DOI: 10.18721/JCSTCS.10303
УДК 004.4'6

ЭКСПЕРИМЕНТАЛЬНЫЕ АСПЕКТЫ ОЦЕНКИ ПРОПУСКНОЙ СПОСОБНОСТИ ПАМЯТИ КРУПНОМАСШТАБНЫХ СИСТЕМ С АРХИТЕКТУРОЙ CCNUMA

П.Д. Дробинцев, В.П. Котляров, А.В. Левченко

Санкт-Петербургский политехнический университет Петра Великого,
Санкт-Петербург, Российская Федерация

Рассмотрен подход к прогнозированию производительности широкого спектра научных приложений на современных перспективных архитектурах с глобально адресуемой памятью. Исследованы вопросы моделирования и прогнозирования пропускной способности памяти для приложений с гибридным параллелизмом MPI/OpenMP. Тест производительности HPCG использован для создания рабочей нагрузки, репрезентативной широкому спектру вычислительных и коммуникационных задач актуальных научных приложений. Проведены эксперименты по проверке модели на реальном кластере с общей памятью, имеющем архитектуру ccNUMA с 12 ТБ RAM и загруженном в режиме единого образа операционной системы, с целью определения границы размера задачи и демонстрации улучшенных показателей для целевой архитектуры по сравнению с базовой моделью. Модель позволит надежно оценить производительность современных и будущих систем ccNUMA, имеющих более 24 ТБ оперативной памяти на одном узле, и сравнить их экспериментальные результаты с другими проблемно-ориентированными архитектурами во всем мире.

Ключевые слова: оценка производительности; ccNUMA; HPCG; пропускная способность памяти; архитектура NUMA.

Ссылка при цитировании: Дробинцев П.Д., Котляров В.П., Левченко А.В. Экспериментальные аспекты оценки пропускной способности памяти крупномасштабных систем с архитектурой ccNUMA // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. 2017. Т. 10. № 3. С. 32–41. DOI: 10.18721/JCSTCS.10303

EXPERIMENTAL ASPECTS OF MEMORY BANDWIDTH FOR HPC SYSTEMS WITH CCNUMA ARCHITECTURE

P.D. Drobintsev, V.P. Kotlyarov, A.V. Levchenko

Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russian Federation

We have considered an approach to estimating the performance for a wide range of science applications calculated on modern HPC systems with globally addressed memory. Modeling and estimation of memory bandwidth have been examined for a set of applications with parallel structure based on MPI/OpenMP technology. The HPCG benchmark was used to create a workload representing a wide range of calculation and communication tasks in science applications. A set of experiments for checking the model on a real HPC system with globally addressed memory (ccNUMA architecture with 12 Tb of memory with single image of operating system installed) was conducted for estimating the size of the task and highlighting the benefits of

optimized model usage. The optimized model will allow to estimate the performance of modern and future systems developed based on the ccNUMA architecture which contains 24 Tb of memory in one node. The model will also allow to compare the results of NUMA systems with other modern HPC architectures.

Keywords: benchmarking; ccNUMA; HPCG; memory bandwidth; NUMA architecture.

Citation: Drobintsev P.D., Kotlyarov V.P., Levchenko A.V. Experimental aspects of memory bandwidth for HPC systems with ccNUMA architecture. St. Petersburg State Polytechnical University Journal. Computer Science. Telecommunications and Control Systems. 2017, Vol. 10, No. 3, Pp. 32–41. DOI: 10.18721/JCSTCS.10303

Введение

В современных условиях промышленного подхода к решению сложных задач моделирования для организации широкого спектра процессов цифрового управления производством, наукой, сервисом требуются огромные вычислительные мощности суперкомпьютеров. Для их эффективного использования и прогнозирования производительности реализации приложений на современных перспективных архитектурах суперкомпьютеров с глобально адресуемой памятью важны исследования пропускной способности памяти для приложений с гибридным параллелизмом MPI/OpenMP.

Особенности архитектуры ccNUMA

Современные системы с архитектурой Cache-Coherent Uniform Memory Access (ccNUMA) могут предоставить большее количество оперативной памяти на один многомашинный узел (макроузел) с одним образом операционной системы, чем это доступно в стандартном вычислительном кластере. Однако асимметричная природа архитектуры ccNUMA порождает ряд негативных эффектов неоднородности, таких, например, как образование «горячих точек» памяти (memory hot-spotting), варьирующаяся сложная многоуровневая структура задержки удаленного доступа и несовпадение моделей доступа к данным и фактического распределения данных в памяти [1–3]. Эти и другие факторы оказывают разнонаправленное воздействие на пропускную способность памяти макроузла ccNUMA, что порождает серьезные проблемы для научных приложений с интенсивным нерегулярным доступом к памяти. При этом определение величины пропускной способности памяти, исходя из теоретического максимума для конкретной вычислительной системы, является сложной задачей [4]. Таким образом, существующий гипотетический прогноз пропускной способности памяти систем ccNUMA нередко неубедителен.

Главная цель настоящего исследования – надежная оценка производительности современных и будущих систем ccNUMA. В данной статье представлены только предварительные экспериментальные результаты оценки, моделирования и прогнозирования пропускной способности памяти ccNUMA. Тест производительности HPCG (High Performance Conjugate Gradients) использован для создания рабочей нагрузки с низким отношением вычислений к доступу к данным, что характерно для основных коммуникационных и вычислительных моделей современных научных приложений [5].

В работе посредством расширения существующей модели производительности HPCG спрогнозирована эффективная пропускная способность памяти реальной системы с глобально адресуемой памятью, оснащенной 12 ТБ логически неделимой оперативной памяти и загружаемой в режиме единого образа операционной системы. Эксперимент дал возможность сопоставить полученные результаты с результатами других проблемно-ориентированных высокопроизводительных архитектур и спрогнозировать пропускную способность памяти будущих систем ccNUMA. Также в работе продемонстрированы важные технические аспекты, связанные с запуском HPCG на системах с большим объемом памяти одного многомашинного узла.

Предпосылки исследования и существующие работы

Ключевые вопросы существующих пу-

бликаций, связанных с NUMA и HPCG, позволяют учесть широкий спектр проблем, порожденных архитектурой ccNUMA (сюда относятся (1) публикации, касающиеся моделирования производительности гибридного параллелизма MPI/OpenMP, (2) исследования эффектов NUMA, оказывающих влияние на производительность, и (3) публикации, непосредственно связанные с HPCG, в т. ч. описание референсной модели производительности HPCG).

Для гибридного параллелизма в [6] предложена модель, которая используется и как модель оценки пропускной способности памяти, и как модель оптимального распределение ядер. В работе [7] дана ценная информация о моделировании межпроцессорной пропускной способности памяти для анализа производительности различных потоков и локаций размещения данных.

Гибридный подход для разработки высокоуровневых моделей производительности крупномасштабных вычислительных систем, сочетающий математическое моделирование и дискретное случайное моделирование, представлен в работе [8]. В исследовании [9] показаны преимущества гибридного программирования MPI/OpenMP для крупномасштабных кластеров NUMA. Исследования по моделированию производительности коммуникационных задач и вычислений в гибридных приложениях MPI/OpenMP выполнены в работе [9].

Работы по HPCG известны с 2013 г., с момента первой реализации теста. В работе Dongarra et al. [5] дано описание разрешенных и запрещенных оптимизаций HPCG. Несколько исследований [10–16] описывают ранний опыт оптимизации HPCG на Tianhe-2, «Ангара» и Sunway TaihuLight System. В дополнение к практическому опыту эксплуатации теста существует базовая аналитическая модель производительности [4] HPCG, которая является основополагающей для настоящей работы. Модель позволяет оценить время выполнения главных вычислительных и коммуникационных задач для реализации метода симметричного Гаусса-Зейделя (SymGS), вычисления произведения разреженной матрицы на вектор (SpMV), вычисления скалярного произведения векторов (DDOT), вычисления суммы векторов (WAXPBY). Вместе с моделью процедур межузловых коммуникаций полная модель позволит надежно прогнозировать производительность HPCG.

Как следует из сказанного выше, основная ценность исследования состоит в том, что HPCG может внести ясность в вопрос сопоставления производительности ccNUMA с результатами множества проблемно-ориентированных архитектур, отличных от ccNUMA. Эволюционные аспекты и экспериментальное применение упомянутых работ являются вкладом настоящего исследования.

Особенности расширенной модели

Основное достижение настоящей работы – это экспериментальное предсказание пропускной способности памяти системы с архитектурой ccNUMA с использованием эталонной модели [4]. Известными проблемами производительности параллельных приложений на архитектуре ccNUMA являются: (1) локальность доступа к данным; (2) объем обмена данными между потоками; (3) эффективная пропускная способность памяти [17].

Большое значение имеет эффективная пропускная способность памяти, которая является параметром в модели всех вычислительных процедур HPCG. Вклад настоящей работы – в использовании гибридной версии HPCG, а не только версии с чистым MPI, как в модели [4]. Следует учитывать, что HPCG хорошо сбалансирован на уровне MPI, отсюда производительность реализации MPI выше, чем производительность гибридной версии MPI/OpenMP. Кроме того, OpenMP не обеспечивает поддержку ccNUMA. Несмотря на это, идея исследования заключается в том, что гибридная версия является дополнительным серьезным вызовом для многомашинных узлов с архитектурой ccNUMA, провоцируя появление ряда эффектов, критичных для производительности, таких как «горячие точки» памяти (memory hot-spotting).

В табл. 1 показан оценочный диапазон вариантов модели, которые были рассмотрены или имеют такую перспективу. В этой

Таблица 1

Сравнение базовой и расширенной версии модели

Возможности модели	Базовая	Расширенная
SYMGS (время выполнения)	Рассмотрено	+ BW
SpMV (время выполнения)	Рассмотрено	+ BW
WAXPB (время выполнения)	Рассмотрено	+ BW
DDOT (время выполнения)	Рассмотрено	+ BW
Allreduce, Halo (время выполнения)	Рассмотрено	+ Задержка выч. сети
Гибридный параллелизм (MPI/OpenMP)	Не рассмотрено	Рассмотрено
Пропускная способность памяти	Не рассмотрено	Рассмотрено
Задержка вычислительной сети	Не рассмотрено	Рассмотрено
Факторы NUMA	Не рассмотрено	Рассмотрено
Техники оптимизации HPCG	Не рассмотрено	Не рассмотрено

статье предложены пока только экспериментальные аспекты эффективной оценки пропускной способности памяти (BW).

Из негибридной модели известно полное время выполнения [4]:

$$Iter_{time(\sec)} = MG + SpMV(depth = 0) + \\ + 3(DDOT + WAXPB).$$

Гибридная версия HPCG (MPI/OpenMP) более чувствительна к пропускной способности памяти, чем версия MPI, и может иметь большую производительность [18], особенно для случая системы с глобально адресуемой памятью. Для OpenMP модель времени выполнения предложена ранее в работе Wu et al. [19]:

$$Perf = (\mathrm{Re}\, f_{MPI} + OMP) \times \\ \times \frac{Total_{exec\_time(\sec)}}{Comp_{exec\_time(\sec)} + Comm_{exec\_time(\sec)}},$$

где *OMP* представляет модель внутриузловой производительности OpenMP:

$$OMP = T_{c_1} + (BW_n - 1)\frac{T_{c_2} - T_{c_1}}{BW_2 - 1}.$$

Здесь использована формула *OMP* для моделирования времени выполнения приложения OpenMP на *n* ядрах, основанная на модели производительности для 1 и 2 ядер (Tc) и пропускной способности памяти (BWn) [19]. Пропускную способность памяти можно получить из базовой модели [4] для каждого вычислительного ядра HPCG следующим образом:

$$BW_{SYMGS}(Bytes / \sec) = \\ = \frac{(nx \times ny \times nz) / 2^{3\times d} \times (20 + 20 \times 27)(Bytes)}{SYMGS_{exec\_time(\sec)}},$$

$$BW_{SpMV}(Bytes / \sec) = \\ = \frac{(nx \times ny \times nz) / 2^{3\times d} \times (20 + 20 \times 27)(Bytes)}{SpMV_{exec\_time(\sec)}},$$

$$BW_{WAXPB}(Bytes / \sec) = \\ = \frac{(nx \times ny \times nz) / 2^{3\times d} \times 24(Bytes)}{WAXPB_{exec\_time(\sec)}},$$

$$BW_{DDOT}(Bytes / \sec) = \\ = \frac{(nx \times ny \times nz) / 2^{3\times d} \times 16(Bytes)}{DDOT_{exec\_time(\sec)}},$$

где *SYMGS* является наиболее затратным [20].

Несмотря на то, что все вычислительные процедуры были смоделированы исчерпывающим образом, остаются важные параметры, полученные эмпирически. К их числу, помимо пропускной способности памяти, относится задержка вычислительной сети. Влияние задержки на прогноз производительности HPCG оценивается авторами базовой модели [4] как незначительное. В работе для эмпирической оценки латентности вычислительной сети использован модуль KNEM ядра Linux, обеспечивающий высокопроизводительную внутриузловую MPI-связь для больших сообщений [21].

Таблица 2

Варианты конфигурации

Особенности архитектуры	Базовый сервер	Макроузлы с единым образом ОС		
		Минимальный	Средний	Максимальный
RAM	188 ГБ	752 ГБ	3 ТБ	12 ТБ
Узлы NUMA	6	24	96	384
Плата/Сокет/Ядра	1/3/48	4/12/192	16/48/768	64/192/3072

Экспериментальные результаты

Поскольку системы ccNUMA с объемом памяти более 3 ТБ являются достаточно редкими, и кажется сложным получить набор различных гигантских многомашинных узлов, мы используем систему ccNUMA в следующих конфигурациях (табл. 2).

Для более глубокого изучения проблем, связанных с NUMA, проведены эксперименты с гибридной версией HPCG. Данная версия была запущена на макроузлах с 188 ГБ RAM (48 ядер) с объединением памяти макроузла до 3 ТБ RAM (768 ядер) и с последующей интеграцией в один макроузел с объемом памяти до ≈12 ТБ RAM (3072 ядра) на заключительном этапе. Базовый для многомашинного узла сервер основан на процессоре AMD Opteron 6380, межсоединение имеет топологию 3D Torus. Используем ядро Linux 4.11 с набором патчей для поддержки драйвера Block Transfer Engine для контроллеров узлов NumaChip.

HPCG был скомпилирован с оптимизированной библиотекой *libgomp*, которая поддерживает локальное хранилище потоков (thread-local storage – TLS) для данных с числом потоков более 1024. Отдельный стек размером до 2 ГБ выделялся под каждый поток HPCG. Генерация инструкций для предварительной выборки памяти использовалась для повышения производительности циклов, которые обращаются к большим массивам. Нагрузка балансировалась для повышения эффективности приложения OpenMP, распределяя потоки через все доступные узлы NUMA, используя FPU и уменьшая нагрузку на интерфейс памяти и кэш третьего уровня. Генерация инструк-

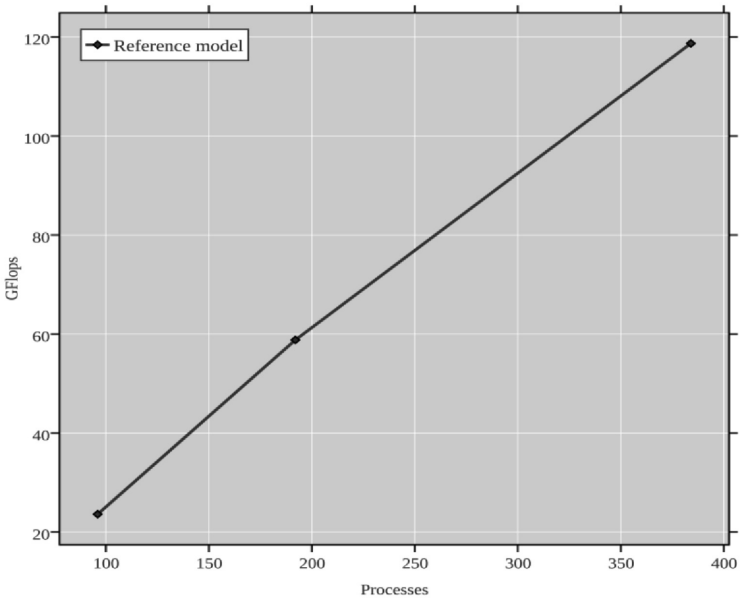

Рис. 1. Прогноз производительности HPCG, основанный на использовании базовой модели

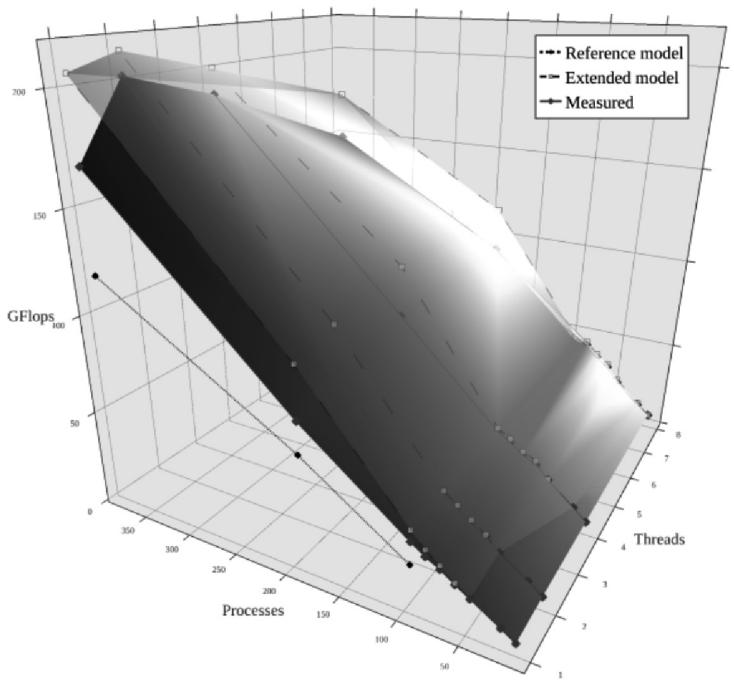

Рис. 2. Прогнозируемые и фактически полученные результаты HPCG на целевой крупномасштабной системе с 12 ТБ RAM

ций для предварительной выборки памяти использовалась для повышения производительности циклов обработки больших массивов. Результаты моделирования с помощью базовой модели приведены на рис. 1.

На рис. 2 сделанные прогнозы сопоставлены с фактическими результатами гибридного HPCG на макроузле с 12 ТБ оперативной памяти. Кроме того, приведены результаты сравнения с эталонной моделью. В отличие от результатов работы [4], гибридный вариант HPCG нелинейно

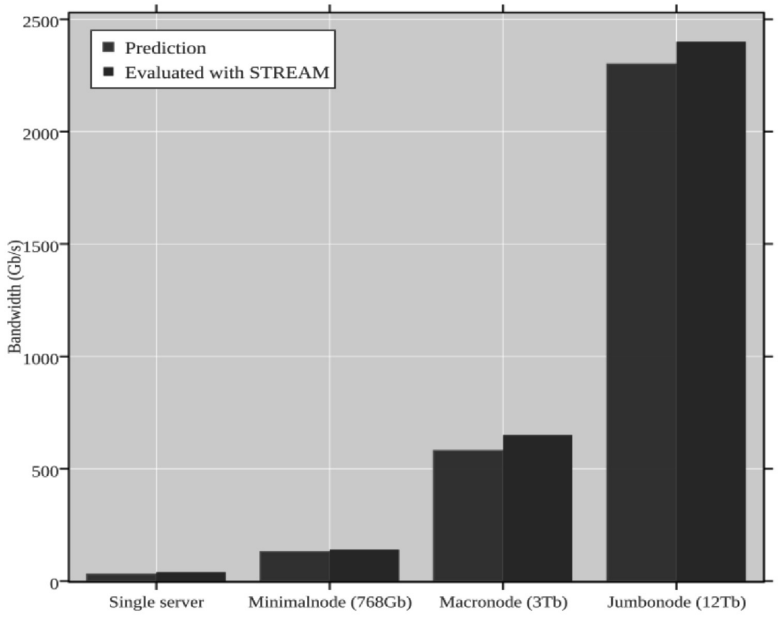

Рис. 3. Результаты теста STREAM и прогноз

масштабируется в системе ccNUMA; неравномерность системы приводит к разделению поверхности, причины которого исследуются. На рис. 3 приведено сравнение смоделированной пропускной способности памяти макроузла и результатов STREAM Benchmark.

В прогнозах производительности будущих систем ccNUMA с объемом памяти более 24 ТБ можно предположить, что пропускная способность памяти будет оставаться узким местом производительности HPCG. При крупномасштабном запуске HPCG с максимальным размером задачи около 7 ТБ памяти оказывается занято, причем прогнозируется пропорционально высокое потребление памяти, поскольку будущие системы ccNUMA будут иметь по крайней мере 4 ГБ RAM на ядро. Задержка вычислительной сети будет расти. Основываясь на данных моделирования, ожидается производительность не менее 400 GFlops для макроузла с 24 ТБ оперативной памяти.

В отношении существующих машин без ccNUMA, HPCG становится единой метрикой для сравнения различных проблемно-ориентированных архитектур и уменьшает разрыв в производительности, созданный оценками, сделанными на основе LINPACK. Например, экспериментальная система ccNUMA демонстрирует удовлетворительную производительность HPCG по сравнению с результатами технического отчета [15] для суперкомпьютера Sunway TaihuLight [22], предполагая, что память ccNUMA медленнее по сравнению с нынешним лидерам TOP500.

Заключение

В статье представлен экспериментальный подход к оценке пропускной способности памяти современных систем ccNUMA. HPCG Benchmark использован для создания рабочей нагрузки, сопоставимой с современными научными приложениями. Существующая модель производительности HPCG расширена за счет использования гибридного MPI/OpenMP и дополнена факторами, влияющими на пропускную способность памяти. В результате предсказана пропускная способность памяти реальной системы ccNUMA с 12 ТБ оперативной памяти. Используемый подход может применяться для сравнения текущих и будущих машин ccNUMA.

Показано (рис. 3), что расхождение между фактически полученными результатами STREAM Benchmark и выведенными из эталонных моделей составляет до 12 %. Хотя вся программная среда была оптимизирована в широких масштабах (ядро Linux, gcc, libgomp, etc.), однако значительные оптимизации HPCG возможны. В ближайшем будущем планируется сосредоточиться на реализации существующих оптимизаций HPCG для случая ccNUMA, поскольку повышение производительности HPCG улучшит производительность реальных приложений [5].

В частности будет уточнена модель локализации кэш-памяти с помощью новой техники оптимизации HPCG, предложенной в статье [16]. Среди других улучшений предполагается заменить формат хранения матрицы CSR на упрощенный SELLPACK для ядер SpMV и SYMGS [10, 16]

Таблица 3

Оценка результатов

Техники оптимизации	Ожидаемые результаты
Раскраска вдоль двух областей *XY* одновременно в SYMGS [16]	Улучшение производительности SYMGS в три раза
Смена формата хранения (ELLPACK вместо CSR) [16]	Ускорение на 5 % в SYMGS и SPMV
Перераспределение данных [23]	Улучшение производительности для больших задач HPCG

(табл. 3 иллюстрирует ожидаемое ускорение). В работе [23] представлена новая модель перераспределения данных, позволяющая снизить издержки удаленного доступа к памяти для приложений с интенсивным вычислением с большими размерами проблемы. Наконец, в ближайшем будущем мы планируем предложить модель задержки вычислительной сети для многомашинного узла.

СПИСОК ЛИТЕРАТУРЫ

1. **Li T., Ren Y., Yu D., Jin S., Robertazzi T.** Characterization of input/output bandwidth performance models in NUMA architecture for data intensive applications // Proc. of the 42nd Internat. Conf. on Parallel Processing. 2013. Pp. 369–378.

2. **Diener M., Cruz E.H., Navaux P.O.** Modeling memory access behavior for data mapping // Internat. Journal of High Performance Computing Applications. 2016. URL: http://hpc.sagepub.com/content/early/2016/04/13/1094342016640056.abstract

3. **Zeng D., Zhu L., Liao X., Jin H.** A Data-Centric Tool to Improve the Performance of Multithreaded Program on NUMA. Springer International Publishing, 2015. Pp. 74–87 // URL: http://dx.doi.org/10.1007/978-3-319-27140-8_6

4. **Marjanović V., Gracia J., Glass C.W.** Performance modeling of the HPCG benchmark // High Performance Computing Systems. Performance Modeling, Benchmarking, and Simulation: 5th Internat. Workshop. New Orleans, LA, USA, 2014. Revised Selected Papers. Springer International Publishing, 2015. Pp. 172–192. URL: http://dx.doi.org/10.1007/978-3-319-17248-4_9

5. **Dongarra J., Heroux M.A., Luszczek P.** High-performance conjugate-gradient benchmark: A new metric for ranking high-performance computing systems // Internat. Journal of High Performance Computing Applications. 2016. Vol. 30. No. 1. Pp. 3–10. URL: http://dx.doi.org/10.1177/1094342015593158

6. **Wang W., Davidson J.W., Soffa M.L.** Predicting the memory bandwidth and optimal core allocations for multi-threaded applications on large-scale NUMA machines // Proc. of the IEEE Internat. Symp. on High Performance Computer Architecture. 2016. Pp. 419–431.

7. **Luo H., Brock J., Li P., Ding C., Ye C.** Compositional model of coherence and NUMA eoɟects for optimizing thread and data placement // Proc. of the IEEE Internat. Symp. on Performance Analysis of Systems and Software. 2016. Pp. 151–152.

8. **Pllana S., Benkner S., Xhafa F., Barolli L.** Hybrid performance modeling and prediction of large-scale computing systems // Proc. of the Internat. Conf. on Complex, Intelligent and Software Intensive Systems, 2008. Pp. 132–138.

9. **Adhianto L., Chapman B.** Performance Modeling of Communication and Computation in Hybrid MPI and OpenMP Applications // Proc. of the 12th Internat. Conf. on Parallel and Distributed Systems. 2006. Vol. 2. Pp. 6.

10. **Zhang X., Yang C., Liu F., Liu Y., Lu Y.** Optimizing and scaling HPCG on Tianhe-2: Early experience // 14th Internat. Conf. on Algorithms and Architectures for Parallel Processing. Dalian, China, 2014. Proceedings, Part I. Springer International Publishing, 2014. Pp. 28–41. URL: http://dx.doi.org/10.1007/978-3-319-11197-1_3

11. **Chen C., Du Y., Jiang H., Zuo K., Yang C.** HPCG: Preliminary evaluation and optimization on Tianhe-2 CPU-only nodes // Computer Architecture and High Performance Computing. Proc. of the IEEE 26th Internat. Symp. on. 2014. Pp. 41–48.

12. **Liu Y., Zhang X., Yang C., Liu F., Lu Y.** Accelerating HPCG on Tianhe-2: A hybrid CPU-MIC algorithm // Proc. of the 20th IEEE Internat. Conf. on Parallel and Distributed Systems. 2014. Pp. 542–551.

13. **Liu F., Yang C., Liu Y., Zhang X., Lu Y.** Reducing communication overhead in the high performance conjugate gradient benchmark on Tianhe-2 // Proc. of the 13th Internat. Symp. on Distributed Computing and Applications to Business, Engineering and Science, 2014, Pp. 13–18.

14. **Liu Y., Yang C., Liu F., Zhang X., Lu Y., Du Y. et al.** 623 Tflop/s HPCG run on Tianhe-2: Leveraging millions of hybrid cores // Internat. Journal of High Performance Computing Applications. 2016. Vol. 30. No. 1. Pp. 39–54. URL: http://hpc.sagepub.com/content/30/1/39.abstract

15. **Dongarra J.** Report on the Sunway Taihulight System. Oak Ridge National Laboratory, Department of Electrical Engineering and Computer Science, University of Tennessee. Tech. Rep. UT-EECS-16-742. Jun. 2016.

16. **Agarkov A., Semenov A., Simonov A.** Optimized implementation of HPCG benchmark on supercomputer with “Angara” interconnect // Proc. of the 1st Russian Conf. on Supercomputing – Supercomputing Days 2015. Moscow State University, 2015. No. 1. Pp. 294–302. URL: http://ceur-ws.org/Vol-1482/294.pdf

17. **Yang R., Antony J., Rendell A.P.** A simple performance model for multithreaded applications executing on non-uniform memory access computers // High Performance Computing and Commu-

nications. 2009.

18. **Nakajima K.** Flat MPI vs. Hybrid: Evaluation of parallel programming models for preconditioned iterative solvers on "T2K Open Supercomputer" // Proc. of the Internat. Conf. on Parallel Processing Workshops. 2009. Pp 73–80.

19. **Wu X., Taylor V.** Performance modeling of hybrid MPI/OpenMP scientific applications on large-scale multicore cluster systems // Proc. of the IEEE 14th Internat. Conf. on Computational Science and Engineering. 2011. Pp. 181–190.

20. **Park J., Smelyanskiy M., Vaidyanathan K., Heinecke A., Kalamkar D.D., Liu X., et al.** Efficient shared-memory implementation of high-performance conjugate gradient benchmark and its application to unstructured matrices // Proc. of the Internat. Conf. for High Performance Computing, Networking, Storage and Analysis. 2014. Pp. 945–955.

21. **Goglin B., Moreaud S.** Knem: A generic and scalablekernelassisted intra-node mpi communication framework // J. Parallel Distrib. Comput. 2013. Vol. 73. No. 2. Pp. 176–188. URL: http://dx.doi.org/10.1016/j.jpdc.2012.09.016

22. **Fu H., Liao J., Yang J., Wang L., Song Z., Huang X., Yang C., et al.** The Sunway TaihuLight supercomputer: system and applications // Science China Information Sciences. 2016. Vol. 59. No. 7. Pp. 1–16. URL: http://dx.doi.org/10.1007/s11432-016-5588-7

23. **Zhang M., Gu N., Ren K.** Optimization of computation-intensive applications in cc-NUMA architecture // Proc. of the Internat. Conf. on Networking and Network Applications. 2016. Pp. 244–249.

Статья поступила в редакцию 02.08.2017

REFERENCES

1. **Li T., Ren Y., Yu D., Jin S., Robertazzi T.** Characterization of input/output bandwidth performance models in NUMA architecture for data intensive applications. *Proc. of the 42nd International Conference on Parallel Processing*, 2013, Pp. 369–378.

2. **Diener M., Cruz E.H., Navaux P.O.** Modeling memory access behavior for data mapping. *International Journal of High Performance Computing Applications*. 2016. Available: http://hpc.sagepub.com/content/early/2016/04/13/1094342016640056.abstract

3. **Zeng D., Zhu L., Liao X., Jin H.** *A Data-Centric Tool to Improve the Performance of Multithreaded Program on NUMA*. Springer International Publishing, 2015, Pp. 74–87. Available: http://dx.doi.org/10.1007/978-3-319-27140-8_6

4. **Marjanović V., Gracia J., Glass C.W.** Performance modeling of the HPCG benchmark. *High Performance Computing Systems. Performance Modeling, Benchmarking, and Simulation: 5th International Workshop, PMBS 2014*, New Orleans, LA, USA, 2014. Revised Selected Papers. Springer International Publishing, 2015, Pp. 172–192. Available: http://dx.doi.org/10.1007/978-3-319-17248-4_9

5. **Dongarra J., Heroux M.A., Luszczek P.** High-performance conjugate-gradient benchmark: A new metric for ranking high-performance computing systems. *International Journal of High Performance Computing Applications*, 2016, Vol. 30, No. 1, Pp. 3–10. Available: http://dx.doi.org/10.1177/1094342015593158

6. **Wang W., Davidson J.W., Soffa M.L.** Predicting the memory bandwidth and optimal core allocations for multi-threaded applications on large-scale NUMA machines. *Proc. of the IEEE International Symposium on High Performance Computer Architecture*, 2016, Pp. 419–431.

7. **Luo H., Brock J., Li P., Ding C., Ye C.** Compositional model of coherence and NUMA eoɟects for optimizing thread and data placement. *Proc. of the IEEE International Symposium on Performance Analysis of Systems and Software*, 2016, Pp. 151–152.

8. **Pllana S., Benkner S., Xhafa F., Barolli L.** Hybrid performance modeling and prediction of large-scale computing systems. *Proc. of the International Conference on Complex, Intelligent and Software Intensive Systems*, 2008, Pp. 132–138.

9. **Adhianto L. Chapman B.** Performance Modeling of Communication and Computation in Hybrid MPI and OpenMP Applications. *Proc. of the 12th International Conference on Parallel and Distributed Systems*, 2006, Vol. 2, Pp. 6.

10. **Zhang X., Yang C., Liu F., Liu Y., Lu Y.** Optimizing and scaling HPCG on Tianhe-2: Early experience. *14th International Conference on Algorithms and Architectures for Parallel Processing*. Dalian, China, 2014. Proceedings, Part I. Springer International Publishing, 2014, Pp. 28–41. Available:http://dx.doi.org/10.1007/978-3-319-11197-1_3

11. **Chen C., Du Y., Jiang H., Zuo K., Yang C.** HPCG: Preliminary evaluation and optimization on Tianhe-2 CPU-only nodes. *Proc. of the IEEE 26th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD),* 2014, Pp. 41–48.

12. **Liu Y., Zhang X., Yang C., Liu F., Lu Y.** Accelerating HPCG on Tianhe-2: A hybrid CPU-MIC algorithm. *Proc. of the 20th IEEE International Con-*

ference on Parallel and Distributed Systems, 2014, Pp. 542–551.

13. **Liu F., Yang C., Liu Y., Zhang X., Lu Y.** Reducing communication overhead in the high performance conjugate gradient benchmark on Tianhe-2. *Proc. of the 13th International Symposium on Distributed Computing and Applications to Business, Engineering and Science*, 2014, Pp. 13–18.

14. **Liu Y., Yang C., Liu F., Zhang X., Lu Y., Du Y., et al.** 623 Tflop/s HPCG run on Tianhe-2: Leveraging millions of hybrid cores. *International Journal of High Performance Computing Applications*, 2016, Vol. 30, No. 1, Pp. 39–54. Available: http://hpc.sagepub.com/content/30/1/39.abstract

15. **Dongarra J.** *Report on the Sunway Taihulight System*. Oak Ridge National Laboratory, Department of Electrical Engineering and Computer Science, University of Tennessee, Tech. Rep. UT-EECS-16-742, Jun 2016.

16. **Agarkov A., Semenov A., Simonov A.** Optimized implementation of HPCG benchmark on supercomputer with "Angara" interconnect. *Proceedings of the 1st Russian Conference on Supercomputing – Supercomputing Days 2015,* Moscow State University, 2015, No. 1, Pp. 294–302. Available: http://ceur-ws.org/Vol-1482/294.pdf

17. **Yang R., Antony J., Rendell A.P.** A simple performance model for multithreaded applications executing on non-uniform memory access computers. *High Performance Computing and Communications*, 2009.

18. **Nakajima K.** Flat MPI vs. Hybrid: Evaluation of parallel programming models for preconditioned iterative solvers on "T2K Open Supercomputer". *Proc. of the International Conference on Parallel Processing Workshops*, 2009, Pp 73–80.

19. **Wu X., Taylor V.** Performance modeling of hybrid MPI/OpenMP scientific applications on large-scale multicore cluster systems. *Proc. of the IEEE 14th International Conference on Computational Science and Engineering*, 2011, Pp. 181–190.

20. **Park J., Smelyanskiy M., Vaidyanathan K., Heinecke A., Kalamkar D.D., Liu X., et al.** E .cient shared-memory implementation of high-performance conjugate gradient benchmark and its application to unstructured matrices. *Proc. of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2014, Pp. 945–955.

21. **Goglin B., Moreaud S.** Knem: A generic and scalablekernelassisted intra-node mpi communication framework. *J. Parallel Distrib. Comput.*, 2013, Vol. 73, No. 2, Pp. 176–188. Available: http://dx.doi.org/10.1016/j.jpdc.2012.09.016

22. **Fu H., Liao J., Yang J., Wang L., Song Z., Huang X., Yang C., et al.** The Sunway TaihuLight supercomputer: system and applications. *Science China Information Sciences*, 2016, Vol. 59, No. 7, Pp. 1–16. Available: http://dx.doi.org/10.1007/s11432-016-5588-7

23. **Zhang M., Gu N., Ren K.** Optimization of computation-intensive applications in cc-NUMA architecture. *Proc. of the International Conference on Networking and Network Applications*, 2016, Pp. 244–249.

Received 02.08.2017

СВЕДЕНИЯ ОБ АВТОРАХ / THE AUTHORS

ДРОБИНЦЕВ Павел Дмитриевич
DROBINTSEV Pavel D.
E-mail: drob@ics2.ecd.spbstu.ru

КОТЛЯРОВ Всеволод Павлович
KOTLYAROV Vsevolod P.
E-mail: vpk@spbstu.ru

ЛЕВЧЕНКО Алексей Викторович
LEVCHENKO Aleksei V.
E-mail: 2@expx.org