ПРОГРАММИРОВАНИЕ, 2019, № 6, с. 30–39

ПАРАЛЛЕЛЬНОЕ И РАСПРЕДЕЛЕННОЕ ПРОГРАММИРОВАНИЕ

УДК 004.42

ОЦЕНКА ПРОИЗВОДИТЕЛЬНОСТИ ГИБРИДНОГО ВЫЧИСЛИТЕЛЬНОГО КЛАСТЕРА НА БАЗЕ ПРОЦЕССОРОВ IBM POWER8

© 2019 г. С. И. Мальковский[a], А. А. Сорокин[a,*], С. П. Королев[a], А. А. Зацаринный[b,], Г. И. Цой[a]**

[a] *Вычислительный центр ДВО РАН*
680000 Хабаровск, ул. Ким Ю Чена, 65, Россия

[b] *Федеральный исследовательский центр "Информатика и управление" Российской академии наук*
119333, Москва, Вавилова, д. 44, кор. 2, Россия

**E-mail: alsor@febras.net*

***E-mail: alex250451@mail.ru*

Поступила в редакцию 10.01.2019 г.
После доработки 25.02.2019 г.
Принята к публикации 25.02.2019 г.

Работа посвящена оценке производительности гибридного вычислительного кластера, построенного с использованием центральных процессоров IBM POWER8 и сопроцессоров NVIDIA Tesla P100 GPU. Приводится краткое описание архитектуры вычислительной системы и используемого программного обеспечения. Обсуждаются результаты проведенных экспериментов с использованием пакетов STREAM, NPB, Crossroads/NERSC-9 DGEMM, HPL. Сформулированы выводы об эффективности технологии одновременной многопоточности (simultaneous multithreading, SMT) процессоров POWER8, а также производительности компиляторов, специальных и математических библиотек на рассматриваемой архитектуре.

DOI: 10.1134/S0132347419060050

1. Введение. В настоящее время одним из наиболее перспективных направлений развития высокопроизводительных вычислительных систем являются гибридные решения [1], предусматривающие в дополнение к центральным процессорам установку различных "ускорителей вычислений" или сопроцессоров. При подключении к центральному процессору по высокоскоростной шине такие сопроцессоры берут на себя большую часть вычислительной работы.

Впервые появившись в рейтинге top500 [2] самых мощных вычислительных систем в 2006 году, в настоящий момент гибридные вычислительные системы обеспечивают 41% (на ноябрь 2018 года) его суммарной производительности. При этом в первой десятке мирового рейтинга число подобных систем составляет уже 70% и будет в дальнейшем неизбежно расти. Эта тенденция объясняется по крайней мере двумя факторами. Первый обусловлен высокой энергоэффективностью и производительностью специализированных сопроцессоров (графические процессоры, мультипроцессоры и т.д.) по сравнению с процессорами общего назначения, что становится особенно актуальным при создании высокопроизводительных вычислительных систем (до петафлопс). Второй фактор состоит в возможности высокой эффективности решения задач в областях машинного обучения, глубокого обучения и искусственного интеллекта, переживающих сейчас бурный рост.

Среди вычислительных платформ подобного класса можно выделить архитектуру POWER [3], на основе которой создан ряд специализированных вычислительных систем, включающих самые мощные в настоящее время суперкомпьютерные установки в мире – Summit и Sierra с пиковой производительностью большее 200 и 125 ПФлопс соответственно (1 и 2 место в top500 за ноябрь 2018 г.). Однако низкий процент распространения архитектуры POWER в отрасли, не позволяет в полной мере оценить ее вычислительные возможности и эффективность использования для решения актуальных научных задач.

Поэтому задача оценки производительности вычислительной системы, построенной на основе гибридной архитектуры с использованием процессоров IBM POWER8 и ускорителей NVIDIA Tesla P100 GPU, которой посвящена статья, приобретает несомненную актуальность. На основе результатов исследования работы компиляторов, математических библиотек и технологий параллельного программирования в статье сфор-

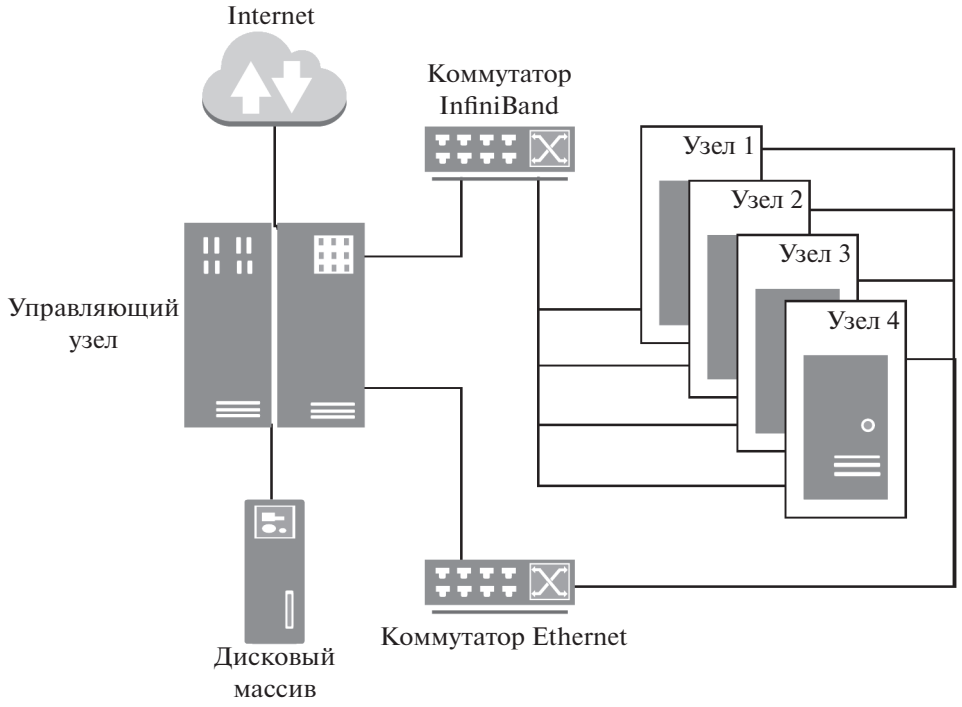

Рис. 1. Схема вычислительной системы.

мулированы научно-практические рекомендации по использованию подобных архитектур, а также эффективных режимах их эксплуатации.

2. Архитектура и программное обеспечение вычислительного кластера. Исследуемая вычислительная система имеет кластерную архитектуру и состоит из одного управляющего и четырех вычислительных узлов (рис. 1). Все они имеют одинаковую конфигурацию за исключением узла управления, который оснащен контроллером Fibre Channel с пропускной способностью в 16 Гбит/с для подключения к внешнему дисковому хранилищу. В качестве сети управления используется сеть, построенная по технологии Gigabit Ethernet, а для передачи данных – сеть EDR InfiniBand с пропускной способностью 100 Гбит/с.

Все узлы кластера представляют собой серверы Sitonica PW22LC (IBM Power Systems S822LC 8335-GTB) в следующей типовой комплектации:

- два процессора IBM POWER8 с максимальной частотой 4.023 ГГц;
- два сопроцессора NVIDIA Tesla P100 GPU с 256 ГБ DDR4 ОЗУ;
- два жестких диска Seagate 1 ТБ 7200RPM;
- контроллер EDR InfiniBand.

Рассмотрим подробно архитектуру и особенности вычислительной системы.

Процессор IBM POWER8 – это суперскалярный процессор, базирующийся на RISC архитектуре (“компьютер с сокращенным набором команд”) POWER [4]. Он содержит 10 вычислительных ядер, поддерживающих технологию одновременной многопоточности (SMT) [5] для восьми потоков и внеочередное выполнение команд. Список всех доступных режимов работы технологии SMT приводится в таблице 1. Каждое вычислительное ядро содержит 32 КБ кэша L1 инструкций, 64 КБ кэша L1 данных и 512 КБ кэша L2. Также на каждое ядро приходится 8 МБ eDRAM кэша третьего уровня, разделяемого между всеми вычислительными ядрами (всего 80 МБ на процессор) [6]. Дополнительный разделяемый кэш L4, реализованный по технологии eDRAM, находится вне процессора на микросхемах Centaur, обеспечивающих планирование и управление обменами данных с памятью (по микросхеме на канал памяти). Его объем составляет 16 МБ на микросхему. Таким образом, на процессор POWER8, имеющий восьмиканальный контроллер памяти, может приходиться до 128 МБ кэша L4.

Вычислительные ядра процессора содержат по 16 исполнительных устройств, из которых 4 используются для выполнения операций над числами с плавающей запятой. При этом поддерживается одновременное выполнение до двух операций над векторными регистрами размером 128 бит, содержащими 4 числа с плавающей запятой одинарной точности или 2 – двойной [4]. Это дает (при выполнении операций умножения-сложения – FMA) возможность выполнения до 8 операций с плавающей запятой за такт. Исходя из

Таблица 1. Режимы работы процессорных ядер

Название	Аппаратных потоков на ядро
ST	1
SMT2	2
SMT4	4
SMT8	8

Таблица 2. Список доступных компиляторов и библиотек

Название ПО	Версия	Поддерживаемые стандарты
Компиляторы		
IBM XL C/C++	13.1.5	C11, C++11
IBM XL Fortran	15.1.5	Fortran 2003, Fortran 2008
GNU C/C++, Fortran	4.8.5	C11, C++11, Fortran 2003, Fortran 2008
PGI C/C++, Fortran	17.4	C11, C++14, Fortran 2003
Библиотеки		
IBM Spectrum MPI	10.1	MPI-3.1
OpenMPI	2.0.2a1	MPI-3.1
IBM ESSL	5.5	–
IBM PESSL	5.2	–

этого может быть определена пиковая производительность процессора, составляющая 0.322 ТФлопс.

Сопроцессоры NVIDIA Tesla P100 GPU, которыми оснащены узлы кластера, реализованы на архитектуре Pascal [7]. Графический процессор каждого из них содержит 56 потоковых процессора (SM), включающих 64 CUDA ядра для выполнения операций над числами с плавающей запятой одинарной точности, 32 CUDA ядра для выполнения операций над числами с плавающей запятой двойной точности, 24 КБ разделяемого кэша L1 и 64 КБ разделяемой памяти. Таким образом для выполнения операций над числами двойной точности могут использоваться 1792 CUDA ядра. Помимо кэша L1, доступного в рамках одного SM, этим ядрам доступно 4 МБ кэша L2 и 16 ГБ памяти HBM2 (полоса пропускания 732 ГБ/с). Пиковая производительность одного графического процессора частотой 1.48 ГГц на операциях с числами двойной точности (при выполнении операций FMA) составляет 5.3 ТФлопс. Сопроцессоры подключаются к центральным процессорам вычислительных узлов по шине NVLink версии 1.0 с пропускной способностью в 80 ГБ/с.

Для проведения расчетов доступно 10 центральных процессоров (100 вычислительных ядер) и 10 сопроцессоров. Таким образом пиковая производительность всего кластера составляет (2 × 0.322 ТФлопс + 2 × 5.3 ТФлопс) × 5 узлов = = 56.22 ТФлопс, большая часть которой обеспечивается графическими процессорами.

Все узлы вычислительного кластера работают под управлением операционной системы CentOS 7.3. В качестве общей файловой системы при проведении расчетов используется файловая система NFS (Network File System). Для диспетчеризации заданий применяется свободная версия программного обеспечения PBS Professional 14.1. На момент исследования на кластере были установлены компиляторы и библиотеки, список которых приведен в таблице 2.

Для работы с сопроцессорами развернута среда разработки NVIDIA CUDA Toolkit 8.0.61, конкурентное выполнение программного кода нескольких вычислительных процессов, выгружаемого на один GPU, обеспечивается сервисом NVIDIA MPS [8]. Мониторинг кластера осуществляется пакетом Ganglia.

3. Методика тестирования. Исследования производились на гибридном вычислительном кластере с характеристиками, приведенными в п. 1. Все тесты запускались в монопольном режиме. На рассматриваемой системе изучались отдельные вопросы производительности ее аппаратного и программного обеспечения при выполнении параллельных приложений. В частности, оценивалась пропускная способность подсистемы памяти, проводилось сравнение компиляторов и технологий параллельного программирования, исследовались возможности технологии SMT. Также была оценена производительность некоторых подпрограмм математической библиотеки ESSL и в целом вычислительной системы. В работе, в зависимости от поставленной задачи, использовались общепризнанные методы, технологии и про-

Таблица 3. Вычислительные ядра теста STREAM

Название	Операции	Байт/итерацию	Флопс/итерацию
COPY	a(i) = b(i)	16	0
SCALE	a(i) = q*b(i)	16	1
ADD	a(i) = b(i) + c(i)	24	1
TRIAD	a(i) = b(i) + q*c(i)	24	2

Таблица 4. Результаты для теста EP

Ядер (потоков)	SMT	IBM XL			GCC	PGI
		OpenMP	SMPI	OMPI	OpenMP	OpenMP
16(16)	1	71.0	69.5	69.5	31.3	41.3
16(32)	2	94.3	95.4	95.6	52.0	71.6
16(64)	4	118.1	123.1	123.0	70.5	98.7
16(128)	8	141.0	148.1	147.9	77.4	121.0

Таблица 5. Результаты для теста LU

Ядер (потоков)	SMT	IBM XL			GCC	PGI
		OpenMP	SMPI	OMPI	OpenMP	OpenMP
16(16)	1	2758.5	2459.8	2439.1	3089.2	3138.8
16(32)	2	3433.9	3290.7	3234.0	3295.4	3168.4
16(64)	4	3377.0	3585.3	3577.7	3246.6	1933.9
16(128)	8	2744.8	3384.4	3361.1	2690.8	1612.1

граммные средства, краткое описание которых дано в соответствующих разделах.

4. ОЦЕНКА ПРОИЗВОДИТЕЛЬНОСТИ ПОДСИСТЕМ ГИБРИДНОГО ВЫЧИСЛИТЕЛЬНОГО КЛАСТЕРА

4.1. Оценка пропускной способности памяти системы

Процессор POWER8 взаимодействует с памятью с помощью четырехканального контроллера. Каждый из каналов этого контроллера, функционируя на частоте 9.6 ГГц способен считывать 2 байта и записывать 1 байт за раз. Таким образом, теоретическая пропускная способность доступа к памяти для одного сокета составляет 115.2 ГБ/с.

Для оценки реальной пропускной способности подсистемы памяти использовался тест STREAM [9] версии 5.10. Он позволяет оценить устоявшуюся пропускную способность при выполнении операций чтения и записи, выполняющихся совместно с арифметическими операциями. Тест содержит четыре вычислительных ядра, описание которых представлено в таблице 3.

Тестирование выполнялось для различного числа вычислительных потоков, равномерно распределенных по сокетам (на каждом запускалось по половине потоков). При числе потоков, не превышающих 20, использовался режим ST. При числе потоков, равных 40, 80 и 160, использовались режимы SMT2, SMT4 и SMT8 соответственно. Результаты тестов приведены на рис. 2.

Как следует из полученных зависимостей, эффект повышения пропускной способности памяти достигается при последовательном увеличении числа потоков до 8 на узел. При этом для эффективной утилизации четырехканального контроллера памяти CPU может оказаться достаточным как минимум 4 потока. Дальнейшее увеличение числа потоков вплоть до 10 на сокет приводит к незначительному увеличению пропускной способности (на 6–7.5%). При этом максимальная пропускная способность памяти, составляющая 180 ГБ/с (78.3% от пиковой), достигается для теста TRIAD при запуске по одному процессу на ядро (режим ST). При дальнейшем увеличении числа потоков происходит падение пропускной способности памяти, что связано с увеличением числа кэш-конфликтов. Стоит отметить, что скорость падения пропускной способности при использовании режимов SMT2 и SMT4 зависит от объема вычислений, выполняющихся ядрами теста. Так, для тестов COPY, SCALE (10%) и ADD (10.7%) в режиме SMT4 пропускная способность падает не более чем на 11%. При этом для теста TRIAD при переходе к режиму SMT4 пропускная способность снижается на 19.5%. Это связано с возрастанием числа конфликтов за вычислительные ресурсы ядер процессора.

4.2. Оценка эффективности функционирования технологии SMT, производительности компиляторов и технологий параллельного программирования

Оценка эффективности различных режимов функционирования технологии SMT при выполнении параллельных вычислений с использованием технологий OpenMP и MPI, а также для сравнения производительности различных компиляторов и реализаций библиотеки MPI выпол-

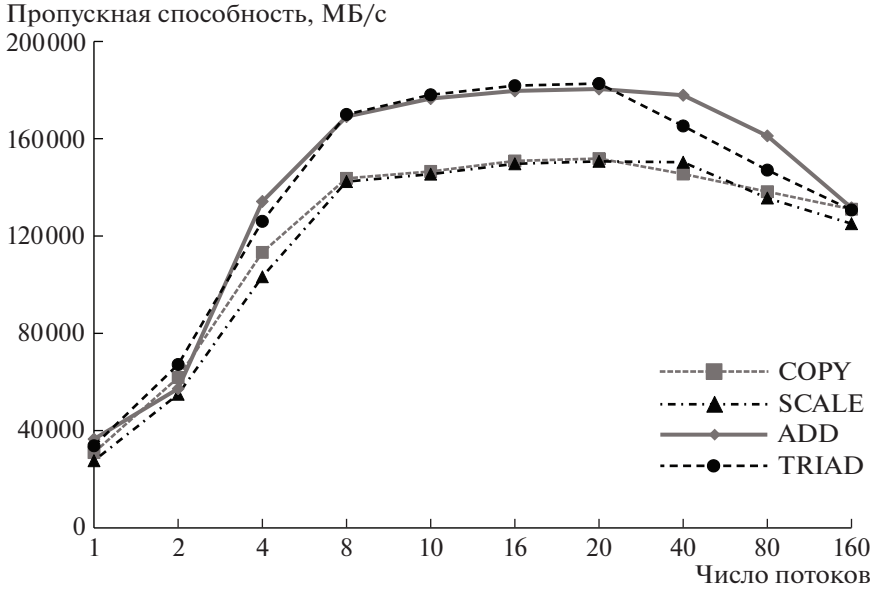

Рис. 2. Результаты теста STREAM.

нялась с использованием теста NAS Parallel Benchmark (NPB) [10]. Тест NPB создан для оценки производительности параллельных вычислений. В настоящей работе использовался тест NPB версии 3.3. Он состоит из ряда простых задач: ядер и приложений. Ядра и приложения могут производить вычисления в классах сложности: S, W, A, B, C, D. С увеличением класса сложности возрастает размерность основных массивов данных и количество итераций в основных циклах программ.

Для оценки эффективности компиляторов и технологии SMT при выполнении параллельных OpenMP и MPI приложений проводились численные эксперименты с использованием тестов EP, LU, MG, CG, FT и IS (класс сложности D). Расчеты выполнялись на 16 процессорных ядрах одного вычислительного узла в режимах ST, SMT2, SMT4 и SMT8. Далее представлены результаты проведенных экспериментов, причем их порядок выстроен в соответствии с уровнем нагрузки на сеть передачи данных. Показатели производительности усреднялись по результатам 5 испытаний.

Тест “EP” служит для оценки производительности в расчетах с плавающей запятой при отсутствии заметных межпроцессорных взаимодействий. Он включает в себя генерацию псевдослучайных нормально распределенных чисел. В табл. 4 показана достигнутая производительность в MOPS (миллион операций в секунду) в расчете на процессорное ядро, в скобках указано суммарное число вычислительных потоков (процессов для MPI). Можно отметить, что применение технологии SMT положительно сказывается на производительность данного теста. Так, при увеличении числа вычислительных процессов с одного до 8 на ядро, производительность удваивается для всех исследуемых компиляторов и технологий параллельного программирования. При этом использование компилятора IBM XL позволило получить в два раза большую производительность по сравнению с компиляторами GCC и PGI. В пределах одного узла технология OpenMP показывает одинаковую производительность с MPI.

В тесте “LU” проводится LU разложение. В табл. 5 показаны полученные результаты оценки производительности. Можно видеть, что оптимальным SMT режимом для данного теста является режим SMT2, при котором на каждом процессорном ядре выполняется по 2 потока, а дальнейшее увеличение их числа приводит к снижению производительности. При этом максимальную производительность в данном режиме демонстрирует версия, собранная с использованием компилятора IBM XL. В режиме ST оптимальным является компилятор PGI.

В тесте “MG” находится приближенное решение трехмерного уравнения Пуассона с периодическими граничными условиями. Используется многосеточный алгоритм. В табл. 6 показаны результаты проведенных экспериментов. Можно видеть, что использование технологии SMT не приводит к повышению производительности данного теста и оптимальным режимом использования процессорных ядер является режим ST. При этом все компиляторы и технологии параллельного программирования показывают практически одинаковую производительность.

Таблица 6. Результаты для теста MG

Ядер (потоков)	SMT	IBM XL			GCC	PGI
		OpenMP	SMPI	OMPI	OpenMP	OpenMP
16(16)	1	2624.3	2581.4	2595.0	2539.3	2539.6
16(32)	2	2598.1	2271.4	2279.3	2521.0	2487.9
16(64)	4	2523.5	2431.1	2484.1	2542.2	2378.7
16(128)	8	2355.0	1819.5	1920.1	2522.5	2194.5

Таблица 7. Результаты для теста CG

Ядер (потоков)	SMT	IBM XL			GCC	PGI
		OpenMP	SMPI	OMPI	OpenMP	OpenMP
16(16)	1	125.4	553.0	554.6	126.4	123.9
16(32)	2	150.1	603.5	602.0	152.2	153.4
16(64)	4	152.2	362.3	372.0	154.0	153.2
16(128)	8	153.1	518.0	527.0	152.8	153.4

В тесте “CG” решается СЛАУ с разряженной произвольной матрицей методом сопряженных градиентов. Коммутации в MPI реализации организованы с помощью неблокирующих двухточечных взаимодействий. Из табл. 7 видно, что максимальная производительность в данном тесте достигается при использовании режима SMT2. При этом все компиляторы демонстрируют практически одинаковую производительность. Использование технологии MPI позволяет ускорить вычисления в 4 раза по сравнению с технологией OpenMP. Это связано с тем, что массивы данных в OpenMP версии теста не помещаются в кэш [11].

В тесте “FT” решается 3-D задача с использованием дискретного преобразования Фурье. Взаимодействие между процессами MPI версии осуществляется с помощью следующих коллективных операций: MPI_Reduce, MPI_Barrier, MPI_Bcast, MPI_Alltoall. Данные, приведенные в табл. 8, показывают, что в пределах одного узла технология OpenMP при использовании компилятора IBM XL показывает производительность в два раза выше, чем технология MPI или OpenMP с остальными исследуемыми компиляторами. Оптимальным режимом функционирования процессора является режим SMT2. При дальнейшем увеличении числа вычислительных потоков на ядро производительность резко падает.

В тесте “IS” осуществляется параллельная сортировка большого массива целых чисел, см. табл. 9. Передача сообщений между процессами MPI версии осуществляется с помощью операций MPI_Alltoall и MPI_Allreduce. Результаты экспериментов показывают, что в пределах одного узла технология MPI показывает производительность в 5 раз выше, чем OpenMP. При этом оптимальным режимом работы процессора для данной технологии является режим ST, а для технологии OpenMP – режим SMT4. Все исследуемые компиляторы показывают приблизительно одинаковый уровень производительности.

Исходя из полученных результатов можно сделать вывод о том, что технология SMT позволяет повысить утилизацию ядер центрального процессора при выполнении не оптимизированных приложений, производительность которых ограничена скоростью выполнения вычислительных операций. Если же производительность приложения ограничена пропускной способностью памяти, то при использовании технологии SMT может

Таблица 8. Результаты для теста FT

Ядер (потоков)	SMT	IBM XL			GCC	PGI
		OpenMP	SMPI	OMPI	OpenMP	OpenMP
16(16)	1	4064.8	1890.6	2639.0	2560.1	2172.7
16(32)	2	4195.5	2162.1	3042.1	2913.8	2317.8
16(64)	4	2250.1	2171.9	2962.4	2105.4	1903.8
16(128)	8	1086.3	1418.9	1694.2	1071.1	1096.6

Таблица 9. Результаты для теста IS

Cores (threads)	SMT	IBM XL			GCC	PGI
		OpenMP	SMPI	OMPI	OpenMP	OpenMP
16(16)	1	27.8	111.6	124.5	25.9	23.3
16(32)	2	33.3	89.9	95.3	32.6	31.9
16(64)	4	40.4	73.4	79.5	40.2	37.6
16(128)	8	34.9	48.2	57.7	34.1	31.9

наблюдаться снижение производительности из-за увеличения конфликтов доступа к памяти. При этом среди всех рассмотренных компиляторов во всех тестах наибольшую производительность показал код, сгенерированный компилятором IBM XL. Приложения, разработанные с использованием технологий OpenMP и MPI, в большинстве случаев демонстрируют схожий уровень производительности. Прирост производительности от использования технологии OpenMP может наблюдаться в тех случаях, когда требуется реализация интенсивного взаимодействия между вычислительными потоками.

4.3. Сравнение производительности реализаций библиотеки MPI

Для оценки производительности реализаций библиотеки MPI проводились численные эксперименты с использованием тестов EP и IS. Первый из них, как уже упоминалось выше, является примером приложений, в которых практически отсутствует передача сообщений между вычислительными процессами. Второй тест напротив характеризуется активным межпроцессным взаимодействием с использованием коллективных MPI операций. Оба теста запускались с использованием от 4 до 64 вычислительных процессов (1–4 вычислительных узла) в режиме ST. В качестве среды передачи сообщений в пределах одного узла использовалась оперативная память, между узлами – сеть InfiniBand (интерфейс PAMI для Spectrum MPI, vader и openib для Open MPI). По их результатам были построены графики ускорения S, приведенные на рисунке 3. Ускорение рассчитывалось по формуле $S_N = R_N/R_4$, где R_N – производительность, полученная при выполнении теста на N процессорных ядрах и R_4 – производительность, полученная на четырех процессорных ядрах.

Из рисунка видно, что для теста EP наблюдается практически линейный рост производительности при увеличении числа вычислительных процессов, в то время как для теста IS достигаемое ускорение с ростом числа используемых узлов быстро уменьшается. Это вызвано повышенной нагрузкой на коммуникационную сеть при выполнении большого числа операций MPI_Alltoall и MPI_Allreduce, характерных для теста IS. При этом можно отметить, что библиотека Spectrum MPI позволяет получить большую производительность передачи сообщений между вычислительными узлами, что приводит к лучшей масштабируемости параллельных приложений. В пределах одного вычислительного узла производительность передачи сообщений одинакова для Spectrum MPI и OpenMPI, что подтверждается практически одинаковым ускорением, достигаемым тестами при увеличении числа вычислительных процессов с 4 до 16.

4.4. Оценка общей производительности вычислительного кластера

Для оценки общей производительности вычислительного кластера использовался пакет HPL (A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers) [12] с двумя программными реализациями:

- версия 2.1, оптимизированная для работы на гибридных вычислительных системах, оснащенных графическими сопроцессорами производства компании NVIDIA;
- версия 2.2, собранная с использованием библиотеки IBM Engineering and Scientific Subroutine Library (ESSL) версии 5.5 [13], содержащей оптимизированные подпрограммы для операций линейной алгебры, решения СЛАУ, проведения анализа на собственные значения матриц, выполнения быстрого преобразования Фурье, сортировки, поиска, интерполяции, генерации псевдослучайных чисел и т.д.

Для обеспечения совместимости с проприетарными и свободными реализациями математических библиотек операции линейной алгебры реализованы в виде подпрограмм BLAS (Basic Linear Algebra Subprograms), что позволяет использовать ее с пакетом HPL. Все подпрограммы BLAS уровня 3 библиотеки ESSL поддерживают автоматическую выгрузку вычислений на графические сопроцессоры. На рис. 4 представлены графики зависимости производительности R подпрограммы DGEMM от размерности матрицы N для различных режимов функционирования рассмат-

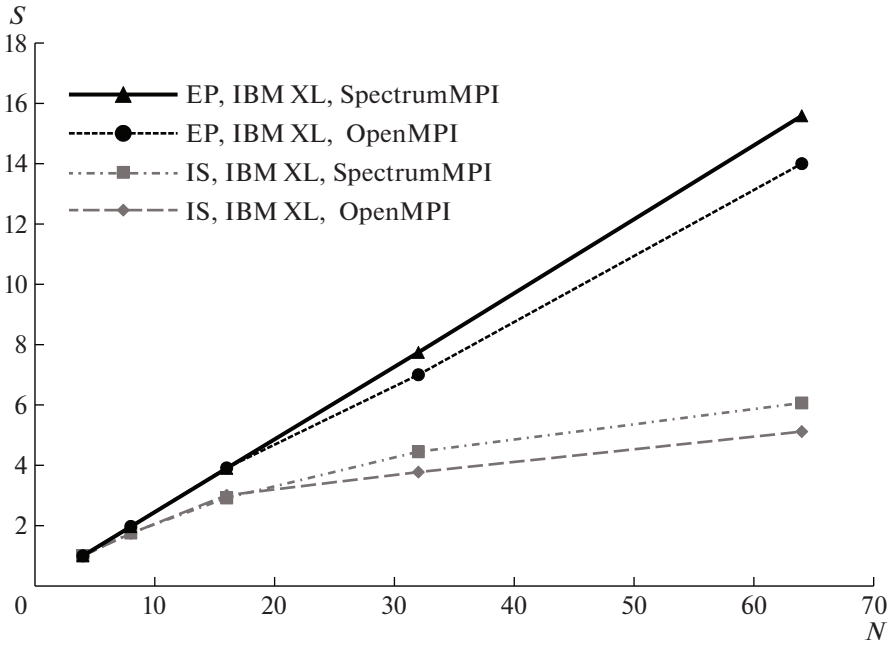

Рис. 3 Зависимость ускорения S от числа процессорных ядер N для тестов EP и IS.

риваемой библиотеки. Вычисления проводились с использованием бенчмарка Crossroads/NERSC-9 DGEMM [14] для одного NUMA узла, включающего 1 процессорный сокет и 1 GPU. Из рисунка видно, что при $N > 2000$ производительность версии подпрограммы, выполняющейся на GPU (SMP CUDA), становится больше производительности параллельной версии подпрограммы (SMP), выполняющейся на 10 процессорных ядрах. При этом для матрицы размерностью 20000 производительность SMP CUDA версии библиотеки более чем в 6 раз выше, чем производительность SMP версии – 3.1 ТФлопс и 0.5 ТФлопс соответственно. Для оценки возможности ускорения приложений путем переноса части вычислений на графические сопроцессоры в версии HPL 2.2 использовалась SMP CUDA версия библиотеки ESSL. Исходный код обоих версий пакета собирался с использованием компилятора IBM XL и библиотеки Spectrum MPI (версии ПО приведены выше). На узлах был установлен драйвер NVIDIA версии 361.119.

Программа HPL решает СЛАУ $Ax = b$ методом LU-разложения с выбором ведущего элемента, где A – плотно заполненная вещественная матрица двойной точности размерности N. Данный тест позволяет оценить реальную производительность вычислительной системы.

При проведении тестирования нужно учитывать, что величина получаемых результатов сильно зависит от заданных параметров теста, поэтому требуется его настройка на каждой вычислительной системе. На рассматриваемом вычислительном кластере максимальное значение производительности для первой версии пакета было получено при использовании следующих параметров теста: $N = 380160$; NB = 768; P × Q = 5 × 2; алгоритм передачи (BCAST) – 1ringM; на каждых 10 процессорах запускалось по 1 MPI процессу, к каждому из которых привязывался один графический ускоритель. Оно составило 40.39 ТФлопс или 72% от пиковой. Для второй версии максимальная производительность в 27.41 ТФлопс (49% от пиковой) была достигнута при следующих настройках: $N = 350000$; NB = 1856; P × Q = 10 × 10; алгоритм передачи (BCAST) – 1ring; на каждом процессорном ядре запускалось по 1 MPI процессу; к процессам, запущенным на сокете 0 привязывался сопроцессор 0, а к процессам, запущенным на сокете 1 – сопроцессор 1.

Из результатов проведенных тестов видно, что максимальная производительность в тесте HPL может быть достигнута с использованием низкоуровневых оптимизаций исходного кода данного приложения, выполненных с использованием технологий NVIDIA CUDA, OpenCL, OpenACC и т.п. При этом автоматическая выгрузка части вычислений на графические сопроцессоры с использованием оптимизированных библиотек позволяет ускорять выполнение некоторых приложений без модификации их исходного кода. Данная возможность реализована в некоторых математических библиотеках, например, в библиотеке IBM ESSL.

5. Заключение. В статье проведена комплексная оценка производительности гибридного вы-

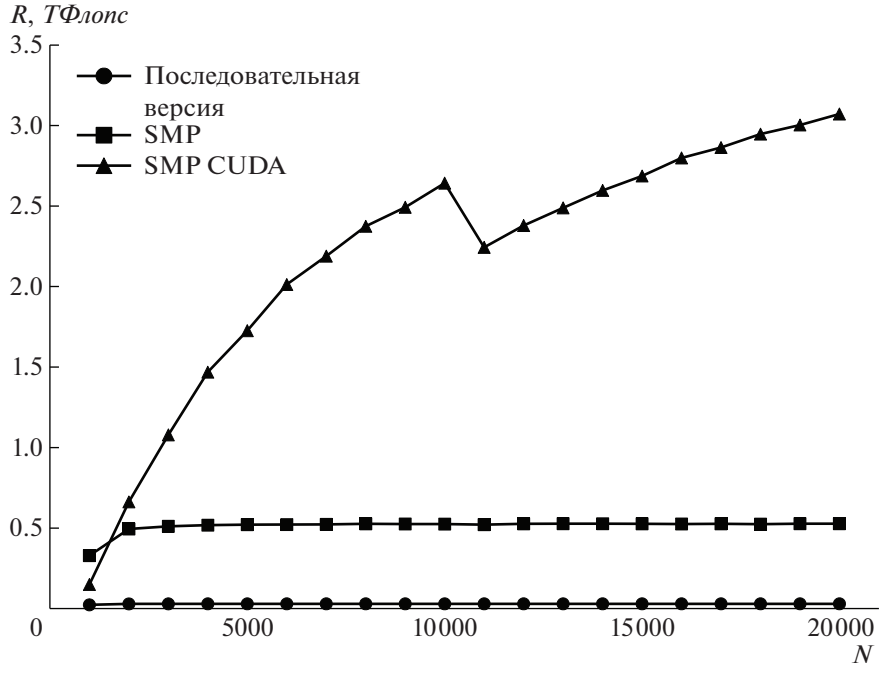

Рис. 4 Зависимость производительности подпрограммы DGEMM библиотеки ESSL от размера матрицы.

числительного кластера, построенного на процессорах IBM POWER8 и ускорителях NVIDIA Tesla P100 GPU. По результатам исследований можно сделать вывод о допустимости применения подобных систем не только для решения задач в области машинного обучения, глубокого обучения и искусственного интеллекта, но и для проведения вычислений с использованием различных прикладных программных средств. Их высокая скорость работы может быть достигнута за счет архитектурных особенностей центральных процессоров (в частности, технологии SMT), а также выгрузки части вычислений на сопроцессоры, что становится возможным благодаря появлению специализированных библиотек (в данном случае, IBM ESSL), которые позволяют ускорять выполнение приложений без модификации их исходного кода. По мере развития этих технологий и систем управления ресурсами гибридных вычислительных систем их можно рассматривать как основу для организации универсальной и эффективной инфраструктуры высокопроизводительных вычислений.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 18-29-03196. При проведении численных расчетов было использовано оборудование Центра коллективного пользования “Центр данных ДВО РАН” (ВЦ ДВО РАН, г. Хабаровск) [15] и Федерального исследовательского центра “Информатика и управление РАН” (г. Москва).

СПИСОК ЛИТЕРАТУРЫ

1. *Shan A.* 2006. Heterogeneous Processing: a Strategy for Augmenting Moore’s Law. https://www.linuxjournal.com/article/8368.
2. TOP500 Supercomputer Sites. 2019. https://www.top500.org.
3. *Karkhanis T.S., Moreira J.E.* IBM Power Architecture. In: *Padua D.* (eds) Encyclopedia of Parallel Computing. Springer, Boston, MA. 2011. P. 2175.
4. *Sinharoy B., Van Norstrand J.A., Eickemeyer R.J., Le H.Q., Leenstra J., Nguyen D.Q., Konigsburg B., Ward K., Brown M.D., Moreira J.E., Levitan D., Tung S., Hrusecky D., Bishop J.W., Gschwind M., Boersma M., Kroener M., Kaltenbach M., Karkhanis T., Fernsler K.M.* IBM POWER8 processor core microarchitecture // IBM Journal of Research and Development. 2015. V. 59. № 1. P. 2:1–2:21.
5. *Eggers S.J., Emer J.S., Levy H.M., Lo J.L., Stamm R.L., Tullsen D.M.* Simultaneous multithreading: a platform for next-generation processors // IEEE Micro. 1997. V. 17. № 5. P. 12–19.
6. *Starke W.J., Stuecheli J., Daly D.M., Dodson J.S., Auernhammer F., Sagmeister P.M., Guthrie G.L., Marino C.F., Siegel M., Blaner B.* The cache and memory subsystems of the IBM POWER8 processor // IBM Journal of Research and Development. 2015. V. 59. № 1. P. 3:1–3:13.
7. NVIDIA Tesla P100: The Most Advanced Datacenter Accelerator Ever Built. Featuring Pascal GP100, the World’s Fastest GPU. Whitepaper, 2016. 45 p.
8. Multi-Process Service. NVIDIA. 2015. P. 27.
9. *McCalpin J.D.* Memory Bandwidth and Machine Balance in Current High Performance Computers. IEEE Computer Society Technical Committee on Computer Architecture (TCCA) Newsletter, December 1995.

10. *Bailey D., Barszcz E., Barton J., Browning D., Carter R., Dagum L., Fatoohi R., Fineberg S., Frederickson P., Lasinski T., Schreiber R., Simon H., Venkatakrishnan V., Weeratunga S.* The NAS Parallel Benchmarks. RNR Technical Report RNR 94-007, March, 1994.

11. *Saini S., Chang J., Hood R., Jin H.* A scalability Study of Columbia using the NAS Parallel Benchmarks // Computational Methods in Science and Technology, Special Issue (1). 2006. P. 33–45.

12. *Dongarra J. J., Luszczek P., Petite A.* The LINPACK benchmark: Past, present and future // Concurrency Computation Practice and Experience. 2003. V. 15. № 9. P. 803–820.

13. ESSL Guide and Reference. IBM. 2016. P. 1436.

14. *Austin B., Wright N.J.* Measurement and Interpretation of Micro-benchmark and Application Energy Use on the Cray XC30 // 2014 Energy Efficient Supercomputing Workshop. 2014. P. 51–59.

15. *Sorokin A.A., Makogonov S.I., Korolev S.P.* The Information Infrastructure for Collective Scientific Work in the Far East of Russia // Scientific and Technical Information Processing. 2017. V. 4. P. 302–304.