от размера сообщения, сек. Размер сообщений в элементах типа MPI_DOUBLE.

Передача короткого сообщения на фоне длинного. Тест демонстрирует (Рис. 3) как время передачи короткого сообщения (туда и обратно) зависит от размера пакета. Видно, что использование пакетирования позволяет доставлять короткое сообщение пока выполняется доставка длинного сообщения. Тест проводится таким образом, что короткое сообщение начинает передаваться после того, как начата передача длинного сообщения, и его доставка заканчивается раньше окончания доставки длинного сообщения.

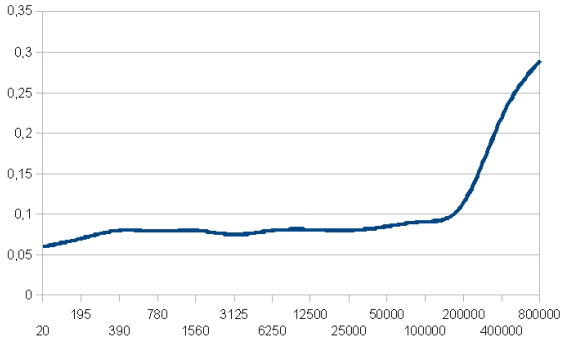

Рис. 3. Время кругового прохождения короткого сообщения на фоне доставки длинного сообщения, сек., в зависимости от размера пакета сообщения в элементах типа MPI_DOUBLE.

Решение волнового уравнения. Ниже представлены графики, демонстрирующие характеристики исполнения программы решения волнового уравнения с помощью двухслойной явной схемы. На всех диаграммах запись «P (NxS)» означает «всего P ядер, из них N на кластере НГУ и S – на кластере ССКЦ.

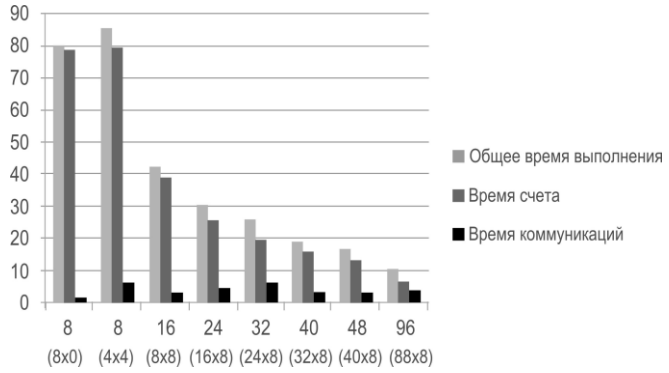

Рис. 4. Время решения волнового уравнения явным методом на NumGRID, сек.

На Рис.4 видно, что при переходе к распределенным вычислениям (с 8x0 на 4x4) увеличивается общее время работы программы за счет увеличения расходов на коммуникации. Расходы на коммуникации в данном примере увеличиваются в 5 раз, что приводит к росту времени работы программы на ~6%.

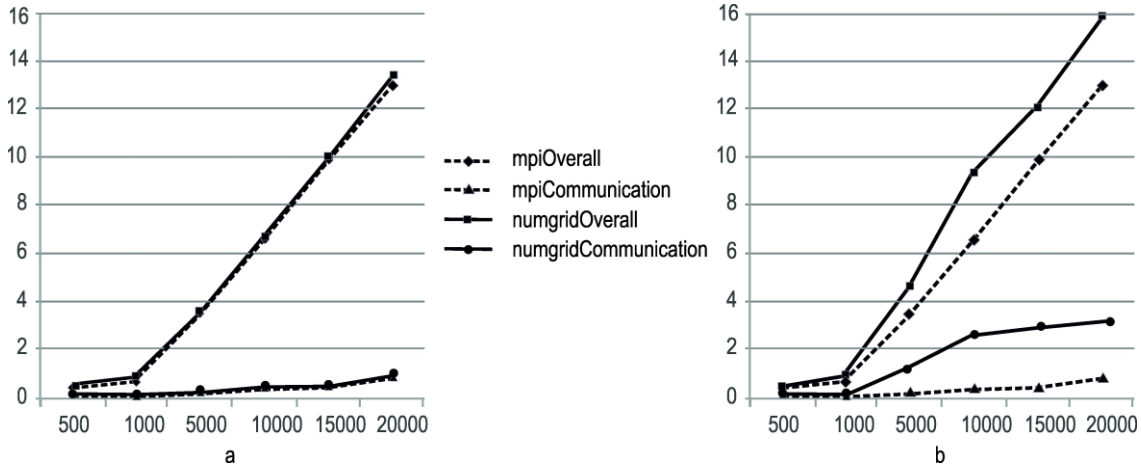

Рис. 5. Зависимость времени исполнения программы от увеличения объема коммуникаций на примере решения волнового уравнения для разных размеров задачи и способов распределения вычислений, время в сек.

На Рис. 5 показано, как изменяется время решения двумерного волнового уравнения на двух процессах в зависимости от размера задачи и способа распределения вычислений. Обозначения: mpiOverall – время работы программы, процессы которой расположены в рамках одного кластера; mpiCommunication – время, потраченное программой на коммуникации; соответствующие графики numgrid показывают время для процессов, расположенных на разных кластерах.

Распределение вычислений выполняется методом декомпозиции области. Таким образом, каждый процесс обрабатывает половину области. На части ***a*** диаграммы показана ситуация, в которой область моделирования увеличивается (горизонтальная ось) по разрезанной размерности. При этом размер границы разреза, очевидно, не изменяется и объем коммуникаций между процессами остается постоянным. На части ***b*** диаграммы показано, что происходит, когда изменяется размер области по другой размерности. В этом случае, с изменением размера области соответственно увеличивается длина разреза

и объем коммуникаций. По сравнению с ситуацией ***a***, видно, что в ситуации ***b*** время коммуникаций между процессами, расположенными на разных кластерах, существенно увеличивается в отношении времени коммуникаций между процессами внутри кластера.

Генерация случайных чисел и расчет статистик. Ниже представлены результаты тестирования программы, которая параллельно генерирует случайные величины и вычисляет статистики. Программа используется для моделирования физических процессов методами Монте-Карло. Программа осуществляет редкие коллективные коммуникации для сбора статистик, в остальное время процессоры выполняют существенные по объему вычисления независимо. Это позволяет ожидать, что относительно плохая пропускная способность сети между кластерами незначительно скажется на общей производительности программы.

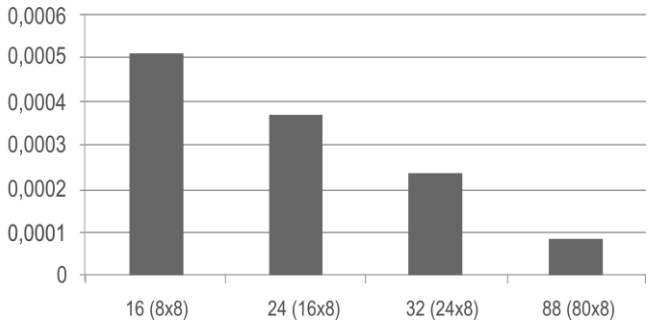

Рис. 6. Время генерации 2000 значений случайной величины и расчета статистик на NumGRID, сек. При этом время генерации 2000 значений случайной величины и расчета статистик на одном ядре кластера 1: 0,0109 сек.

Время исполнения программы при переходе от решения на одном ядре к решению на 16 ядрах, по 8 ядер на каждом кластере, уменьшается в 21 раз. Объяснить подобное сверхлинейное ускорение можно более эффективным использованием кэшей процессоров при увеличении количества процессоров и низкими коммуникционными расходами, свойственными данной задаче. При дальнейшем увеличении количества вовлеченных ресурсов так же отмечается сверхлинейное ускорение. Рис. 7 дает более ясное представление об ускорении времени решения при увеличении количества процессоров.

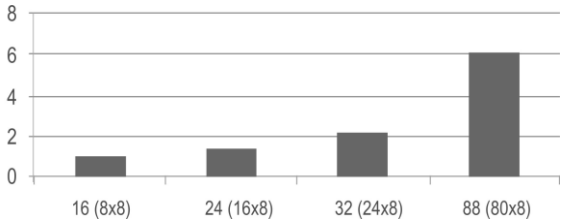

Рис. 7. Ускорение генерации 2000 значений случайной величины и расчета статистик на NumGRID относительно времени работы программы на двух узлах: один узел кластера НГУ (8 ядер) и один узел кластера ССКЦ (8 ядер).

4. Заключение

В проекте NumGRID разработаны средства для установления общей коммуникационной среды между процессами параллельной программы, распределенными по внутренним узлам нескольких вычислительных кластеров.

В ходе экспериментов показано, что в зависимости от способа распределения вычислений между кластерами и выбора размера пакетов меняется эффективность работы приложений. Показано, что использование дополнительных процессоров из другого кластера позволяет уменьшать время выполнения приложения при определенных конфигурациях запуска.

Несмотря на существенную разницу в производительности систем коммуникации внутри кластеров и между кластерами, время решения задач увеличивается умеренно для решения волнового уравнения на том же количестве процессоров и уменьшается для решения задачи поиска статистик.

Таким образом, можно заключить, что существует класс задач, для которого применение NumGRID будет целесообразно.

СПИСОК ЛИТЕРАТУРЫ

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. *Globus Toolkit Homepage.* – URL: http://www.globus.org/toolkit. Дата обращения: 15.05.2012.
4. *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.
5. *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.
6. *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.
7. *Motohiko Matsuda, Tomohiro Kudoh, and Yutaka Ishikawa.* Evaluation of MPI Implementations on Grid-connected Clusters using an Emulated WAN Environment // Proceedings of the 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID.03), IEEE, 2003.
8. *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.