Экспериментальная оценка результатов внедрения технологии NVIDIA GPUDirect на суперкомпьютере НИУ ВШЭ*

Р.А. Чулкевич, В.И. Козырев, П.С. Костенецкий, А.А. Раимова

Национальный исследовательский университет «Высшая школа экономики»

Оптимизация использования вычислительных ресурсов на высокопроизводительных кластерах является важной задачей в условиях высокой загрузки. Одним из способов такой оптимизации является применение современных технологий. В то же время, на разных серверных архитектурах поведение технологий может отличаться. В частности, влияние оказывает то, как именно осуществляется взаимодействие компонентов аппаратной архитектуры (например, между GPU и InfiniBand адаптером). В данной статье анализируется применение технологий NVIDIA GPUDirect RDMA и NVIDIA GPUDirect Copy на различных архитектурах вычислительных узлов суперкомпьютерного комплекса cHARISMa. Рассматривается изменение задержки и скорости передачи данных между GPU на разных вычислительных узлах при различных комбинациях задействованных технологий. В лучших случаях задержка при передаче данных уменьшилась в 7.8 раза, а увеличение пропускной способности составило до 286%. Полученные результаты показывают, что применение технологий GPUDirect Copy и GPUDirect RDMA с учетом аппаратной архитектуры может значительно ускорять выполнение задач, как использующих частые обмены с памятью GPU в рамках одного узла, так и выполняющих обмены между GPU на нескольких вычислительных узлах.

Ключевые слова: GPUDirect RDMA, GDR, GPUDirect Copy, GDRC, вычислительный кластер, NUMA, InfiniBand.

1. Введение

С увеличением числа пользователей и научных проектов, выполняемых на суперкомпьютерном комплексе *cHARISMa* НИУ ВШЭ, нагрузка на его вычислительные ресурсы существенно возросла. Возникла потребность в увеличении эффективности использования имеющихся вычислительных ресурсов, путем применения новых программных инструментов и оптимизаций. Одним из таких инструментов стала разработанная в НИУ ВШЭ система HPC TaskMaster, позволяющая автоматически находить и отменять некорректно запущенные и неэффективные задачи пользователей [6]. Другим инструментом для увеличения эффективности суперкомпьютера стала оптимизация системного программного обеспечения. Так обновление программного обеспечения системы хранения данных суперкомпьютера позволило повысить скорость записи данных на 32.8% [1], что значительно ускорило задачи, активно использующие файловое хранилище.

Данная работа посвящена новому этапу оптимизации программной конфигурации суперкомпьютерного комплекса *cHARISMa*, ускорению выполнения вычислительных задач, использующих графические процессоры, путем изменения процесса обмена данными с памятью GPU. Весьма важным и значимым в этой области является семейство технологий *NVIDIA GPUDirect*.

Технологии *NVIDIA GPUDirect* внедряются на многих суперкомпьютерных комплексах [3, 9, 12]. Например, в статье [3] приводятся данные о латентности в типовых задачах машинного обучения и анализируются преимущества внедрения аппаратного ускорения передачи данных. Авторы получили результат, что GPUDirect RDMA может сократить задержки передачи данных от 15 до 50%, что существенно ускоряет задачи машинного обучения.

* Исследование выполнено с использованием суперкомпьютерного комплекса НИУ ВШЭ [5].

Эффективность технологий *GPUDirect* напрямую связана с аппаратной архитектурой вычислительных узлов кластера, поэтому важно сравнивать производительность до и после внедрения технологий и использовать их в случае получения положительного эффекта. В этой статье рассказывается о результатах применения на всех типах вычислительных узлов суперкомпьютера НИУ ВШЭ двух технологий *GPUDirect Remote Direct Memory Access* и *GPUDirect Copy*.

Технология *GPUDirect Remote Direct Memory Access* (далее – *GDR*) предназначена для исключения центрального процессора и оперативной памяти из процесса обмена данных между GPU на разных вычислительных узлах. GDR основана на возможности GPU-ускорителей соотносить (*expose*) часть памяти GPU с областью памяти PCIe-устройства, известной также как базовые адресные регистры *BAR (Base Address Register*). В этом случае PCIe-устройства получают прямой доступ к памяти GPU через Peer-to-Peer-соединение, исключая необходимость в использовании CPU и промежуточных буферов оперативной памяти.

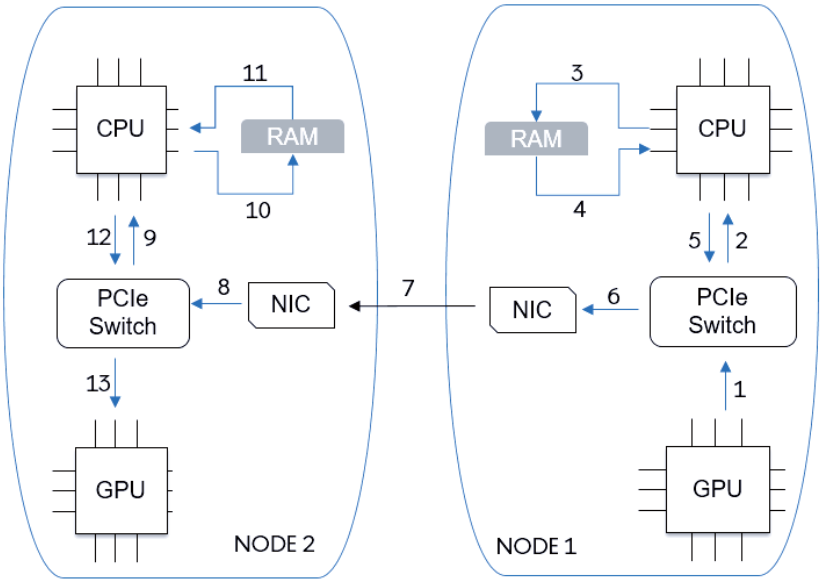

Рис. 1. Обмен данными между GPU без применения GDR

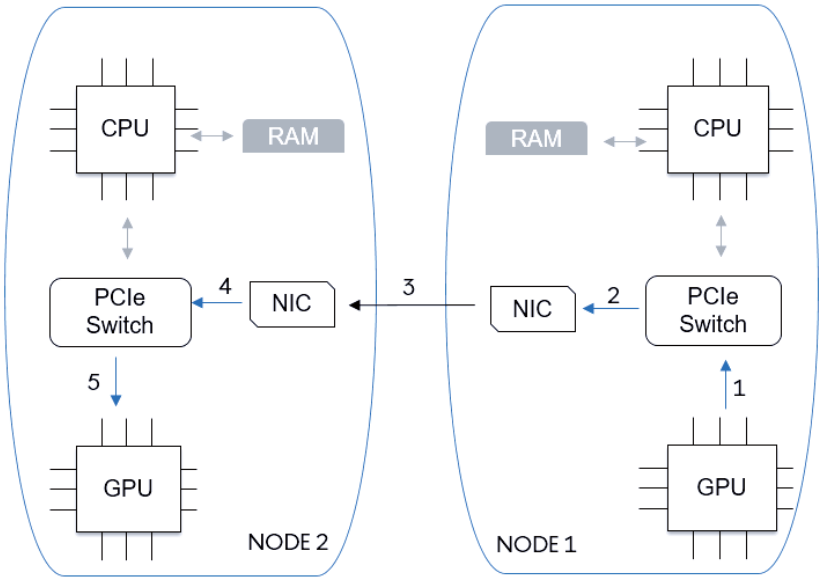

Рис. 2. Обмен данными между GPU с применением GDR

На рис. 1 и рис. 2 показано сравнение процесса передачи данных между двумя GPU, расположенными на разных вычислительных узлах, в стандартном варианте и с применением

технологии. Без GDR данные сначала попадают в оперативную память с использованием CPU, затем передаются через сетевой адаптер в оперативную память другого вычислительного узла и в конечном итоге размещаются в оперативной памяти графического ускорителя. При использовании GDR процесс обмена данными выполняется через сетевые адаптеры напрямую, уменьшая задержки и увеличивая скорость передачи. Если в обычном режиме передача выполняется за 13 этапов, то при использовании GDR, количество этапов сокращается до 5.

Если основной целью GDR является реализация прямого доступа в память GPU с различных устройств, то технология *GPUDirect Copy* (далее – *GDRC*) предназначена для создания сопоставлений памяти GPU с CPU (*CPU Mapping of the GPU Memory*). В результате, в пространстве пользователя создается сопоставление (mapping) памяти GPU, которое может быть использовано, как если бы это была обычная оперативная память вычислительного узла. Это позволяет снизить задержки и накладные расходы (overhead) при передаче данных из/в GPU.

На рис. 3. показано сравнение использования стандартного механизма работы с памятью через cudaMemcpy и механизмов из библиотеки GDRCopy. CudaMemcpy использует DMA Engine API для перемещения данных между CPU и памятью GPU [2]. Это приводит к задержкам и накладным расходам при передаче при передаче данных. GDRCopy позволяет CPU напрямую обращаться к памяти GPU через сопоставления BAR (Base Address Registers), значительно уменьшая задержку при передаче.

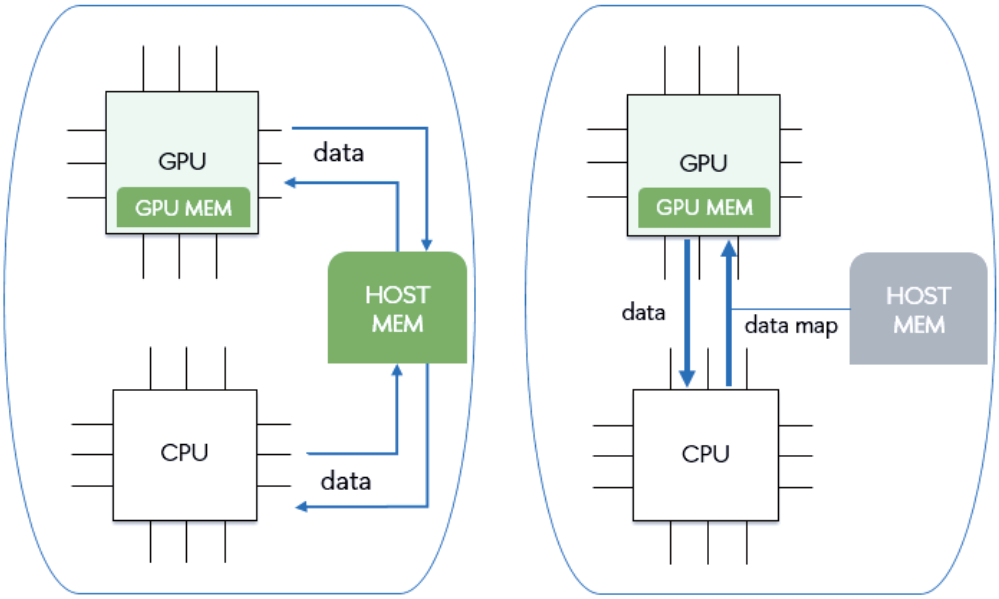

Рис. 3. Разница обмена данными с памятью GPU:
слева – по умолчанию cudaMemcpy, справа – с применением GDRC

Поддержка GDR и GDRC реализована во всех современных коммуникационных библиотеках (OpenMPI, MVAPICH2, NVIDIA HPC-X и др.). Это позволяет задействовать новые технологии во многих вычислительных задачах, путем подключения обновленных библиотек.

2. Архитектура вычислительных узлов

В суперкомпьютере НИУ ВШЭ *cHARISMa* есть три типа вычислительных узлов с GPU, обладающих различной аппаратной архитектурой.

1. *Tun A/B*: Dell PowerEdge C4140K, 2 x Intel Xeon Gold 6152, 768/1536 ГБ ОЗУ, 4 x NVIDIA Tesla V100 32 ГБ NVLink, 2 x InfiniBand EDR – см. Рис 4.
2. *Tun C*: Dell PowerEdge C4140M, 2 x Intel Xeon Gold 6240R, 768 ГБ ОЗУ, 4 x NVIDIA Tesla V100 32 ГБ NVLink, 2 x InfiniBand EDR – см. Рис 4.
3. *Tun E*: HPE XL675d Gen10+, 2 x AMD EPYC 7702, 1024 ГБ ОЗУ, 8 x NVIDIA Tesla A100 80 ГБ NVLink, 2 x InfiniBand EDR – см. Рис 5.

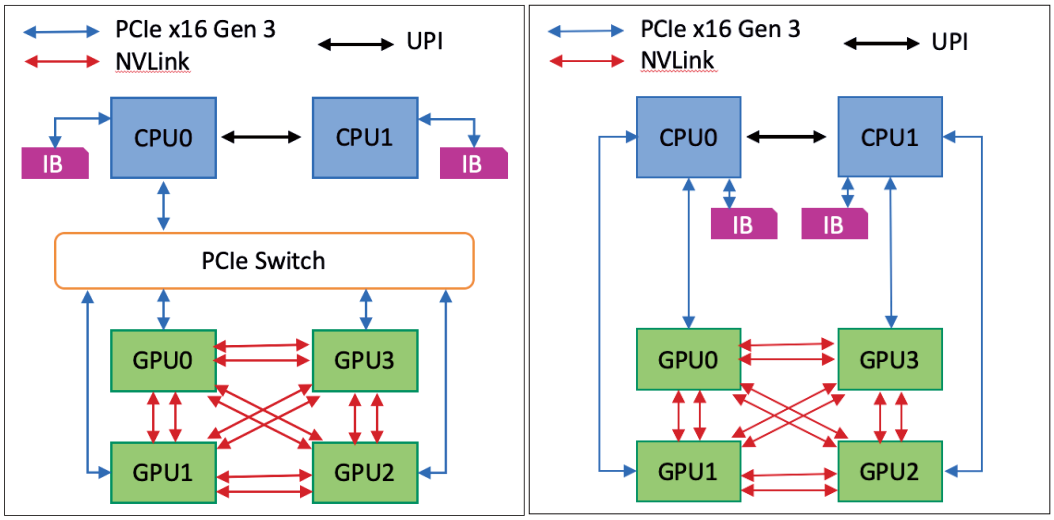

Рис. 4. Аппаратная архитектура вычислительных узлов типов A, B (слева) и C (справа)

На вычислительных узлах типов A/B и C доступ GPU-ускорителей к InfiniBand адаптеру производится через PCIe-мост и центральный процессор. Дополнительно, на вычислительных узлах типа A и B, доступ к модулю GPU-ускорителей с хоста осуществляется через внутренний PCIe-коммутатор, что усложняет процесс передачи данных и немного уменьшает производительность в пиковых режимах.

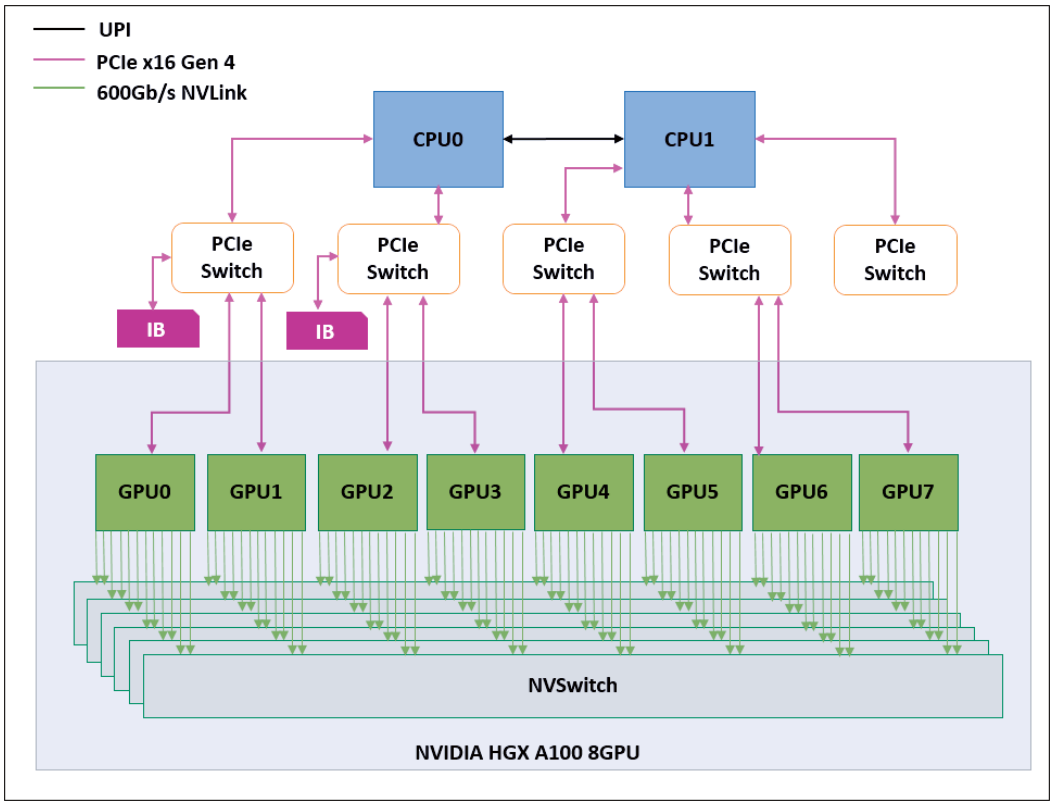

Рис. 5. Аппаратная архитектура вычислительных узлов типа E

Вычислительные узлы типа E сконструированы по подобию «архитектурного эталона» от NVIDIA, который был реализован в NVIDIA DGX A100. Здесь все GPU ускорители размещены на отдельной плате NVIDIA HGX и объединены скоростными внутренними коммутациями NVSwitch. В данной архитектуре доступ GPU к InfiniBand-адаптеру и NVMe-диску может происходить напрямую через PCIe-коммутатор, не требуя предварительной передачи данных в оперативную память узла.

В каждом вычислительном узле, независимо от его типа, установлено два сетевых адаптера InfiniBand EDR 100 Гб/с, подключенных к разным сетевым коммутаторам Mellanox.

Конфигурация вычислительных узлов сильно влияет на эффективность использования технологий GDR и GDRC. Скорость передачи данных при использовании GPUDirect RDMA также снижается при наличии архитектурных особенностей PCIe и NUMA-эффектов [2]. Стоит также отметить, что модуль ядра GDRC может быть задействован в системном ПО UCX, необходимом для эффективной работы более крупного системного ПО – OpenMPI (GPU-Aware), которое, в свою очередь, является фундаментом для сборок многих прикладных пакетов и может оказывать влияние на их производительность при расчетах.

3. Конфигурация программного обеспечения

Базовая операционная система кластера *cHARISMa* – Linux Centos 7.9 [1].

В процессе внедрения технологий NVIDIA GPUDirect была выполнена сборка и установка следующего системного программного обеспечения:

1) CUDA v12;
2) библиотека для высокоскоростных и низко-латентных сетей UCX v1.14.1 с параметрами *--with-cuda* и *--with-gdrcopy*;
3) библиотека для выполнения глобальных операций UCC v1.2.0 с параметрами *--with-cuda* и *--with-ucx*;
4) OpenMPI v4.1.5 с параметрами *--with-cuda, --with-ucx* и *--with-ucc*.

4. Тестирование

В рамках данного исследования на всех типах вычислительных узлов замерялись две величины: задержка и пропускная способность при двунаправленной передаче данных из GPU одного узла в GPU другого узла. Замеры выполнялись с использованием набора тестов OSU Micro-Benchmarks v7.0.1 [7], собранного с указанным выше набором ПО.

Тестирование задержки при передаче данных производилось по принципу «пинг-понг». Узел-отправитель посылает сообщение определенного размера и ожидает ответа от узла-получателя. Тот, в свою очередь, получает сообщение и отправляет его обратно с тем же объемом данных. В процессе тестирования проводится несколько итераций и в качестве итогового результата выбирается средняя величина односторонней задержки.

Тестирование пропускной способности осуществлялось путем отправки узлом-отправителем фиксированного количества сообщений одного размера, следующих друг за другом. Узел-получатель отправлял ответ только после получения всех этих сообщений. Производилось несколько итераций этого процесса, и пропускная способность рассчитывалась на основе прошедшего времени (с момента отправки первого сообщения до момента получения ответа от узла-получателя) и количества байт, отправленных узлом-отправителем. Затем узлы менялись ролями, и отправка сообщений производится в обратную сторону. Итоговый результат является суммарной пропускной способностью в две стороны.

Для чистоты экспериментов, тестирование проводилось во время профилактических работ, когда на суперкомпьютере отсутствовала какая-либо посторонняя вычислительная нагрузка. На выбранных вычислительных узлах были использованы одни и те же ядра центрального процессора, области оперативной памяти, GPU-ускоритель и сетевой адаптер, объединенные в один *узел NUMA* (*NUMA node*). Известно, что влияние *NUMA* (*Non-Uniform Memory Access*) может быть существенно [4, 11]. Корректная привязка MPI-процессов к правильному узлу NUMA во время тестирования обеспечивалась с помощью утилиты *numactl* [8].

При выполнении тестирования учитывался тот факт, что GDRCopy в отношении задержек на передачу данных эффективна для небольших сообщений [10] – опытным путем было установлено, что независимо от типа узла, для cHARISMa это размер до 8КБ: именно на этом размере задержка превышает стандартную (без применения технологий), а затем GDRCopy просто не оказывает эффекта. Для GPUDirect RDMA наоборот более интересна пропускная способность, оценка которой не столь значима на пакетах с маленьким размером. Поэтому для наглядности мы приводим графики задержки для сообщений с размером до 8КБ, а графики

пропускной способности – свыше 8КБ. Результаты тестирования для каждого типа GPU-узлов суперкомпьютера *cHARISMa* приведены на рис. 6–11.

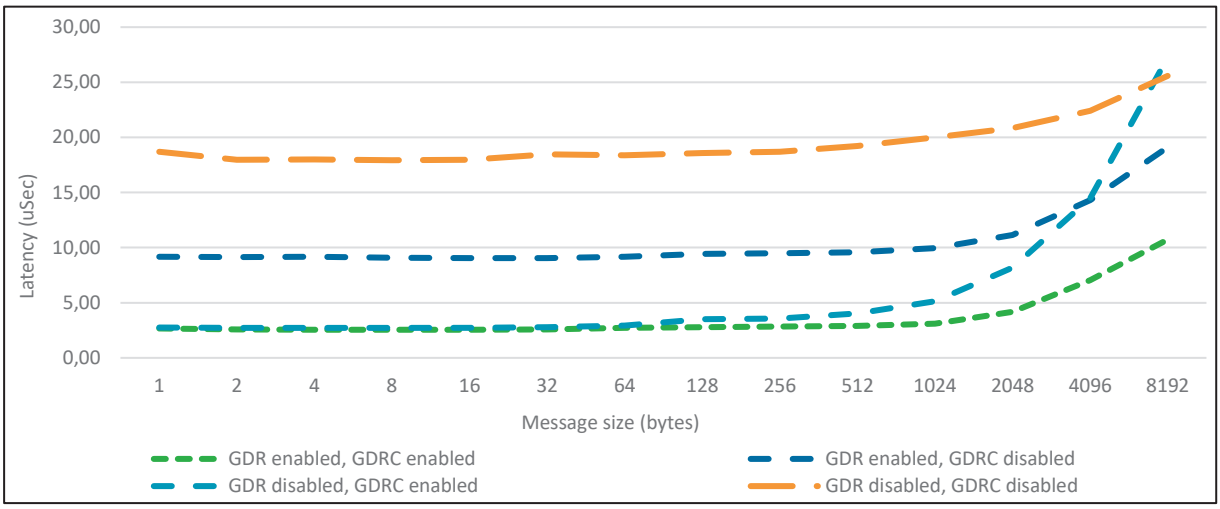

Рис. 6. Задержка передачи на узлах типа A/B

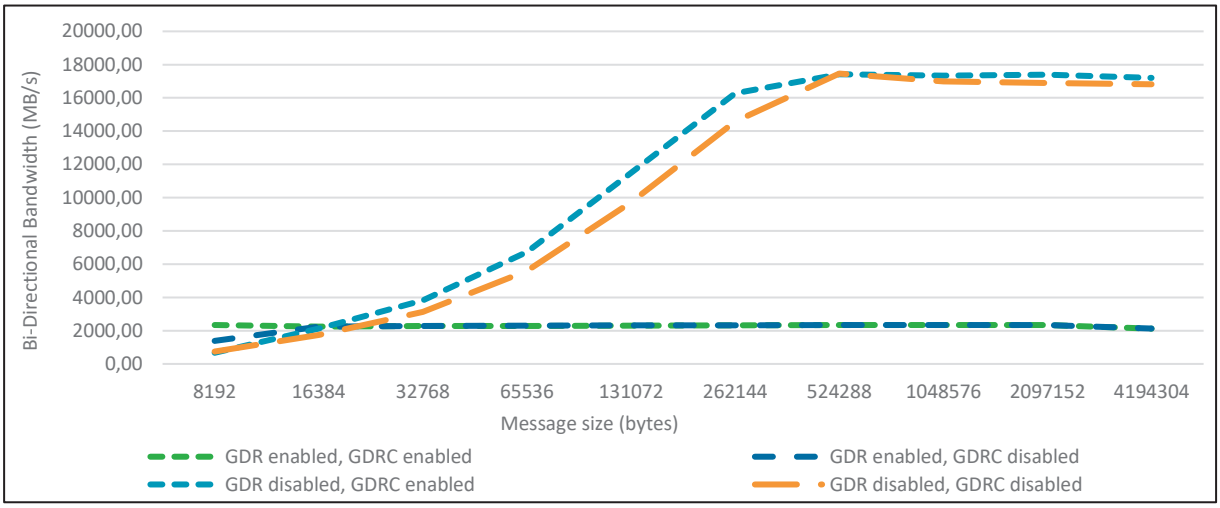

Рис. 7. Пропускная способность на узлах типа A/B

На узлах типа A/B задержка при передаче данных уменьшилась в 7 раз. Наибольший прирост эффективности этого показателя заметен при передаче небольших пакетов данных, размером до 32 байт. Включение GDR на таких узлах дает прирост пропускной способности на пакетах данных, размером до 16 КБ, но при увеличении размера пакета GDR значительно замедляет передачу данных.

На вычислительных узлах типа A/B и C заметно влияние различающейся аппаратной архитектуры и, конкретно, необходимость взаимодействия с PCIe-мостом при передаче данных (см. рис. 4). Так на узлах типа C задержка при передаче данных изначально ниже из-за более эффективной аппаратной архитектуры. После включения GDR и GDRC задержка уменьшилась в 5.8 раз. Наибольшее уменьшение задержки видно на небольших пакетах данных, размером до 32 байт. В то же время, пропускная способность существенно ускоряется при размере пакетов от 64 КБ.

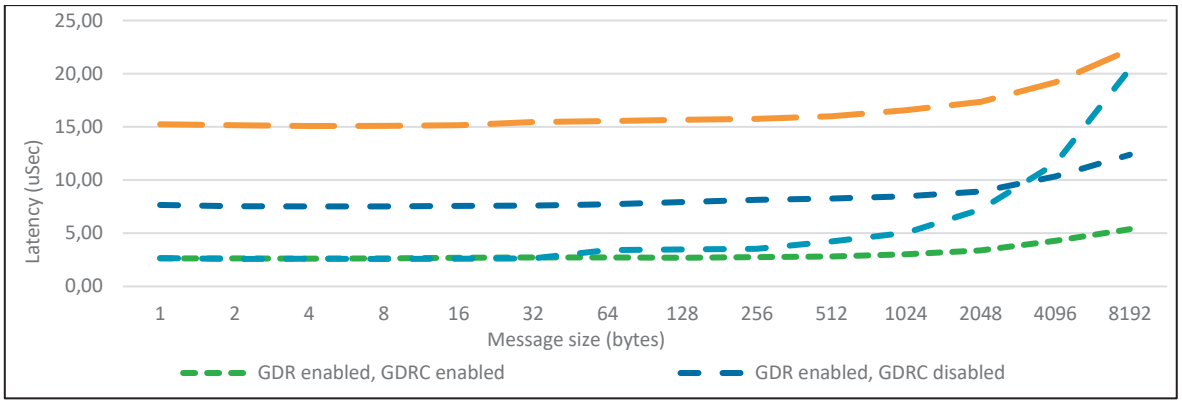

Рис. 8. Задержка передачи на узлах типа C

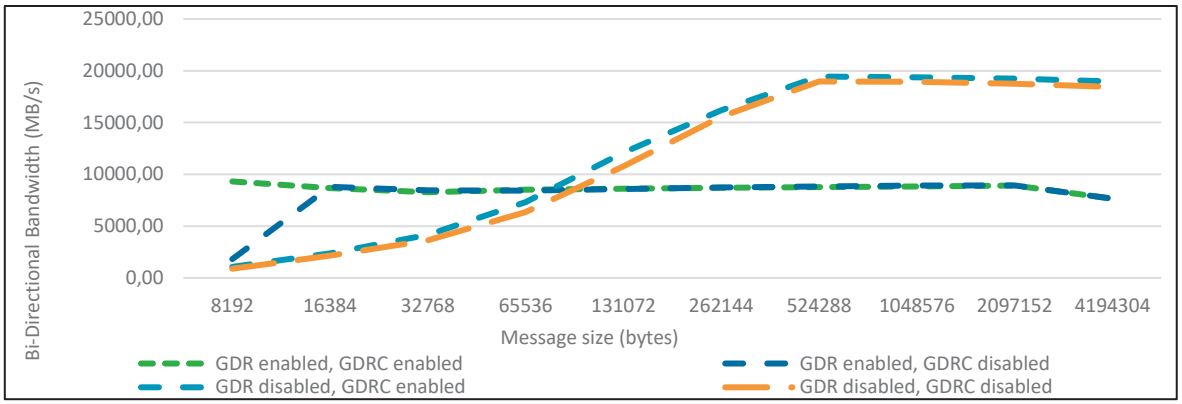

Рис. 9. Пропускная способность на узлах типа C

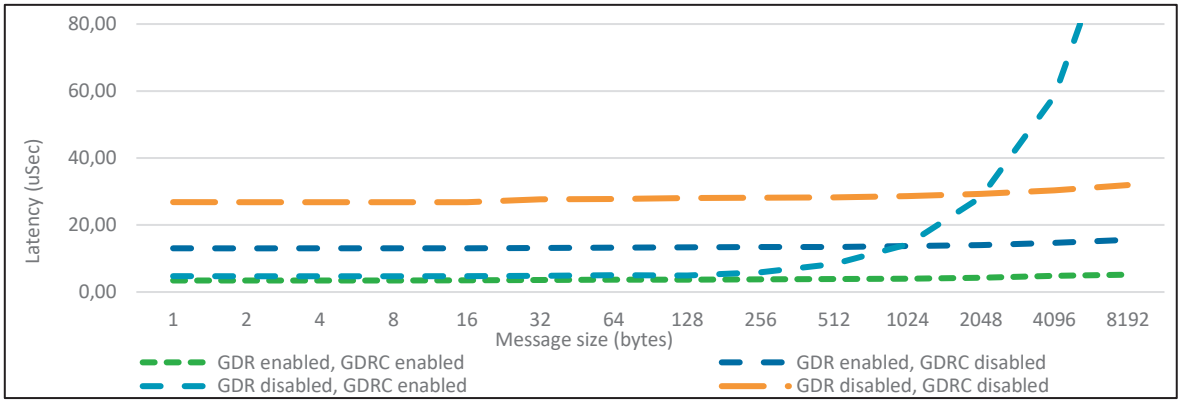

Рис. 10. Задержка на узлах типа E

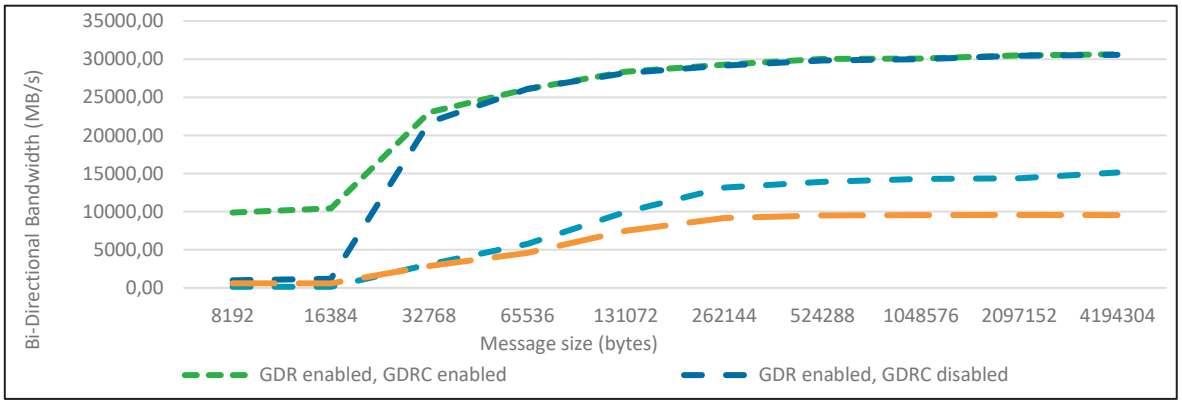

Рис. 11. Пропускная способность на узлах типа E

Наилучшие результаты применения технологий GDR и GDRC были получены на наиболее современных вычислительных узлах типа E. Задержка при передаче данных уменьшилась в 7.8 раз. Увеличение пропускной способности составило до 286% на небольших пакетах данных и до 100% на пакетах данных размером более 64 КБ. Пиковая скорость передачи данных достигла 30 400 МБ/с (243.2 Гигабит/с). Негативного влияния от включения сразу двух технологий (GDR и GDRC) на вычислительных узлах типа E замечено не было, поэтому по умолчанию они будут включены обе.

5. Выводы

Оптимальным вариантом для снижения задержки на передачу данных в память GPU для всех типов узлов стало совместное применением обеих технологий: GDR и GDRC. Что касается пропускной способности, то здесь архитектурные особенности сыграли ключевую роль, так лучшим вариантом для узлов типов A/B, C оказался вариант с применением GDRC, но с отключением GDR. Для наиболее современных вычислительных узлов Типа E – наибольший эффект дало включение обеих технологий.

Технологии GDR и GDRC доступны для использования на суперкомпьютере cHARISMa НИУ ВШЭ. Пользователи могут активировать их использование путем установки переменных окружения во время постановки задачи в очередь. В дальнейшем планируется реализовать более удобный автоматический механизм применения технологий, исключающий человеческие ошибки.

Внедрение современных коммуникационных технологий положительно сказывается на производительности суперкомпьютера cHARISMa в условиях высокой нагрузки. В результате внедрения и настройки технологии GPUDirect на суперкомпьютере cHARISMa НИУ ВШЭ производительность ПО, использующего GPU, повысилась на 3% по данным системы HPC TaskMaster [6] (на данный момент это общая цифра сводной статистики – в дальнейшем планируется оценить влияние на различные прикладные пакеты и процессы обучения искусственных нейронных сетей). Исходя из статистики расчетов с применением GPU на данном суперкомпьютере, такой прирост можно оценить как высвобождение до 37 000 GPU-часов машинного времени в год. Дополнительное машинное время позволяет выполнять большее количество научных проектов без затрат на покупку новых вычислительных узлов.

Литература

1. Чулкевич Р.А., Козырев В.И., Шамсутдинов А.Б., Костенецкий П.С. Сравнение производительности параллельной СХД суперкомпьютера с разными версиями файловой системы Lustre // Russian Supercomputing Days, 2022.
2. Ammendola R. et al. GPU peer-to-peer techniques applied to a cluster interconnect // 2013 IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum, Cambridge, MA, USA, 2013, pp. 806-815. DOI: 10.1109/IPDPSW.2013.128.
3. Hanafy W. et. al. Understanding the Benefits of Hardware-Accelerated Communication in Model-Serving Applications // arXiv, 2023/
4. Hollowell C. et. al. The Effect of NUMA Tunings on CPU Performance // Journal of Physics: Conference Series. 2015. Vol. 664, P. 092010. DOI: 10.1088/1742-6596/664/9/092010.
5. Kostenetskiy P.S., Chulkevich R.A., Kozyrev V.I. HPC Resources of the Higher School of Economics // Journal of Physics: Conference Series. 2021. Vol. 1740, No. 1. P. 012050. DOI: 10.1088/1742-6596/1740/1/012050.
6. Kostenetskiy P.S., Chulkevich R.A., Kozyrev V.I., Shamsutdinov A.B. HPC TaskMaster - Task Efficiency Monitoring System for the Supercomputer Center // Parallel Computational Technologies. PCT 2022. Communications in Computer and Information Science, vol 1618. Springer, Cham. DOI: 10.1007/978-3-031-11623-0_2.

7. OSU Micro-Benchmarks. The Ohio State University. URL: http://mvapich.cse.ohio-state.edu/benchmarks/ (дата обращения: 01.07.2023).
8. RedHat Performance Tuning Guide. URL: https://access.redhat.com/documentation/ru-ru/red_hat_enterprise_linux/7/html/performance_tuning_guide/sect-red_hat_enterprise_linux-performance_tuning_guide-tool_reference-numactl (дата обращения: 01.07.2023).
9. Rossetti D. Benchmarking GPUDirect RDMA on Modern Server Platforms. URL: http://devblogs.nvidia.com/benchmarking-gpudirect-rdma-on-modern-server-platforms (дата обращения: 01.07.2023).
10. Shi R., et.al. Designing efficient small message transfer mechanism for inter-node MPI communication on InfiniBand GPU clusters // 21st International Conference on High Performance Computing, HiPC 2014. 10.1109/HiPC.2014.7116873.
11. Spafford K., Meredith J., Vetter J. Quantifying NUMA and contention effects in multi-GPU systems // Proceedings of the Fourth Workshop on General Purpose Processing on Graphics Processing Units (GPGPU-4), 2011. Article 11, 1–7. DOI: 10.1145/1964179.1964194.
12. Venkatesh A., Subramoni H., et.al. A high performance broadcast design with hardware multicast and GPUDirect RDMA for streaming applications on InfiniBand clusters // 21st International Conference on High Performance Computing (HiPC), Goa, India, 17-20 Dec. 2014, Proceedings. P. 1-10, 2014 doi: 10.1109/HiPC.2014.7116875.