УДК 519.6

ПРОГРАММНЫЙ КОМПЛЕКС S-MPI ДЛЯ ОБЕСПЕЧЕНИЯ РАЗРАБОТКИ, ОПТИМИЗАЦИИ И ВЫПОЛНЕНИЯ ВЫСОКОПАРАЛЛЕЛЬНЫХ ПРИЛОЖЕНИЙ НА СУПЕРКОМПЬЮТЕРНЫХ КЛАСТЕРНЫХ СИСТЕМАХ

Г. И. Воронов, В. Д. Трущин, В. В. Шумилин, Д. В. Ежов
(ООО "Центр компетенций и обучения", г. Саров)

Описывается разрабатываемый в рамках контракта с Министерством образования и науки РФ отечественный программный комплекс S-MPI, включающий в себя библиотеку S-MPI и программные инструменты для анализа корректности и эффективности параллельных приложений. Приводится сравнение производительности S-MPI и двух других широко распространенных реализаций MPI-2 (Open MPI 1.5.4 и MVAPICH 1.2.7).

Ключевые слова: MPI, параллельные вычисления, проверка корректности, инструментальные средства анализа.

Введение

Для эффективного использования современных кластерных вычислительных систем необходимы эффективные, надежные и удобные в использовании программные комплексы, поддерживающие стандарт MPI-2 [1, 2] и обеспечивающие ускоренную разработку, оптимизацию и выполнение *высокопараллельных* приложений. При реализации таких программных комплексов приходится решать множество теоретических и практических проблем, наиболее важными из которых являются отказоустойчивость, масштабируемость, адаптация к современным коммуникационным средам и топологии кластеров, поддержка смешанных моделей программирования и т. п.

Указанные проблемы в том или ином виде решаются в различных коммерческих и свободно распространяемых реализациях стандарта MPI-2, наиболее известны из которых Intel MPI, IBM Platform MPI, MPICH2, MVAPICH2 [3] и Open MPI [4—6]. Однако разработка всех этих программных продуктов находится под контролем иностранных государств или ассоциаций промышленных групп. Кроме того, в настоящее время нет ни одной реализации, которая бы удовлетворяла всем предъявляемым требованиям и, в частности, обеспечивала поддержку разрабатываемого в России аппаратного и программного обеспечения кластерных систем. Поэтому задача разработки отечественного программного комплекса, поддерживающего стандарт MPI-2, является очень актуальной.

В данной статье представлены состав и функциональное назначение разрабатываемого программного комплекса S-MPI, а также первые результаты его применения в отечественных промышленных приложениях.

Состав и назначение программного комплекса S-MPI

Разрабатываемый программный комплекс (ПК) S-MPI предназначен:

- для создания параллельных прикладных программ и обеспечения их выполнения на широком спектре высокопроизводительных вычислительных систем, включая *облачные* и *грид*-сети;
- обеспечения полнофункциональной среды ускоренной разработки, отладки и оптимизации параллельных приложений.

Указанную функциональность ПК обеспечивают следующие компоненты:

1) библиотека MPI, получившая название S-MPI, которая позволяет создавать и выполнять параллельные программы, использующие для распараллеливания функции стандарта MPI-2;
2) программный компонент настройки параметров библиотеки MPI, предназначенный для подбора оптимальных параметров библиотеки как для конкретного кластера, так и конкретного приложения;
3) компонент проверки корректности, предназначенный для обнаружения реальных и потенциальных проблем, связанных с некорректным использованием функций MPI;
4) компонент профилирования и трассировки приложения для сбора и сохранения информации о вызовах MPI-функций во всех процессах приложения;
5) компонент обработки данных с графическим интерфейсом для обеспечения визуализации профилировочной и трассировочной информации о выполнении параллельного приложения;
6) программа установки ПК MPI-2 на вычислительную систему.

Ядром ПК S-MPI является библиотека S-MPI. Все остальные компоненты ПК используются для отладки, анализа производительности и оптимизации параллельных приложений, собранных с библиотекой S-MPI.

Библиотека S-MPI

Одной из главных особенностей библиотеки является универсальность — обеспечение независимости параллельных программ от архитектур многопроцессорных систем, на которых они выполняются. То есть однажды откомпилированное и собранное с библиотекой S-MPI приложение может выполняться на кластерных системах с различной архитектурой, в том числе гибридной (CPU, GPU, Intel Phi), и произвольной коммуникационной средой из числа поддерживаемых (таких как Ethernet и InfiniBand).

В качестве кодовой базы для разработки отечественной библиотеки S-MPI была выбрана наиболее распространенная и подходящая для этих целей реализация стандарта MPI-2 с открытым кодом Open MPI.

К основным достоинствам библиотеки Open MPI можно отнести:

- наиболее приспособленную для расширений функциональности модульную архитектуру MCA (Modular Component Architecture);
- поддержку неоднородных коммуникационных сред;
- средства обеспечения надежности;
- широкий набор поддерживаемых возможностей;
- хорошую производительность и масштабируемость выполнения приложений.

Но, как и все коды открытого доступа, библиотека Open MPI имеет ряд недостатков, затрудняющих ее промышленное использование:

- отсутствие универсальности (т. е. зависимость выполнения приложения от программной и аппаратной среды кластера, на котором библиотеки и приложение были собраны);
- недостаточную стабильность;
- негарантированную бинарную совместимость между версиями;
- отсутствие средств для проверки корректности использования MPI в приложениях.

По сравнению с Open MPI библиотека S-MPI существенно усовершенствована в части универсальности использования, стабильности и надежности, а также дополнена новыми возможностями. Она, в частности, обеспечивает:

- универсальность выполнения прикладной программы на кластерах с произвольной коммуникационной средой;
- автоматический выбор наиболее эффективных коммуникационных сред из числа доступных;
- автоматическое определение и использование топологии кластерной системы для выбора оптимальных алгоритмов функционирования компонентов библиотеки;
- специальные топологические алгоритмы коллективных операций;
- новый механизм передачи сообщений через общую память;
- поддержку многопоточности MPI-процессов уровня MPI_THREAD_MULTIPLE вне зависимости от природы потоков (Posix, OpenMP, TBB);

– гибкое управление размещением процессов и их потоков на вычислительных ядрах;
– сбор внутренней статистики о вызовах MPI-функций приложением (интенсивность вызовов, использованное время, аргументы вызова и т. д.).

За счет перечисленных и других усовершенствований производительность библиотеки S-MPI существенно улучшена по сравнению с Open MPI. На рис. 1, 2 приведено интегральное сравнение производительности по всем тестам IMB (Intel® MPI Benchmarks) двух версий библиотеки Open MPI (1.5.4 и 1.6.3) с библиотекой S-MPI версии 0.1 для разных конфигураций запуска. Правильность применения методов усовершенствования S-MPI подтверждается интегральным ускорением до двух и более раз для разных длин сообщений.

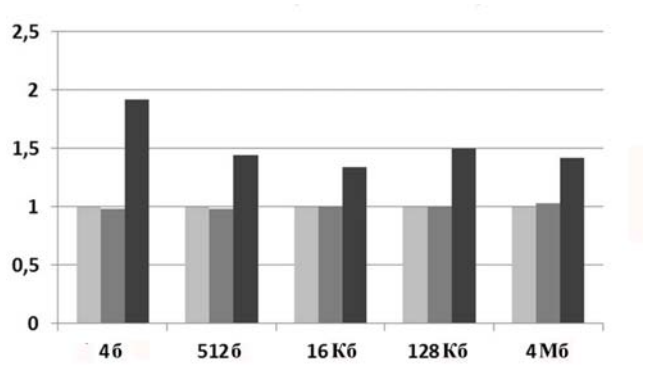

Рис. 1. Интегральное сравнение производительности библиотек MPI на тестах IMB для общей памяти (16 процессов на одном узле для процессора AMD Interlagos): светло-серые столбцы — Open MPI 1.5.4; серые — Open MPI 1.6.3; черные —S-MPI

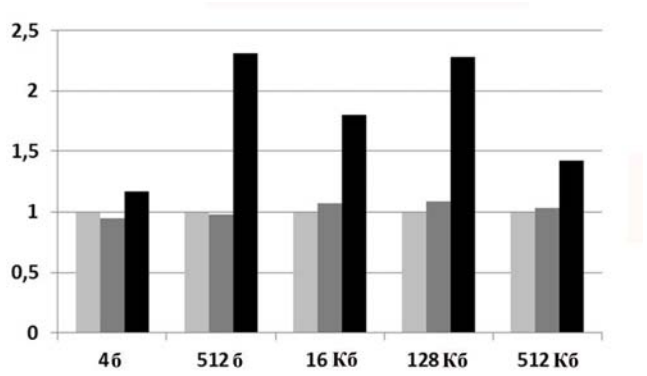

Рис. 2. Интегральное сравнение производительности библиотек MPI с коммуникационной средой InfiniBand QDR (384 процессов на 32 узлах архитектуры Intel Westmere): светло-серые столбцы — Open MPI 1.5.4; серые — Open MPI 1.6.3; черные — S-MPI

Библиотека S-MPI была успешно испытана на тестовых моделях промышленных приложений ЛОГОС-прочность [7, 8] и ЛОГОС-CFD [8, 9], разрабатываемых в РФЯЦ-ВНИИЭФ. Следует отметить, что для этих запусков были использованы оптимальные параметры библиотеки S-MPI, подобранные вручную, исходя из особенностей указанных приложений. Как видно из рис. 3, 4, выполнение приложений ускорилось до 18 % для ЛОГОС-CFD и до 30—40 % для ЛОГОС-прочность. Нет сомнения в том, что использованные оптимальные параметры для S-MPI могут быть определены компонентом автоматической настройки параметров MPI.

Помимо производительности, основными характеристиками библиотеки MPI являются масштабируемость и отказоустойчивость.

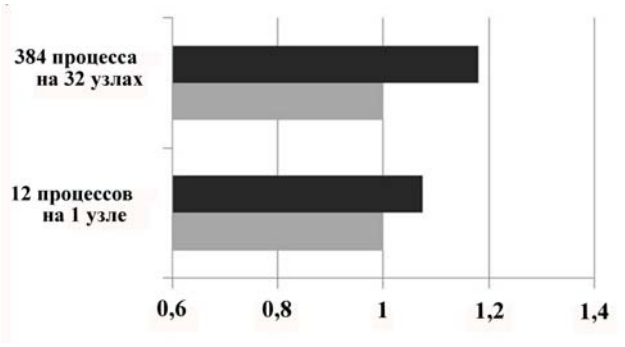

Рис. 3. Сравнение производительности тестовой модели ЛОГОС-CFD (ЛОГОС-гидродинамика) с разными библиотеками MPI для разных конфигураций запуска: серый цвет — MVAPICH 1.2.7; черный — S-MPI 0.1

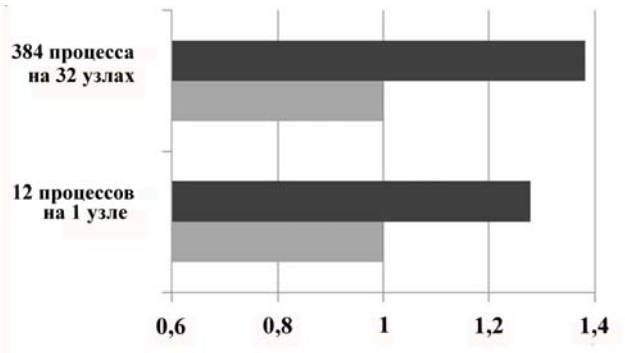

Рис. 4. Сравнение производительности тестовой модели ЛОГОС-прочность (ЛЭГАК-ДК) с разными библиотеками MPI для разных конфигураций запуска: серый цвет — MVAPICH 1.2.7; черный — S-MPI 0.1

Решение проблемы масштабируемости изначально заложено в архитектуру Open MPI — библиотека поддерживает до сотен тысяч одновременно работающих процессов. Библиотека S-MPI еще более усовершенствована в части масштабируемого запуска и уменьшения накладных расходов библиотеки в сверхбольших запусках.

Для обеспечения отказоустойчивости в библиотеке реализованы механизмы автоматического переключения на альтернативную коммуникационную среду в случае отказа основной, средства автоматического продолжения выполнения приложения в случае отказа одного из каналов передачи данных в многоканальной конфигурации коммуникационной среды, а также поддержка контрольных точек (checkpoint/restart).

Инструменты анализа параллельных приложений

Создание эффективной параллельной программы требует значительно больше ресурсов (людских и временных), чем создание последовательной программы. Поэтому актуальны и востребованы инструментальные средства, помогающие отладить и оптимизировать параллельную программу.

Создание параллельной программы упрощается при наличии средств проверки правильности (корректности) использования функций MPI-библиотеки. В ПК S-MPI разработан инструмент MCCT (MPI Correctness Checking Tool). Он содержит около ста проверок, выполняющих анализ параметров используемых MPI-функций, проверку целостности передаваемых данных между MPI-процессами, отслеживание возможных потерь двухточечных сообщений, проверку условий возникновения потенциальных и реальных взаимоблокировок (deadlocks) между MPI-процессами с учетом возможных условий гонки (race conditions) и множество других. Предусмотрена возможность отслеживания стека вызовов, которая позволяет более точно указать проблемное место в приложении.

Инструмент MCCT является динамически подгружаемым на этапе запуска MPI-задачи, т. е. все проверки осуществляются в реальном времени на стадии исполнения (runtime). При этом он не оказывает чрезмерного влияния на производительность проверяемого MPI-приложения по сравнению с программами-конкурентами, такими как Intel Message Checker, Marmot, MUST.

В процессе написания параллельных приложений на первый план, помимо отладки, выходят вопросы их эффективности (с точки зрения параллелизма). Решить эти вопросы с использованием стандартных средств анализа обычных приложений невозможно. Поэтому необходимы инструменты для анализа эффективности параллельных приложений и сбалансированности их процессов. Эти инструменты должны базироваться на сборе профилирующей (статистической) и трассировочной (реализующей зависимость от времени) информации и средствах ее последующего анализа.

Существует несколько пакетов, которые предназначены для анализа параллельных приложений; наиболее известны коммерческие продукты ITAC (Intel Trace Analyzer and Collector), Vampire, предоставляющие широкие возможности для анализа. Но они также имеют ряд недостатков, особенно в части масштабируемости при работе с сотнями и тысячами процессов. С увеличением размеров запусков параллельных приложений объемы собираемых и затем анализируемых трасс растут и, начиная с некоторого значения, становятся критически большими. Это могут быть трассы объемом в десятки и сотни гигабайтов, а в близком будущем уже и терабайтов, и даже десятки терабайтов. Трассы такого объема можно собирать, но очень трудно обрабатывать с использованием традиционных методов доступа к данным, так как время ожидания результата становится неприемлемо большим.

Вновь разработанный формат трассы XTF (eXtended Trace Format) решает эту проблему путем изменения способа организации данных в файле с возможностью получения времени доступа порядка $O(\log\langle\text{количество записей}\rangle)$, т. е. слабо зависящего от роста объема трассы. Основная идея состоит в использовании страничноструктурированного индексно-последовательного контейнера данных с возможностью прямого доступа к индивидуальной записи.

Использование XTF позволит ускорить процесс сбора трассы в несколько раз по сравнению с программами-конкурентами (ITAC, Vampire), но основной выигрыш ожидается при обработке трасс — процесс загрузки трассы ускорится в десятки и более раз из-за замены используемого конкурентами традиционного последовательного формата трасс. При этом сохраняется вся имевшаяся полнота функциональности.

Программный компонент обработки данных с графическим интерфейсом предназначен для визуализации, обработки и анализа трасс параллельных приложений, собранных с помощью компонента профилирования и трассировки приложения. Компонент обеспечивает возможность поиска проблемных с точки зрения производительности мест в параллельных приложениях с целью повышения их эффективности и улучшения масштабируемости.

Графический визуализатор трасс использует кросс-платформенную библиотеку Qt [10], поэтому может функционировать под разными операционными системами (Linux, Windows). Компонент визуализации обеспечивает возможность автоматизированного графического анализа статистической и профилирующей информации с использованием:

- событийных шкал зависимости от времени для функций, коллективных и двухточечных операций;
- качественных и количественных отображений интенсивности обмена сообщениями между процессами;
- отображения информации от модуля проверки корректности;
- статистики по функциям, сообщениям, коллективным операциям;
- диаграмм сбалансированности процессов.

Предлагаемый инструмент сбора трасс выгодно отличается от конкурентных тем, что является отказоустойчивым к сбоям выполнения параллельной программы — информация о выполнении программы до момента сбоя будет сохранена и может быть проанализирована.

Заключение

ПК S-MPI призван стать базой для обеспечения высокопараллельных расчетов в разных отраслях экономики. Библиотека S-MPI, удовлетворяющая современным требованиям и превосходящая зарубежные аналоги по основным характеристикам, позволит повысить эффективность использования отечественных компьютерных мощностей. Объединение в одном комплексе программных средств для разработки, исполнения, анализа и оптимизации параллельных приложений (сбор трасс, анализ эффективности, нахождение узких мест в производительности, проверка корректности кода и т. п.) позволит сократить сроки разработки и отладки параллельных приложений. И наконец, ПК S-MPI даст возможность адаптироваться к возможным отечественным компонентам аппаратного обеспечения и специфическим топологиям/конфигурациям отечественных кластеров (в том числе разрабатываемым в РФЯЦ-ВНИИЭФ).

В дальнейшем ПК S-MPI может стать базой для разработки программного обеспечения, позволяющего получить эксафлопсную производительность.

Работа по созданию ПК S-MPI ведется в рамках контракта (№ 07.524.12.4020) с Министерством образования и науки РФ.

Список литературы

1. *Message Passing Interface Forum.* MPI: A Message Passing Interface // Proc. "Supercomputing'93". Los Alamitos: IEEE Computer Society Press, 1993. P. 878—883.
2. *Message Passing Interface Forum.* MPI-2: Extensions to the Message-Pasing Iterface. http://www.mpi-forum.org/docs/mpi2-report.pdf.
3. MVAPICH: MPI over InfiniBand, 10 GigE/ iWARP and RDMAoE. http://mvapich.CSE.ohio-stste.edu/.
4. *Gabriel E., Fagg G. E., Bosilca G. et al.* Open MPI: Goals, concept, and design of a next generation MPI implementation // Proc. 11th European PVM/MPI Users' Group Meeting. Budapest, Hungary. September 2004. P. 97—104.
5. *Graham R. L., Woodall T. S., Squyres J. M.* Open MPI: A flexible high performance MPI // 6th Int. Conf. on Parallel Processing and Applied Mathematics in Poznan. Poland, September 2005.
6. *Squyres J. M., Lumsdaine A.* The component architecture of Open MPI: Enabling third-party collective algorithms // Proc. 18th ACM Int. Conf. on Supercomputing, Workshop on Component Models and Systems for Grid Applications / Ed. by V. Getov, T. Kielmann. France, St. Malo. July 2004. P. 167—185.
7. *Циберев К. В., Артамонов М. В., Авдеев П А. и др.* Параллельный пакет программ ЛЭГАК-ДК для расчета задач гидрогазодинамики и прочности на неструкту-

рированных сетках в лагранжево-эйлеровых переменных // Сб. тез. докл. XI межд. семинара "Супервычисления и математическое моделирование". Саров: РФЯЦ-ВНИИЭФ, 2009. С. 111.

8. *Дерюгин Ю. Н., Козелков А. С., Спиридонов В. Ф. и др.* Многофункциональный высокопараллельный пакет программ ЛОГОС для решения задач тепломассопереноса и прочности // Сб. тез. докл. Санкт-Петербургского науч. форума "Наука и общество". С.-Пб.: Изд-во Политехнического ун-та, 2012.
9. *Козелков А. С., Дерюгин Ю. Н., Зеленский Д. К. и др.* Многофункциональный пакет программ ЛОГОС для расчета задач гидродинамики и тепломассопереноса на многопроцессорных ЭВМ: базовые технологии и алгоритмы // Тр. XII Межд. семинара "Супервычисления и математическое моделирование". Саров, 11—15 октября 2010 г. Саров: РФЯЦ-ВНИИЭФ, 2011. С. 215—230.
10. Библиотека Qt. http://qt.digia.com (коммерческая версия); http://qt-project.org (свободно распространяемая версия).

Статья поступила в редакцию 20.02.13.