Оптимизация теста HPCG для суперкомпьютеров с сетью «Ангара»

А.А. Агарков, А.С. Семенов, А.С. Симонов

ОАО «НИЦЭВТ»

Данная работа посвящена оптимизированной реализации теста HPCG для суперкомпьютеров с сетью «Ангара», разработанной в ОАО «НИЦЭВТ». Тест HPCG заключается в решении системы линейных уравнений с разреженной матрицей методом сопряженных градиентов с предобуславливателем. Предлагаемая реализация включает оптимизации уровня вычислительного узла, в том числе изменение формата хранения разреженной матрицы, OpenMP-распараллеливание и векторизацию, а также оптимизацию межпроцессных обменов для сети «Ангара» при помощи использования библиотеки SHMEM вместо библиотеки MPI. В работе представлены первые результаты на макетном 36-узловом кластере с сетью «Ангара», которые превосходят результаты базовой версии теста HPCG 2.4. *Ключевые слова: коммуникационная сеть «Ангара», библиотека SHMEM, HPCG*

Введение

HPCG [1] — новый тест, предназначенный для дополнения теста LINPACK (HPL) [2], который в настоящее время используется для ранжирования суперкомпьютеров в списке TOP500. В основе HPCG лежит решение линейных уравнений с разреженной матрицей большой размерности при помощи итерационного метода сопряженных градиентов с многосеточным предобуславливателем. В отличие от HPL тест HPCG обеспечивает стрессовую нагрузку подсистемы памяти вычислительных узлов и коммуникационной сети, представляя значительный класс современных суперкомпьютерных приложений.

Тест HPCG стал предметом пристального внимания исследователей всего мира, появились работы по его оптимизации для суперкомпьютеров Tianhe-2 [3, 4], K-computer [5], исследуются возможности ускорителей NVIDIA GPU [6] и Intel Xeon Phi [3], вышла работа по предсказанию производительности HPCG [7]. Российские исследователи также уделяют внимание этому тесту [8, 9].

В ОАО «НИЦЭВТ» разрабатывается высокоскоростная коммуникационная сеть «Ангара» с топологией «многомерный тор» [10–14]. В 2013 году выпущен кристалл маршрутизатора сети [15], в 2014 году представлены предварительные результаты оценочного тестирования 12-узлового кластера с этой сетью [16]. В настоящий момент ведется оценочное тестирование 36-узлового кластера с сетью «Ангара».

Перечисленные работы по оптимизации HPCG не предоставляют исходных кодов, в связи с этим и для более глубокого понимания этого теста было принято решение разработать собственную реализацию, призванную, в частности, демонстрировать возможности сети «Ангара». Данная работа представляет первые результаты оптимизации HPCG.

1. Описание теста HPCG

В тесте HPCG решается система линейных уравнений $Ax = b$ с разреженной матрицей A и данным начальным приближением x_0. Матрица системы представляет собой дискре-

тизацию уравнения в частных производных эллиптического типа с 27-точечным шаблоном на регулярной трехмерной сетке. Строки в разреженной матрице представляют точки в сетке. Матрица имеет достаточно простую структуру, что удобно для инициализации задачи и проверки правильности оптимизированных решений, однако оптимизированные пользовательские реализации не должны опираться на структуру матрицы для ускорения вычислений, поэтому следует полагать, что имеется разреженная матрица общего вида. В частности, нельзя отказываться от косвенной адресации при доступе к структурам данных, в которых хранится матрица.

В HPCG каждый MPI-процесс отвечает за подмножество строк матрицы A и элементов вектора b, соответствующее размеру подобласти $N_x * N_y * N_z$, который задается пользователем в конфигурационном файле. Процессы выстраиваются в трехмерную решетку $P = P_x * P_y * P_z$, где P — общее число процессов. Итоговый размер задачи равен $G_x * G_y * G_z$, где $G_x = P_x * N_x, Gy = P_y * N_y, G_z = P_z * N_z$.

В тесте HPCG решается система уравнений при помощи выполнения в базовой реализации (версия 2.4) 50 итераций метода сопряженных градиентов CG. В начале каждой итерации метода CG выполняется предобуславливание системы с помощью итеративного многосеточного метода (выполняется 3 итерации) с использованием симметричного сглаживателя Гаусса-Зейделя.

В соответствии со спецификацией пользователь может оптимизировать следующие функции:

1. Реализацию метода симметричного Гаусса-Зейделя (ComputeSYMGS) и код многосеточного предобуслаливателя.
2. Вычисление произведения разреженной матрицы на вектор (ComputeSPMV).
3. Вычисление скалярного произведения векторов (ComputeDOT).
4. Вычисление суммы векторов вида: $a * X + b * Y$ (ComputeWAXPBY).

Пользовательское решение может также изменять формат хранения матрицы и производить перестановки строк. Однако по требованию спецификации оптимизированная пользовательская реализация должна достичь той точности решения системы, что и базовая версия HPCG. Оптимизации могут приводить к уменьшению скорости сходимости задачи, в этом случае при тестировании реализации выполняется необходимое число итераций, которое больше базовых 50 итераций. Время, требующееся на дополнительные итерации, а также время преобразования структур данных, необходимых для оптимизации, входит во время решения задачи и учитывается при подсчете производительности.

2. Оптимизации

Выполненные в работе оптимизации теста HPCG разделяются на две категории: одноузловые оптимизации и оптимизации межузловых коммуникаций. Одноузловые оптимизации включают организацию более эффективной работы с оперативной памятью внутри вычислительного узла, векторизацию и распараллеливание функций с помощью технологии OpenMP. Такие оптимизации общеизвестны, они рассматривались в работах [3–5]. Напрямую эти оптимизации не касаются сети «Ангара», но они необходимы для получения высокой производительности, их описание находится в разделах 2.1–2.3. В разделе 2.4 рассмотрены специфические для сети «Ангара» межузловые коммуникации.

Результаты производительности для вариантов одноузловой оптимизации получены на узле с двумя шестиядерными процессорами с включенным режимом Hyper-Threading. Параметры узла (тип A) представлены во втором столбце таблицы 1.

Таблица 1. Параметры вычислительных узлов

Параметр	**Узел типа A**	**Узел типа B**
Процессор	2xIntel Xeon E5-2630	Intel Xeon E5-2660
Тактовая частота процессора, ГГц	2.3	2.2
Количество ядер в узле	12	8
Размер кэша L3, МБ	15	20
Пиковая производительность узла, Гфлопс	220.8	140.8

Таблица 2. Параметры вычислительного кластера с сетью «Ангара»

Параметр	**Значение**
Количество узлов типа A	24
Количество узлов типа B	12
Общая пиковая производительность, Гфлопс	6988.8
Интерфейс процессора с сетью «Ангара»	PCIe gen2 x16
Тактовая частота сетевого адаптера, МГц	500
Задержка передачи 16-байтового сообщения между двумя узлами, SHMEM/MPI, мкс	0.7/1

2.1. Выделение непрерывного блока памяти

В исходном варианте HPCG память для массивов элементов и индексов разреженной матрицы выделяется блоками, каждый блок имеет размер, равный максимальному количеству ненулевых элементов в строке. Блочное выделение памяти препятствует эффективному использованию всей подсистемы памяти, в частности кэшей данных, кэша дескрипторов страниц (TLB). Первое, что было предпринято — это непрерывное выделение памяти для этих массивов. Результаты производительности для локальной области 16x16x128 приведены на рис. 2. Выделение непрерывного блока памяти позволило получить прирост производительности в среднем в полтора раза по сравнению с базовой реализацией.

2.2. Раскраска матрицы

Функция ComputeSYMGS, реализующая симметричный метод Гаусса-Зейделя (SYMGS), занимает наибольшее время среди всех остальных функций, поэтому является одной из основных, но и наиболее сложных задач для оптимизации.

В тесте HPCG для построения исходной матрицы используется 27-точечный шаблон в трехмерном пространстве (см. рис. 1), каждая точка заданной области соответствует какой-то строке матрицы и элементу вектора. Применение SYMGS на каждом уровне сетки включает в себя обмен граничными значениями между соседними вычислительными узлами, за которым следует прямой и обратный ход SYMGS. На выходе функция SYMGS возвращает обновленный вектор значений x. Основная сложность заключается в том, что для получения значения x_i используются новые значения $x_1 ... x_{i-1}$ в прямом и $x_{i+1} ... x_n$ в обратном ходе, что делает невозможным параллельное выполнение всех таких операций.

Для распараллеливания SYMGS, в частности при оптимизации теста HPCG [3–5], применяется раскрашивание точек области. Первый вариант оптимизации состоял в раскраске области в 8 цветов (см. рис. 1), при такой раскраске элементы вектора, соответствующие одному цвету, могут вычисляться параллельно. Это сделало возможным параллельное вычисление элементов сначала во время прямого, затем во время обратного хода SYMGS, но ухудшило сходимость метода. На 12-ти потоках OpenMP ускорение относительно предыдущего варианта с непрерывной памятью составило более 3-х раз, см. рис. 2.

Во втором варианте раскраска производилась всего в 2 цвета (см. рис. 3) вдоль оси Z. Точки, соответствующие плоскостям XY одного цвета, могут рассчитываться параллельно друг другу, в сравнении с предыдущим вариантом раскраски это немного улучшает сходимость, что влияет на производительность, см. рис. 2. Если раскрашивать не по одной, а по две плоскости XY (см. рис. 4), то сходимость улучшается еще больше, и на рис. 2 видно, что этот вариант показывает наилучшие результаты из всех предложенных вариантов раскраски. Производительность этого варианта в 4.8 раза превышает производительность вариант с непрерывной памятью на 12-ти потоках OpenMP. Такой вариант не упоминается ни в одной из других работ [3–5]. Объяснением высокой производительности может быть то, что при раскрашивании по плоскостям происходит больше переиспользования данных в кэше при обращении к памяти, потому что потоки последовательно перебирают элементы, соответствующие соседним точкам одной плоскости.

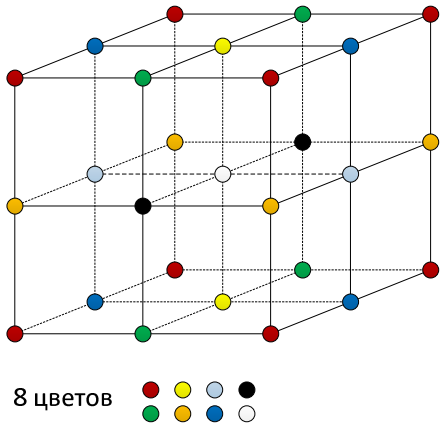

Рис. 1. Раскраска области в 8 цветов

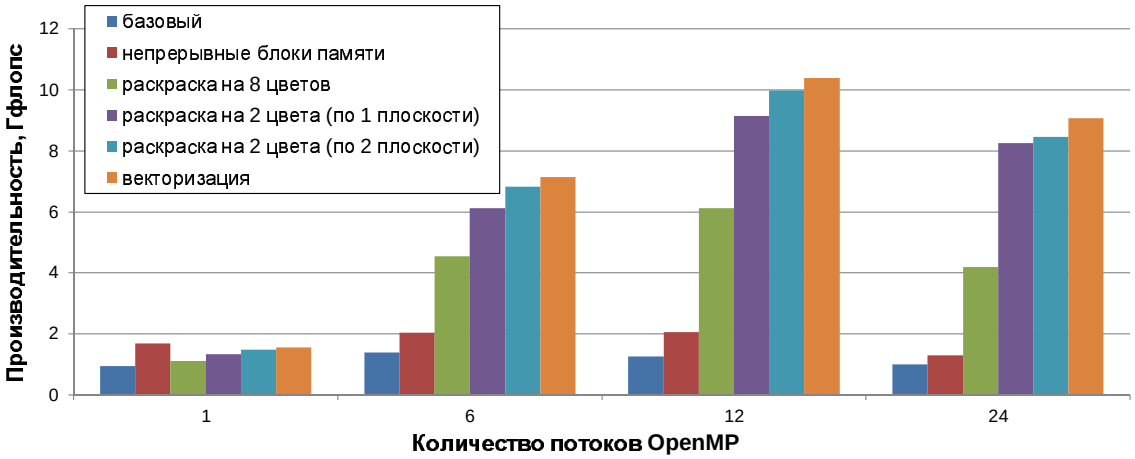

Рис. 2. Производительность одноузловых вариантов оптимизации HPCG

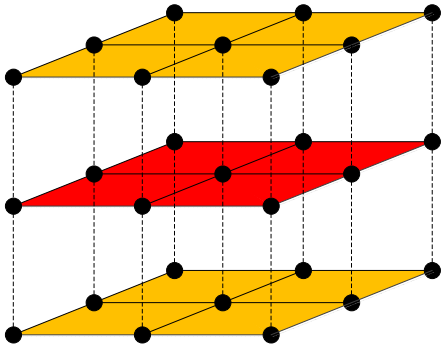

Рис. 3. Раскраска области в 2 цвета

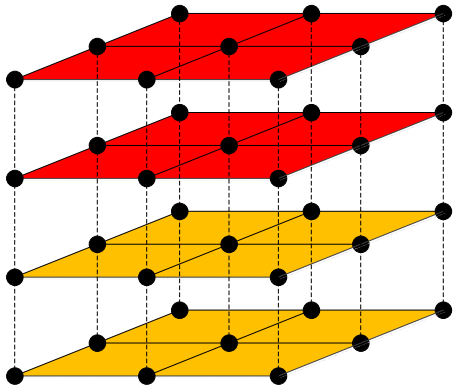

Рис. 4. Раскраска области в 2 цвета по две плоскости

2.3. Векторизация

Следующим этапом оптимизаций была векторизация вычислений. Поскольку во всех строках матрицы, кроме соответствующих граничным элементам, по 27 ненулевых значений, то используемый по умолчанию формат хранения разреженной матрицы можно заменить на формат ELLPACK, в котором количество ненулевых элементов в строке фиксировано, и этот факт используется для упрощения структуры данных. В данной реализации для формата ELLPACK количество ненулевых элементов в строке выбрано равным 28, что требуется для кратности длины массива элементов строки длине векторного регистра расширения системы команд Intel AVX — по 4 значения двойной точности в векторе. Смена формата матрицы позволила улучшить векторизацию кода функций ComputeSYMGS и ComputeSPMV, что обеспечило прирост производительности 5%.

Эффективная векторизация кода функций ComputeSYMGS и ComputeSPMV затруднительна из-за того, что обращения к вектору x являются косвенными, то есть происходят по

индексам, соответствующим ненулевым элементам строк матрицы. Основной фактор, сдерживающий производительность этих функций — пропускная способность чтения из памяти в режиме малой полезной доли данных, выбираемых из памяти при одном обращении. Однако полученный прирост производительности не является окончательным результатом.

2.4. Оптимизация межузловых коммуникаций

Поскольку маршрутизаторы сети «Ангара» создавались с аппаратной поддержкой библиотеки SHMEM [17], для увеличения производительности межузловых коммуникаций использовалась эта библиотека вместо библиотеки MPI [18], используемой в базовой версии HPCG.

В функции ExchangeHalo, которая используется при выполнении функций SYMGS и SPMV для обмена граничными значениями, посылки значений в оптимизированной версии осуществляются при помощи функции записи данных на удаленный узел библиотеки SHMEM (shmem_double_put). Однако после удаленной записи данных в библиотеке SHMEM необходима синхронизация. Первая реализация использовала общую барьерную синхронизацию: каждый процесс ожидал, когда завершатся удаленные записи всех процессов при помощи функции shmem_barrier, и только после этого начинал выполнение основных циклов SYMGS и SPMV. Во втором варианте использовалась синхронизация, реализованная при помощи дополнительных удаленных записей, но только с теми процессами, с которыми происходит обмен. Каждый процесс начинал выполнять основные итерации после того, как получит необходимые данные для вычислений от соседних процессов.

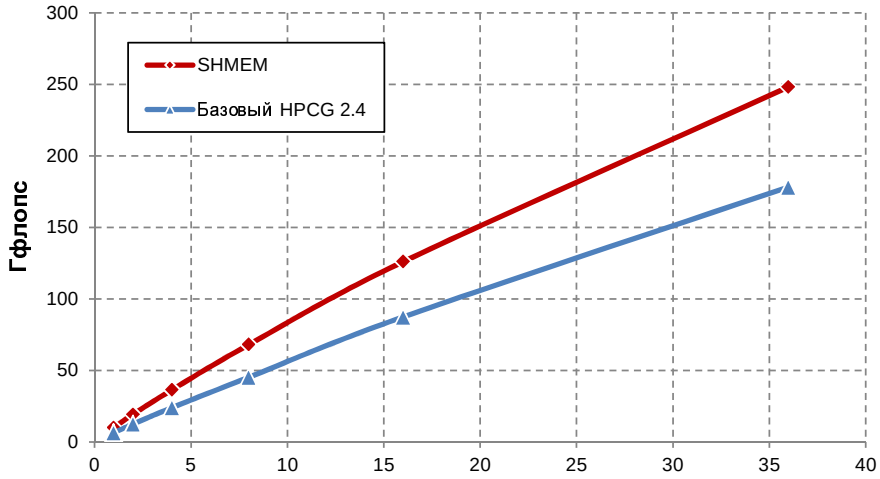

Рис. 5. Производительность на распределенной памяти базового варианта в сравнении с оптимизированным вариантом с использованием библиотеки SHMEM

Сравнение результатов работы каждой реализации представлено на рис. 5. Результаты измерялись на 36-узловом кластере с сетью «Ангара», его параметры приведены в таблице 2. На каждом из узлов при запуске теста HPCG использовалось 8 процессов. Результаты производительности приведены для размера локальной области процесса 16x16x16. На рисунке приведены результаты для базовой версии 2.4 теста и для оптимизированной версии, включающей одноузловые оптимизации (за исключением OpenMP) и использование библиотеки SHMEM.

Диаграмма на рис. 5 показывает, что на 36 узлах выигрыш в производительности оптимизированной версии с использованием SHMEM составляет 39%.

Заключение

Данная работа представляет первую версию оптимизированной реализации теста HPCG для суперкомпьютеров с сетью «Ангара», разработанной в ОАО «НИЦЭВТ». Предлагаемая реализация включает оптимизации уровня вычислительного узла, в том числе изменение формата хранения разреженной матрицы, OpenMP-распараллеливание метода симметричного Гаусса-Зейделя и векторизацию кода. На одном узле перечисленные приемы позволили получить ускорение 4.8 раза по сравнению с базовой версией HPCG 2.4.

Межпроцессные оптимизации обменов в сети «Ангара» выполнены при помощи использования библиотеки SHMEM вместо библиотеки MPI. Полученные в работе результаты на макетном 36-узловом кластере с сетью «Ангара» позволили получить выигрыш 39% по сравнению с базовой версией 2.4 теста HPCG.

Дальнейшая работа будет направлена на объединение OpenMP-распараллеливания и многоузловых оптимизаций, а также на улучшение результатов и более глубокое исследование.

Литература

1. M. Heroux, J. Dongarra, P. Luszczek. HPCG Technical Specification. Sandia Report SAND2013-8752. Printed October 2013. URL: `https://software.sandia.gov/hpcg/doc/HPCG-Specification.pdf` (дата обращения: 10.06.2015).

2. J. Dongarra, P. Luszczek, A. Petitet. The LINPACK Benchmark: Past, Present, and Future // Concurrency and Computation: Practice and Experience. — Vol. 15(9). — 2003. — P. 803–820. ISSN 1532-0634. URL: `http://www.netlib.org/utk/people/JackDongarra/PAPERS/146_2003_the-linpack-benchmark-pa` (дата обращения: 14.06.2015).

3. J. Park, M. Smelyanskiy, K. Vaidyanathan, et al. Efficient shared-memory implementation of high-performance conjugate gradient benchmark and its application to unstructured matricesi // In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC'14). IEEE Press, Piscataway, NJ, USA, — 2014. — P. 945–955. URL: `http://pcl.intel-research.net/publications/sc14_hpcg.pdf` (дата обращения: 12.06.2015).

4. C. Chen, Y. Du, H. Jiang, K. Zuo, C. Yang. HPCG: Preliminary Evaluation and Optimization on Tianhe-2 CPU-only Nodes // In Proceedings of the 2014 IEEE 26th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD '14). — IEEE Computer Society, Washington, DC, USA. — 2014. — P. 41–48.

5. K. Kumahata, K. Minami. HPCG Performance Improvement on the K computer // Presentation at Supercomputers Conference (SC'14), 2014. URL: `http://www.hpcg-benchmark.org/downloads/sc14/HPCG_on_the_K_computer.pdf` (дата обращения: 14.06.2015).

6. E. Phillips, M. Fatica. A CUDA Implementation of the High Performance Conjugate Gradient Benchmark // High Performance Computing Systems. Performance Modeling, Benchmarking, and Simulation. Lecture Notes in Computer Science. — Vol. 8966. — 2015. P. 68–84. URL: http://www.dcs.warwick.ac.uk/~sdh/pmbs14/PMBS14/Workshop_Schedule_files/8-CUDAHPCG.pdf (дата обращения: 12.06.2015).

7. V. Marjanović, J. Gracia, C. Glass. Performance Modeling of the HPCG Benchmark // High Performance Computing Systems. Performance Modeling, Benchmarking, and Simulation Lecture Notes in Computer Science. — Vol. 8966. — 2015. — P. 172–192. URL: https://hpgmg.org/static/MarjanovicGraciaGlass-PerformanceModelHPCG-2014.pdf (дата обращения: 14.06.2015).

8. Лацис, А.О. Какая нам польза от теста HPCG. / Лацис, А.О., Андреев С.С., Плоткина Е.А., Дбар С.А. // Сборник тезисов докладов НСКФ'2014. — 2014. URL: http://2014.nscf.ru/TesisAll/4_Systemnoe_i_promezhytochnoe_PO/06_072_LatsisAO.pdf (дата обращения: 12.06.2015).

9. Киселев А.В., Киселев Е.А., Корнеев В.В. Анализ структуры информационного графа теста HPCG // Научный сервис в сети Интернет: многообразие суперкомпьютерных миров: Труды Международной суперкомпьютерной конференции (22-27 сентября 2014 г., г. Новороссийск). — М.: Изд-во МГУ, 2014. — С. 49–51. URL: http://agora.guru.ru/abrau2014/pdf/049.pdf (дата обращения: 12.06.2015).

10. Макагон, Д.В. Сети для суперкомпьютеров. / Д.В. Макагон, Е.Л. Сыромятников // Открытые системы. СУБД. — 2011. — № 7. — С. 33–37.

11. Корж, А.А. Отечественная коммуникационная сеть 3D-тор с поддержкой глобально адресуемой памяти для суперкомпьютеров транспетафлопсного уровня производительности. / А.А. Корж, Д.В. Макагон, И.А. Жабин, Е.Л. Сыромятников // Паралельные вычислительные технологии (ПаВТ'2010): Труды международной научной конференции (Уфа, 29 марта – 2 апреля 2010 г.). — Челябинск: Издательский центр ЮУрГУ, 2010. — С. 227–237. URL: http://omega.sp.susu.ac.ru/books/conference/PaVT2010/full/134.pdf (дата обращения: 29.04.2015).

12. Симонов, А.С. Разработка межузловой коммуникационной сети с топологией «многомерный тор» и поддержкой глобально адресуемой памяти для перспективных отечественных суперкомпьютеров. / А.С. Симонов, И.А. Жабин, Д.В. Макагон // Научно-техническая конференция «Перспективные направления развития вычислительной техники» (Москва, 28 июня). — Москва: ОАО «Концерн «Вега», 2011. — С. 17–19.

13. Симонов, А.С. Первое поколение высокоскоростной коммуникационной сети «Ангара» / А.С. Симонов, Д.В. Макагон, И.А. Жабин, А.Н. Щербак, Е.Л. Сыромятников, Д.А. Поляков // Наукоемкие технологии. — 2014. — Т. 15, № 1. — С. 21–28.

14. Слуцкин, А.И. Разработка межузловой коммуникационной сети ЕС8430 «Ангара» для перспективных суперкомпьютеров / А.И. Слуцкин, А.С. Симонов, И.А. Жабин, Д.В. Макагон, Е.Л. Сыромятников // Успехи современной радиоэлектроники. — 2012. — № 1. — С. 6–10.

15. Макагон, Д.В. Кристалл для Ангары / И.А. Жабин, Д.В. Макагон, А.С. Симонов // Суперкомпьютеры. — Зима-2013. — С. 46–49.

16. Изгалин С.П., Симонов А.С., Михеев В.А. Отечественные разработки в области суперкомпьютерных технологий. Предварительные результаты оценки производительности ВКС «Ангара» (EC8430) // Доклад на Московском Суперкомпьютерном Форуме, 2014. URL: http://www.osp.ru/netcat_files/18/10/07_Otechestvennye_razrabotki_v_oblasti_superkomp_ (дата обращения: 14.06.2015).

17. Feind, K. Shared Memory Access (SHMEM) Routines. Cray Research, 1995. URL: https://cug.org/5-publications/proceedings_attendee_lists/1997CD/S95PROC/303_308.PDF (дата обращения: 29.04.2015).

18. Message Passing Interface Forum, MPI: A Message-Passing Interface Standard, 1995. URL: http://www.mpi-forum.org/docs/mpi-1.1/mpi-11-html/node64.html (дата обращения: 29.04.2015).

Optimized implementation of HPCG benchmark on supercomputer with “Angara” interconnect

Alexander Agarkov, Alexander Semenov and Alexey Simonov

Keywords: HPCG, "Angara" interconnect, SHMEM, vectorization, OpenMP

The paper presents the optimized implementation of the HPCG benchmark on "Angara" interconnect developed by "NICEVT". HPCG implements conjugate gradient method with preconditioner to solve a sparse linear system. The proposed implementation includes compute node optimizations, such as new sparse matrix storage format, OpenMP-parallelization and vectorization, as well as interprocess communication optimizations targeted for "Angara" interconnect by using the SHMEM library instead of MPI. Compute node optimization includes changing the sparse matrix storage format, OpenMP-parallelization and vectorization. Obtained performance results on the cluster with "Angara" interconnect significantly exceed the results of the HPCG 2.4 reference implementation.