УДК 004.7; 004.4

РАЗРАБОТКА АРХИТЕКТУРЫ РАСПРЕДЕЛЕННОГО ИНТЕРКОННЕКТА*

Г.Ф. Масич, *Институт механики сплошных сред УрО РАН; Пермский национальный исследовательский политехнический университет*
А.Г. Вотинова, *Пермский научный центр УрО РАН*
А.В. Созыкин, *Институт математики и механики УрО РАН*
А.Г. Масич, *Институт механики сплошных сред УрО РАН*
В.А. Щапов, *Институт механики сплошных сред УрО РАН; Пермский национальный исследовательский политехнический университет*
А.С. Игумнов, *Институт математики и механики УрО РАН*
А.В. Бобров, *Институт механики сплошных сред УрО РАН*
С.Р. Латыпов, *Институт механики сплошных сред УрО РАН*
А.Ю. Береснев, *Институт математики и механики УрО РАН*
Е.Ю. Куклин, *Институт математики и механики УрО РАН*

Описаны разработанные архитектурные решения, используемые при построении распределенной вычислительной среды Уральского отделения РАН. Особое внимание уделено измерениям и способам повышения производительности коммуникационной среды. Обсуждается метод запуска вычислительной задачи в разнородном окружении.

Ключевые слова: *суперкомпьютер, интерконнект, оптическая сеть, MPI, пропускная способность, контейнерная виртуализация.*

Введение

Известно, что суперкомпьютер представляет собой массивно параллельную вычислительную систему с распределенной памятью, состоящий из множества вычислительных узлов, соединенных между собой несколькими независимыми сетями (interconnect InfiniBand / Omni-Path / PCIe / Ethernet и т.д.). Interconnect суперкомпьютеров далее по тексту будем называть «внутренний интерконнект». Совокупность внутренних интерконнектов суперкомпьютеров, систем хранения и их объединяющей скоростной оптической сети назван нами «распределенный интерконнект».

Зарубежные прототипы – проекты DAS-5 [1] и InfiniCortex [2]. DAS-5 (Distributed ASCI Supercomputer) – продолжающийся с 2001 года проект создания и развития распределенного ASCI суперкомпьютера, состоящего из шести территориально распределенных разнородных кластеров в Нидерландах, расположенных в голландских университетах. ASCI является аббревиатурой от «Advanced Schoolfor Computing and Imaging» – голландской магистерской школы. В этом проекте интерконнекты вычислителей соединены по выделенным 10 Гбит/с лямбда-каналам связи научно-образовательной сети SURFnet. Проект InfiniCortex стартовал в 2014 году, использует 10–100 Гбит/с каналы связи между континен-

* Работа выполнена при финансовой поддержке РФФИ и Правительства Пермского края (грант № 11-07-96001).

тами Азии, Северной Америки, Австралии и Европы для соединения отдельных InfiniBand подсетей суперкомпьютеров для создания единого вычислительного ресурса (Galaxy of Supercomputers). Используется технология поддержки работы InfiniBand протокола по трансконтинентальным расстояниям – Obsidian Strategics' Longbow technology.

Наш подход, на данной стадии его реализации, использует Ethernet интерконнект объединяемых вычислителей по каналам $n \times x$ 10 Гбит/с.

Инфраструктура суперкомпьютерной среды GIGA URAL

Распределенная вычислительная среда (рис. 1) строится на вычислительных ресурсах ИМСС УрО РАН в Перми («Тритон», 4,5 Тфлопс) и ИММ УрО РАН в Екатеринбурге («УРАН», 225,85 Тфлопс) [3].

Внутренние интерконнекты суперкомпьютеров используют сеть InfiniBand (IB) 20 Гбит/с и сеть Ethernet 1 Гбит/с (1 GE). Сеть Ethernet используется нами для подключений к распределенному интерконнекту.

Распределенная система хранения данных (РСХД) сформирована на серверах производства компании «Supermicro» общей емкостью 3×72 Тбайта в Екатеринбурге и 36 Тбайт в Перми, которые оснащены портами 10 GE. РСХД, построена на базе программного обеспечения dCache из проекта European Middleware Initiative [4].

Распределенный интерконнект создан посредством соединения Ethernet сетей конечных систем каналами связи DWDM тракта Пермь–Екатеринбург (30 Гбит/с) научно-образовательной сети GIGA URAL [5]. Тракт построен на двух «темных» волокнах оптической длиной L=456 км. CWDM тракт ИМСС–ПНЦ соединяет на скорости 4×10 GE вычислительные ресурсы ИМСС с местом окончания DWDM тракта в Перми.

План адресации. Используется приватный блок IPv4 адресов 10.0.0.0/8. В структуре четырехбайтового адреса выделены поля [10.<techno><site-id>.<node-id>.<node-id>], указывающие на техноло-

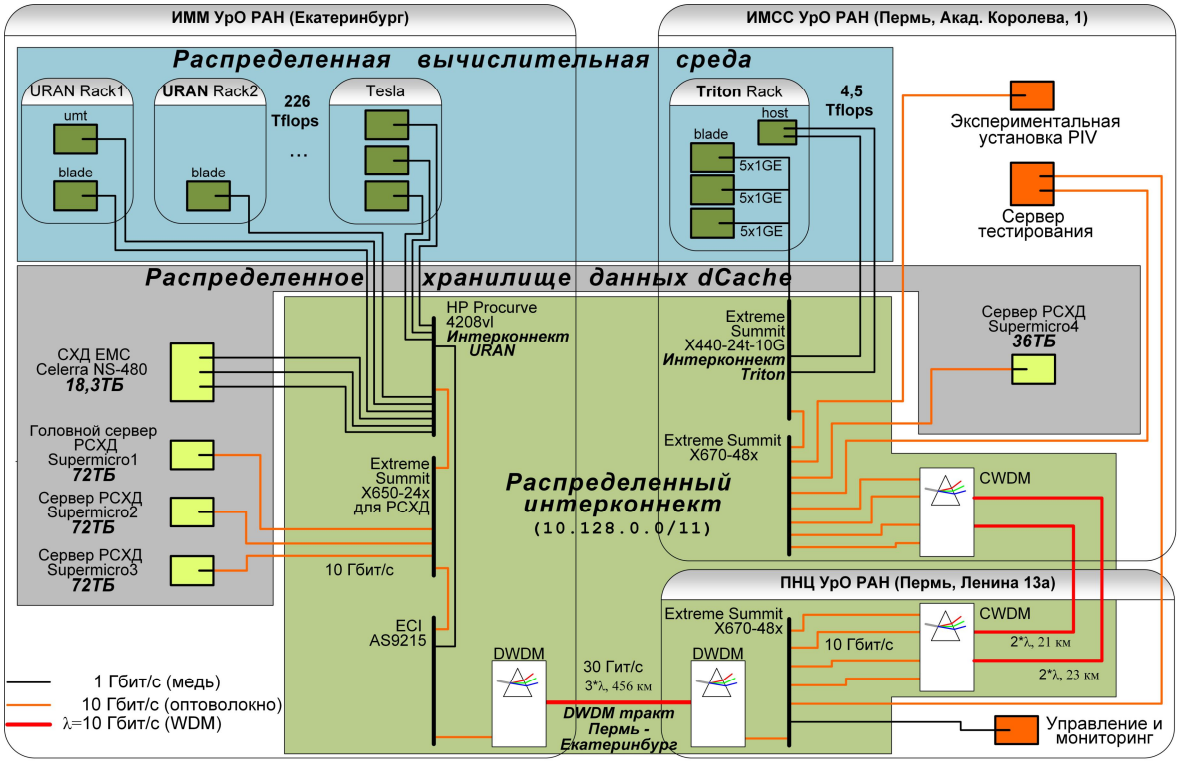

Рис. 1. Физическая структура распределенной суперкомпьютерной среды

гию сети в поле <techno> (Ethernet/IB), место расположения <site-id> (Пермь / Екатеринбург) и собственно нумерацию <node-id> портов вычислительных узлов и узлов хранения данных. Распределен приватный блок адресов 10.128.0.0/9, который разделен на 4 подсети (для двух Ethernet и двух IB интерконнектов). Используемая в настоящее время подсеть 10.128.0.0/11 объединяет основные Ethernet интерфейсы вычислительных узлов, систем хранения и экспериментальных установок.

Система мониторинга реализована на свободном ПО Zabbix, которое позволяет по протоколу SNMP наблюдать за физическими (температура, сбои, загрузка) и функциональными (объем трафика, количество ошибок) характеристиками отдельных систем. Мониторинг DWDM оборудования осуществляется поставляемой производителем проприетарной системой мониторинга и управления LightSoft v8. Созданная трехкомпонентная система мониторинга, поддерживающая наблюдение и управление за вычислителями, хранилищами и коммуникациями, используется для поддержания круглосуточной работоспособности суперкомпьютерной среды.

Физические пределы скорости и латентности

Суперкомпьютер «Уран» состоит из базовых блоков, в каждом из которых размещено по 32 вычислительных узла, подключенных по 1 GE к двум встроенным Ethernet коммутаторам согласно схеме на рис. 2. Коммутаторы базовых блоков подключены к внешнему коммутатору при помощи четырех линий на скорости 1 GE, объединенных по технологии Link Aggregation Control Protocol (LACP) с целью увеличения суммарной пропускной способности агрегированного канала (транк) до 4 Гбит/с. Суперкомпьютер «Тритон» содержит в каждом базовом блоке 16 узлов, подключенных по 1 GE к одному встроенному коммутатору. Коммутаторы базовых блоков соединены с внешним коммутатором при помощи пяти линий по 1 GE, обеспечивая суммарную пропускную способность танка 5 Гбит/с. Внешние коммутаторы Перми и Екатеринбурга соединены на гарантированной скорости 10 Гбит/с. Для исследования протоколов выполнено подключение серверов тестирования и реализован механизм формирования петли в Екатеринбурге.

Одной из самых распространенных технологий для обмена данными между процессами, выполняющимися на одном или нескольких узлах, является MPI. Сообщения могут передаваться не только от одного процесса к другому, но и внутри групп процессов, например, от одного процесса из группы всем или от всех – к одному.

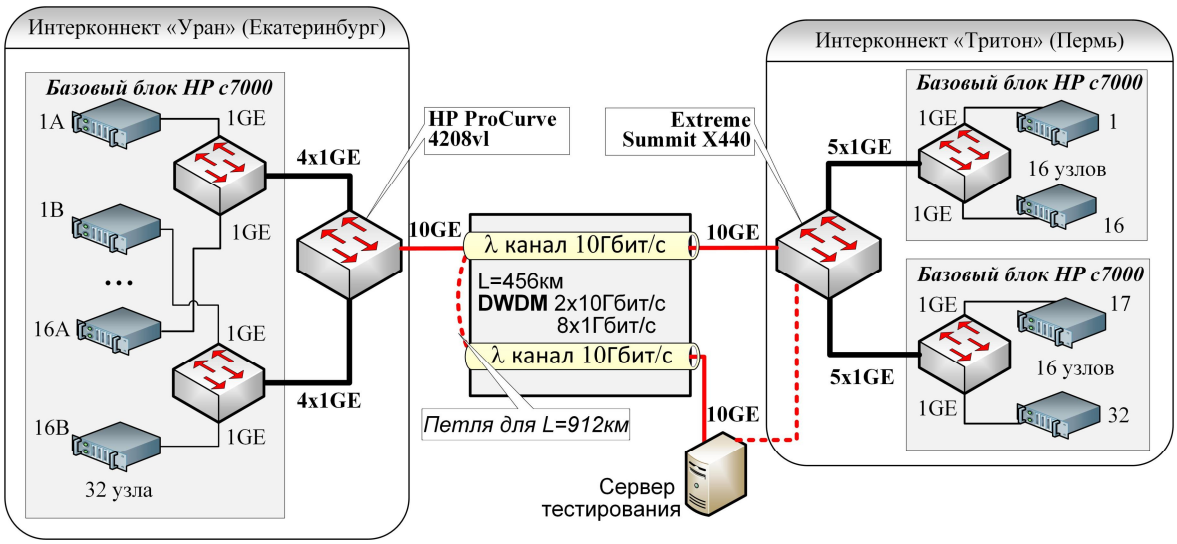

Рис. 2. Структура каналов связи

Как видно из рис. 2, ***физические пределы максимальной скорости*** межузового обмена имеют следующие значения: между двумя любыми узлами – 1 Гбит/с; между четырьмя узами одного и четырьмя узами другого базового блока – 4 Гбит/с. Максимальная скорость обмена данными между вычислительными узлами не может превышать 10 Гбит/с. Расчетное значение задержки в сети (***латентность)*** определяется скоростью распространения сигнала по оптическому волокну, которая является физической константой 200 000 км/с (0,5 мкс/км). Латентности для длин линий 456 км и 912 км (в петле) равны 2,28 мс и 4,56 мс соответственно. Эти величины задержек экспериментально подтверждены ping измерениями RTT (Round Trip Time), значения которых равны 5,3 мс и 10,6 мс. Незначительное превышение измеренной задержки над расчетной (5,3 > 2×2,28) объясняется задержками в коммутационном облаке и транспондерах DWDM системы.

Для эффективной работы алгоритмов управления перегрузкой в протоколах с обратной связью (TCP) требуется, чтобы задаваемый при конфигурировании TCP размер окна перегрузки *cwnd* был не меньше параметра BDP=*bandwidth*×*RTT*. Расчетные значения BDP: 0,6625 MB для канала 456 км/1 GE; 6,625 MB для канала 456 км/10 GE и 13,25 MB для канала 912 км/10 GE.

Результаты измерений

Цель измерений – поиск узких мест коммуникационной среды и способов их устранения. Показано, что наличие даже небольшого количества (~ 10%) стороннего трафика в коммуникационной среде существенно снижает эффективность многих алгоритмов управления перегрузкой TCP. Большинство алгоритмов достигают совокупной скорости передачи данных, близкой к максимальной пропускной способности канала при числе параллельных потоков в диапазоне от 4 до 16.

Утилитой Intel MPI Benchmark измерено фактическое время выполнения PingPong операции между двумя узлами в разных городах в зависимости от размера сообщения. Использованы следующие реализации MPI: OpenMPI, MPich и Intel MPI. Выявлено [6], что время передачи сообщений размером до 128 Кбайт определяется латентностью канала связи, реализация Intel MPI при размере сообщений более 128 Кбайт оказалась более эффективной, чем у OpenMPI и Mpich.

При одновременной (параллельной) передаче данных от четырех узлов в Екатеринбурге до четырех узлов в Перми суммарная скорость составила 2,2 Гбит/с, вместо ожидаемых 4 Гбит/с. Причина – неравномерная балансировка пакетов по линкам транка, основанная на операция XOR трех младших битов IP-адреса отправителя и получателя. Перенумерация узлов решает эту проблему.

Проводилось исследование производительности таких групповых операций, как MPI_Allreduce, MPI_Reduce, MPI_Reduce_scatter, MPI_Allgather, MPI_Allgatherv, MPI_Gather, MPI_Gatherv, MPI_Scatter, MPI_Scatterv, MPI_Alltoall, MPI_Alltoallv, MPI_Bcast [6]. Использованы следующие реализации MPI: OpenMPI, Mpich и Intel MPI. Измерения проводились для следующих комбинаций расположения 16 узлов: чередование узлов Екатеринбург, Пермь, Екатеринбург...; один узел в Перми, пятнадцать в Екатеринбурге; восемь в Екатеринбурге, восемь в Перми; шестнадцать в Екатеринбурге. Оказалось, что во всех трех рассматриваемых реализациях MPI время выполнения групповых операций зависело от порядка указания узлов при запуске MPI-задачи. С практической точки зрения, это означает, что можно ускорить выполнение программы, переупорядочив узлы, например, замеряя время выполнения программы для меньшего размера входных данных или при меньшем числе итераций.

По результатам измерений можно сформулировать следующие рекомендации для повышения производительности. Предпочтительным алгоритмом организа-

ции транков является алгоритм, основанный на IP-адресах. Большая латентность приводит к существенному падению производительности MPI-программ, интенсивно обменивающихся сообщениями. Однако во многих случаях производительность можно улучшить, переупорядочив узлы при запуске распределенной задачи. Парадоксальное поведение Intel MPI_Bcast показывает, что для некоторого класса MPI–задач большая задержка в линии, связывающей два кластера, не приводит к заметной потере производительности.

Метод запуска задач в распределенной вычислительной среде

Одной из проблем, возникающих при разработке распределенного суперкомпьютера, является запуск вычислительной задачи в разнородном окружении. Вычислительные узлы суперкомпьютеров, объединяемые в единое вычислительное пространство, имеют различные версии операционных систем, компиляторов, системных и прикладных библиотек, в том числе библиотек MPI, которые могут быть несовместимы между собой. Это делает невозможным прямой перенос скомпилированных расчетных приложений между такими вычислительными узлами. Для решения этой проблемы можно использовать виртуализации. Однако технологии полной виртуализации вносят существенные накладные расходы, тем самым уменьшая доступную расчетным приложениям производительность. В то же время в операционной системе Linux есть поддержка виртуализации на уровне операционной системы, или контейнерная виртуализация. Использование контейнерной виртуализации позволяет запускать в дочерних контейнерах только ту же операционную систему, что и базовая, однако версии пользовательских окружений дочерних операционных систем могут быть другими и в них может быть установлен другой набор программного обеспечения и библиотек.

Таким образом, использование методов контейнерной виртуализации в операционной системе Linux является хорошим методом построения совместимых окружений для запуска вычислительных задач поверх вычислительных узлов объединяемых суперкомпьютеров. Помимо этого, запуск задач внутри контейнеров позволит предоставлять им и дополнительное программное обеспечение, и библиотеки, которые невозможно установить непосредственно на узлы суперкомпьютеров.

В вычислительной среде УрО РАН на вычислительном кластере «Тритон» ИМСС УрО РАН используется система Docker, предоставляющая инструментарий для управления контейнерами поверх стандартного ядра Linux. В рамках опытной эксплуатации успешно реализован запуск расчетных MPI-задач внутри контейнеров с использованием штатной системы запуска задач SLURM [7].

Выводы

Разработанная архитектура является одним из путей движения к ExaScale системам, решая проблему огромных требований на местах к мощности электропитания. Реализация такой архитектуры возможна только при тесном сотрудничестве системных и сетевых администраторов. Подтверждается лидерство научно-образовательных сетей в области инноваций, четко дифференцирующих себя от коммерческих предложений сетевых услуг.

Библиографический список

1. Distributed ASCI Supercomputer. URL: http://www.cs.vu.nl/das5/connectivity.shtml
2. *Michalewicz M., Southwell D., Tan T. W., Poppe Y., Klasky S., Deng Y., Wolf M., Parashar M., Kurc T., Chang C. C.-S., Matsuoka S., Muira S., Chrzęszczyk J., Howard A.* InfiniCortex: concurrent supercomputing across the globe utilising trans-continental InfiniBand and Galaxy of Supercomputers // Supercomputing 2014: The International Conference for High Performance Computing, Networking, Storage and Analysis. – At New Orleans, LA, USA, November 2014.

3. *Гольдштейн М.Л., Созыкин А.В., Масич Г.Ф., Масич А.Г.* Вычислительные ресурсы УрО РАН. Состояние и перспективы // Параллельные вычислительные технологии (ПаВТ'2013): Труды междунар. науч. конф. – Челябинск: Издательский центр ЮУрГУ, 2013. – С. 330–337.
4. *Kuklin E.Yu., Sozykin A.V., Bersenev A.Yu., Masich G.F.* Distributed dCache-based storage system of UB RAS // Computer Research and Modeling. – 2015. – Vol. 7. – № 3. – P. 559–563.
5. *Масич Г.Ф., Масич А.Г.* От «Инициативы GIGA UrB RAS» к Киберинфраструктуре УрО РАН // Вестник Пермского научного центра УрО РАН. – 2009. – № 4. – С. 41–56.
6. *Берсенев А.Ю., Игумнов А.С., Масич А.Г., Масич Г.Ф., Щапов В.А.* Исследование и анализ производительности распределенного интерконнекта вычислительной среды в УрО РАН // Суперкомпьютерные дни в России: Труды междунар. конф. – М.: Изд-во МГУ, 2016. – С. 199–210.
7. *Щапов В.А., Денисов А.В., Латыпов С.Р.* Применение контейнерной виртуализации Docker для запуска задач на суперкомпьютере // Суперкомпьютерные дни в России: Труды междунар. конф. – М.: Изд-во МГУ, 2016. – С. 505–511.

DISTRIBUTED INTERCONNECT ARCHITECTURE DEVELOPMENT

G.F. Masich[1,4], A.G. Votinova[3], A.V. Sozykin[2], A.G. Masich[1], V.A. Shchapov[1,4], A.S. Igumnov[2], A.V. Bobrov[1], S.R. Latypov[1], A.Yu. Bersenev[2], E.Yu. Kuklin[2]

[1] *Institute of Continuous Media Mechanics RAS UD*
[2] *N.N. Krasovskii Institute of Mathematics and Mechanics RAS UD*
[3] *Perm scientific centre RAS UD*
[4] *Perm National Research Polytechnic University*

This paper describes the developed architectural solutions, which are used in the construction of the distributed computing environment of the Ural Division of RAS. A special attention is paid to measurements and methods of enhancing productivity of the communication environment. The method of computational task launching in a heterogeneous environment is under discussion.

Keywords: supercomputer, interconnect, fiber optic network, MPI, throughput bandwidth, container virtualization.

Сведения об авторах

Масич Григорий Федорович, кандидат технических наук, заведующий лабораторией телекоммуникационных и информационных систем, Институт механики сплошных сред УрО РАН (ИМСС УрО РАН), 614013, г. Пермь, ул. Академика Королева, 1; доцент кафедры информационных технологий и автоматизированных систем (ИТАС), Пермский национальный исследовательский политехнический университет (ПНИПУ), 614990, г. Пермь, Комсомольский пр., 29; e-mail: masich@icmm.ru

Вотинова Анастасия Григорьевна, кандидат физико-математических наук, ведущий специалист, Пермский научный центр УрО РАН (ПНЦ УрО РАН), 614990, г. Пермь, ул. Ленина, 13А; e-mail: votinova@permsc.ru

Созыкин Андрей Владимирович, кандидат технических наук, заведующий отделом вычислительной техники, Институт математики и механики им. Н.Н.Красовского УрО РАН (ИММ УрО РАН), 620990, г. Екатеринбург, ул. Софьи Ковалевской, 16; e-mail: avs@imm.uran.ru

Масич Алексей Григорьевич, младший научный сотрудник, ИМСС УрО РАН; e-mail: mag@icmm.ru

Щапов Владислав Алексеевич, кандидат технических наук, младший научный сотрудник, ИМСС УрО РАН; доцент кафедры ИТАС, ПНИПУ; e-mail: shchapov@icmm.ru

Игумнов Александр Станиславович, заведующий отделом системного обеспечения, ИММ УрО РАН; e-mail: igumnov@imm.uran.ru

Бобров Артем Викторович, младший научный сотрудник, ИМСС УрО РАН; e-mail: bobrov@icmm.ru

Латыпов Станислав Рашидович, ведущий инженер, ИМСС УрО РАН; e-mail: LatypovSR@icmm.ru

Берсенев Александр Юрьевич, ведущий программист, ИММ УрО РАН; e-mail: bay@hackerdom.ru

Куклин Евгений Юрьевич, ведущий специалист, ИММ УрО РАН; e-mail: key@imm.uran.ru

Материал поступил в редакцию 21.10.2016 г.