Исследование и анализ производительности распределенного интерконнекта вычислительной среды в УрО РАН*

А.С. Игумнов[2,4], А.Ю. Берсенев[2,4], А.Г. Масич[1], Г.Ф. Масич[1,3], В.А. Щапов[1,3],
[1]ИМСС УрО РАН, [2]ИММ УрО РАН, [3]ПНИПУ, [4]УрФУ

Современный этап развития технологий распределенных вычислений ориентирован на использование скоростных оптических сетей и проведение “вычислений в памяти”. Именно эти обстоятельства используются нами в работах по созданию распределенной вычислительной среды на ресурсах суперкомпьютерного центра ИММ УрО РАН (Екатеринбург) и ИМСС УрО РАН (Пермь). Ключевая особенность нашего подхода заключается в соединении скоростной оптической сетью “внутренних” интерконнектов вычислителей, систем хранения и экспериментальных установок. Это направление работ, как и архитектурное решение, получило название “распределенный интерконнект”. В данной работе представлены измерения, показывающие влияние сверхдлинного интерконнекта, на производительность MPI программ.

Ключевые слова: MPI, кластер, Ethernet, распределенные вычисления, суперкомпьютеры, латентность, пропускная способность, интерконнект.

1. Введение

Текущая революция в науке – высокопроизводительные вычисления и интенсивные потоки в эру больших данных. Источники больших данных - установки, хранилища, обсерватории и вычислительные модели. Способ перемещения больших данных - доведение оптики до конечных систем, реализуемых в рамках национальных и региональных научно-образовательных оптических сетей (например, Geant2 в Европе, Internet2 в США, Initiative GIGA UrB RAS в России). Этот подход используется нами для end-to-end соединений, имеющихся в Уральском отделении вычислителей и систем хранения, экспериментальных установок и систем визуализации. В рамках «Initiative GIGA UrB RAS» планируется связать скоростной оптической магистралью со спектральным уплотнением каналов (DWDM Backbone) распределенные в пространстве ресурсы научных центров УрО РАН в городах Архангельск, Сыктывкар, Ижевск, Пермь, Екатеринбург (рис. 1). Реализованный DWDM тракт передачи данных Пермь-Екатеринбург соединяет ресурсы Пермского Научного центра с Суперкомпьютерным центром ИММ УрО РАН в Екатеринбурге на скорости 30 Гбит/с [1].

2. Обзор статей близкой тематики

Экспериментальные исследования влияния сетевой среды на MPI программы производились и ранее. Так в статье [2] исследовалась скорость выполнения групповых операций для гомогенной сети. Похожая работа [3] посвящена сравнению скорости выполнения групповых операций, предсказанных распространёнными моделями, с результатами, полученными экспериментальным путём, используя различные реализации MPI. Эксперименты проводились на гомогенной сети.

Авторы другой статьи [4], вышедшей в 2001 году, протестировали производительность двухуровневой сети. На первом уровне использовалась технология Myrinet, а на втором - ATM. Авторы симулировали различные задержки сети и пропускные способности, изменяя настройки сетевого шлюза. Кроме этого, проводилось тестирование и на реальном оборудовании, рас-

* *Исследования проводятся при поддержке РФФИ (грант №14-07-96001), комплексной программы Уральского отделения РАН (проекты № 15-7-1-25, 15-7-1-26).*

положенными в четырёх городах в Нидерландах. Задержка в одну сторону составляла 1,25 — 1,5 мс. Оценивалось время выполнения шести программ, не использующих технологию MPI.

В статье [5] исследовалась производительность 28-километрового интерконнекта Infiniband между городами Heidelberg и Mannheim в Германии. Исследовались только коммуникации точка-точка.

Кроме этого, значительная часть публикаций по данной теме фокусируется на разработке оптимальных алгоритмов для групповых операций в гетерогенных сетях.

Данную работу отличает использование длинного выделенного интерконекта, порядка нескольких сотен километров и описание неожиданного эффекта, возникшего при использовании операций MPI_Reduce, MPI_Gather, MPI_Scatter и MPI_Bcast в реализации Intel MPI.

3. Компоненты распределенной вычислительной среды УрО РАН

Распределенная вычислительная среда УрО РАН формируется на ресурсах двух центров обработки данных (ЦОД): ИММ УрО РАН (Екатеринбург) и ИМСС УрО РАН (Пермь).

В ЦОД ИММ располагаются суперкомпьютер «Уран», пиковой производительностью 225,85 Тфлопс и три сервера распределенной системы хранения данных dCache (РСХД) производства компании Supermicro.

В ЦОД ИМСС располагается вычислительный кластер Triton, пиковой производительностью 4,5 Тфлопс, один сервер распределенной системы хранения данных dCache и экспериментальные установки.

3.1 Архитектура распределенного интерконнекта

Каналы связи (рис. 1), соединяющие «внутренние интерконнект» высокопроизводительных систем суперкомпьютерного центра ИММ УрО РАН (Екатеринбург) и ИМСС УрО РАН (Пермь) созданы посредством системы спектрального уплотнения (DWDM) и Ethernet коммутирующего оборудования. DWDM участок Пермь–Екатеринбург реализован в рамках Инициативы GIGA UrB RAS. Используется темное волокно (dark fiber) магистральных операторов связи (L=456км) и DWDM оборудование компании ECI-Telecom: платформы XDM-2000 на оконечных узлах и XDM-40 на промежуточных узлах. Оптические мультиплексоры (MUX) на 16 лямбда каналов обеспечивают возможность использования транспондеров со скоростью передачи 10–40 Гбит/с в каждой лямбде.

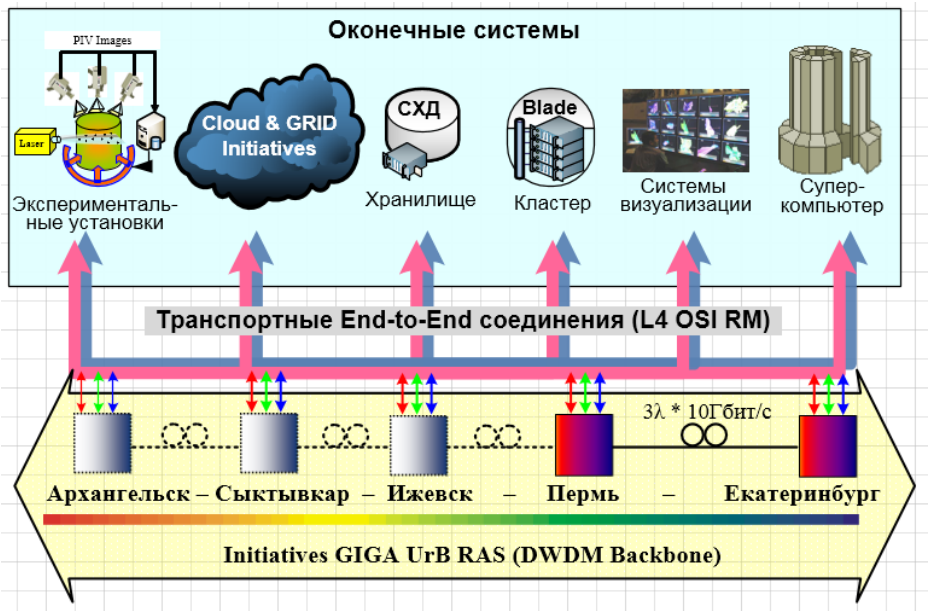

Рис. 1 Инициатива GIGA UrB RAS

3.2 Интерконнект “Уран”

Вычислительный кластер “Уран” состоит из базовых блоков HP BladeSystem c7000, каждый из которых состоит из 32 серверов HP Proliant BL 2x220c G5. Так же в состав кластера входят вычислительные модули других типов, но в измерениях, приведенных в данной работе, они не использовались для обеспечения однотипности оборудования в распределенной сети.

Вычислительные узлы кластера для работы используют две сети передачи данных. Основной сетью обмена данными MPI является InfiniBand 4xDDR пропускной способностью 20 Гбит/с. Дополнительная сеть Ethernet, пропускной способностью 1 Гбит/с, предназначена для управления потоком задач и монтирования файловых систем на вычислительные узлы.
Каждый базовый блок оборудован двумя встроенным Ethernet коммутаторами, к которым на скорости 1 Гбит/с подключен каждый вычислительный узел (рисунок 2). Коммутаторы базовых блоков подключены к внешнему коммутатору ProCurve4208vl при помощи 4-х объединенных в транк (в терминологии HP) соединений 1 Гбит/с (суммарная пропускная способность агрегированного канала – 4 Гбит/с). Коммутатор ProCurve4208vl подключен на гарантированной скорости 10Гбит/с к Ethernet интерконнекту Суперкомпьютера Triton в Перми.

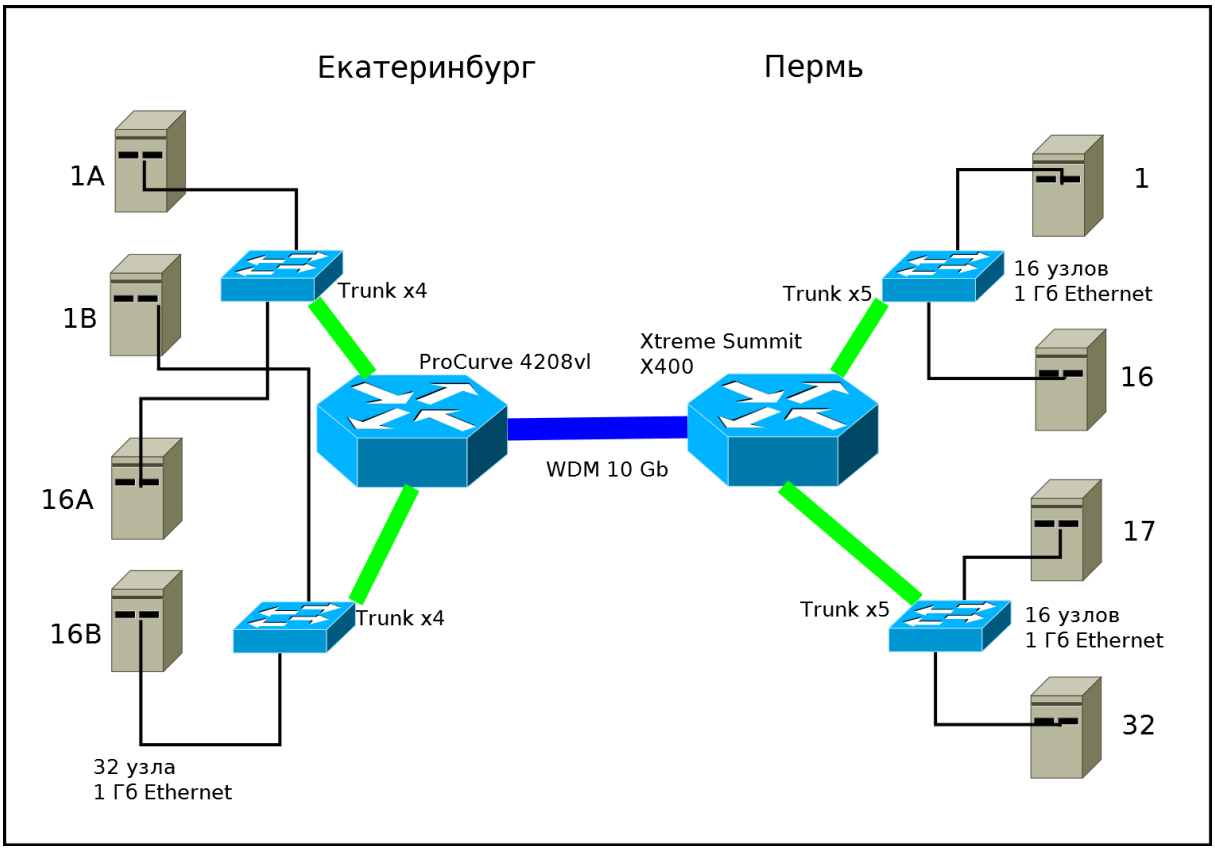

Рис. 2 Структура интерконнекта

3.3 Интерконнект Triton

Вычислительный кластер Triton состоит из трех базовых блоков HP BladeSystem c7000, каждый из которых состоит из 16 серверов HP Proliant BL 460c.

Вычислительные узлы кластера для работы используют две сети передачи данных. Основной сетью обмена данными MPI является InfiniBand 4xDDR пропускной способностью 20 Гбит/с. Дополнительная сеть Ethernet, пропускной способностью 1 Гбит/с, предназначена для управления потоком задач и монтирования файловых систем на вычислительные узлы.

Каждый базовый блок оборудован встроенным Ethernet коммутатором, к которому на скорости 1 Гбит/с подключен каждый вычислительный узел (рисунок 3). Коммутаторы базовых блоков подключены к внешнему коммутатору Extreme Summit X440 при помощи 5-и агрегиро-

ванных по технологии Link Aggregation Control Protocol (LACP) соединений 1 Гбит/с (суммарная пропускная способность агрегированного канала – 5 Гбит/с). Коммутатор Extreme Summit X440 подключен подключен на гарантированной скорости 10Гбит/с к Ethernet интерконнекту Суперкомпьютера “Уран” в Екатеринбурге.

3.4 Вычислительные узлы

Использованные для измерений вычислительные узлы (рис.3) содержит два Intel® Xeon® E5450 по четыре ядра каждый. Ядра в процессоре попарно динамически разделяют кэш 2-го уровня. Два ядра, в зависимости от взаимного расположения, могут обмениваться информацией либо через кэш 2 уровня, либо через шину FSB. Все 8 ядер для чтения/записи данных в ОЗУ, а также для синхронизации одних и тех же данных в разных кэшах используют двойную независимую шину.

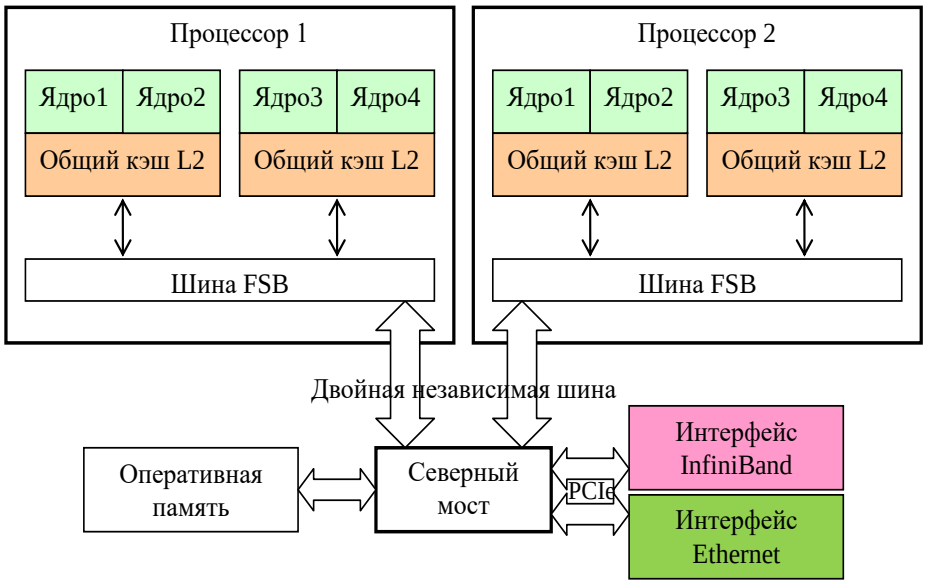

Рис. 3. Схема вычислительного узла кластера

4. Экспериментальная оценка влияния задержек на производительность MPI-операций

Одной из самых распространённых технологий для обмена данными между процессами, выполняющимися на одном или нескольких узлах, является MPI. Он предоставляет программе набор функций для передачи сообщений. 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.

Предмет нашей статьи - кластерная система, расположенная в двух городах на значительном расстоянии друг от друга. Части этой системы соединены высокоскоростным выделенным каналом, но, из-за того, что скорость передачи данных ограничена сверху скоростью света, в этом канале наблюдаются неустранимые задержки в пределах нескольких миллисекунд. Чтобы оценить насколько изменится время выполнения программы, экспериментально оценим скорость выполнения различных MPI-операций.

4.1 Операции точка-точка

Операции точка-точка, такие как MPI_Send и MPI_Recv, являются довольно предсказуемыми в смысле оценки времени их выполнения. Время, через которое принимающий получит сообщение зависит от времени распространения сигнала по оптическому волокну, скорости ввода/вывода портов вычислительных узлов, длины сообщения и процедурных характеристик транспортных протоколов.

Скорость распространения сигнала по оптическому волокну является физической константой 200000 км/с (0,5 мкс/км) и для оптической длины трассы Пермь-Екатеринбург 456 км время распространения (латентность) равно 2,28 мс.

4.1.1. Измерение посредством утилиты ping

Латентность канала можно экспериментально измерить с помощью утилиты ping, входящую в стандартную поставку многих современных операционных систем. Тест, запущенный на несколько суток, показал, что латентность не зависит от времени суток и равна примерно 2,6 мс. Некоторое превышение измеренного значения над рассчитанным (2,6 > 2,28) обусловлены задержками в мультиплексирующем и коммутирующем оборудовании end-to-end тракта

Скорость передачи данных измерялась с помощью утилиты iperf, которая передаёт данные по протоколам TCP или UDP. При передаче данных от хоста в Екатеринбурге до хоста в Перми скорость составила около 930 Мбит/сек. Это предсказуемо, так как внутренняя сеть суперкомпьютеров является гигабитной.

При одновременной передаче данных от четырёх узлов в Екатеринбурге до четырёх узлов в Перми суммарная скорость составила 2,2 Гбит/сек, вместо ожидаемых 4 Гбит/сек.

При увеличении числа узлов до 46 с каждой стороны, удалось добиться суммарной скорости передачи в одну сторону равной 3,7 Гбит/сек. Это объясняется следующими особенностями подключения узлов:

Несмотря на то, что свитч, к которому подключены узлы в Екатеринбурге связан с центральным четырьмя гигабитными линками, объединенными в транк, а коммутатор, к которому подключены узлы в Перми связан с центральным пятью гигабитными линками, они используются примерно на половину от своей максимальной пропускной способности из-за неравномерности загрузки линков транка. Анализ ситуации показал, что причина заключается в неудачной настройке IP адресации внутри корзин. Чередование четных и нечетных вычислительных узлов в сдвоенных вычислительных блоках и их отображение в виде пар в веб-интерфейсе провоцирует администратора на последовательную IP нумерацию.

Таблица 1. IP адресация в корзине кластера

x.x.x.1	x.x.x.2	x.x.x.3	x.x.x.4	…
Node 1A	Node 1B	Node 2A	Node 2B	…

Данное упорядочение, как оказалось, крайне негативно влияет на пропускную способность транка, так как алгоритм распределения пакетов по линкам вычисляет номер линка, используя XOR трёх младших битов IP-адресов отправителя и получателя, а в внутренняя коммутация корзины подключает все четные узлы к одному внутреннему коммутатору, а все нечетные - к другому. В результате, из четырех линков транка на первом коммутаторе выбираются только 1 и 3, а на втором 2 и 4. По сути дела, каждая корзина была подключена к основной сети транком из двух 1 Гб линий, что и показали результаты замеров.

Поскольку перенумерация узлов в корзинах требовала приостановки очереди задач, была предпринята попытка переключить алгоритм транка на анализ MAC-адресов пакетов. Данная попытка так же потерпела неудачу. Дело в том, что каждый вычислительный узел оборудован сдвоенным Ethernet адаптером с двумя последовательными MAC-адресами. Внутренняя разводка корзины всегда использует первый порт адаптера, который всегда имеет нечетный MAC-адрес, а это опять приводит к выбору двух линков в транке из четырех возможных.

4.1.2 Измерение производительности утилитой Intel MPI Benchmark

С помощью утилиты для изменения производительности Intel MPI Benchmark [6] было замерено фактическое время выполнения MPI-операций MPI_Send и MPI_Recv в случае, когда отправитель находится в одном кластере, а получатель в другом. Будем использовать следующие реализации MPI: OpenMPI версии 1.8.1, Mpich версии 3.1.5 и Intel MPI версии 5.1.

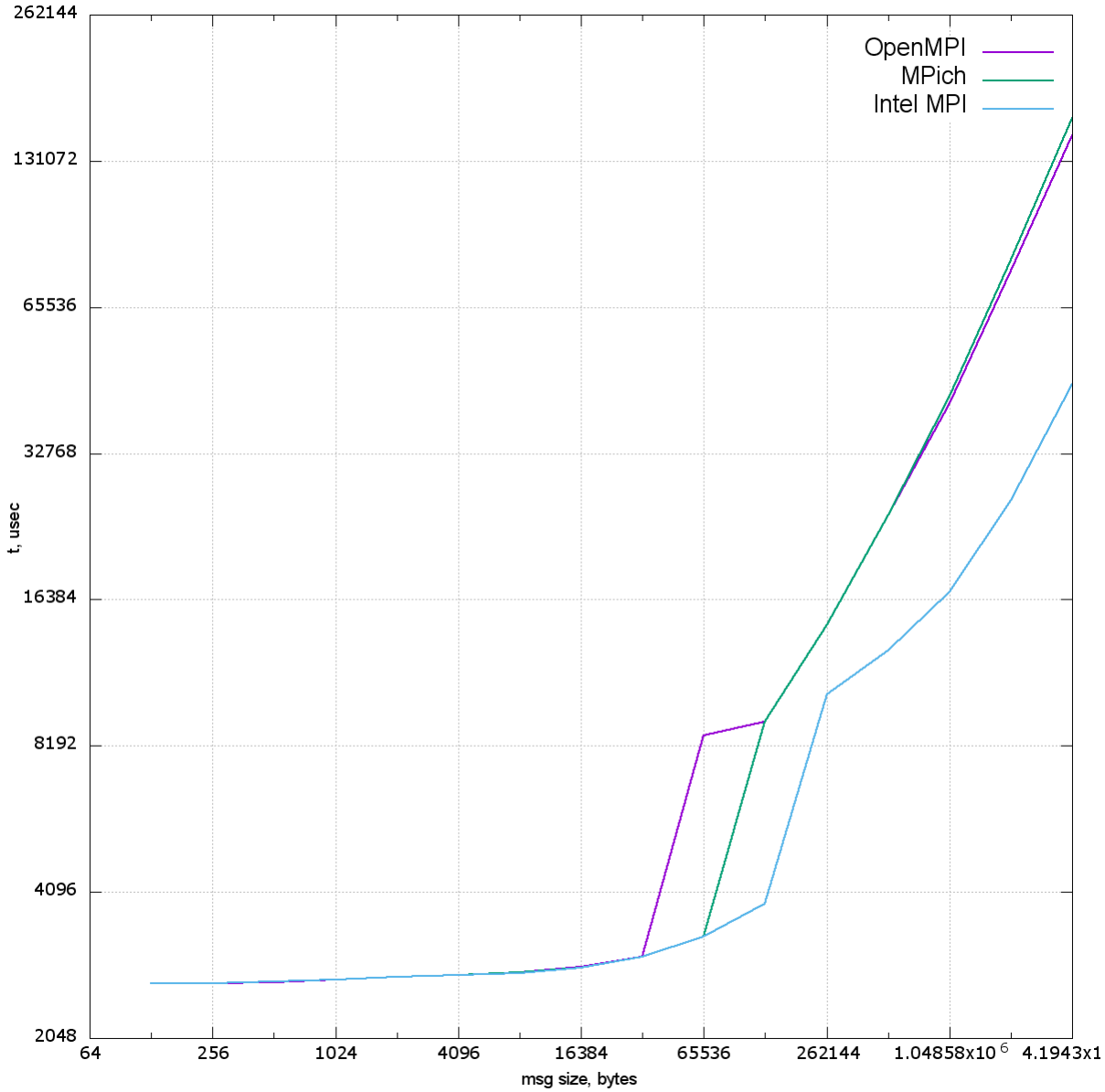

Рис. 4. Зависимость времени выполнения операции pingpong между двумя узлами в разных городах от размера сообщения. Графики - реализации MPI

По результатам тестирования можно сделать следующие выводы.

1) Латентность сети, измеренная с использованием программы Intel MPI Benchmarks 4.1 совпала с латентностью, измеренной утилитой ping

2) При небольшой длине сообщения задержка становится решающим фактором, ограничивающим использование канала

3) Реализация Intel MPI показала себя более эффективной при большом размере сообщений, чем OpenMPI и MPich

Сравним данные результаты с результатами аналогичного теста, проведённого на двух узлах одного кластера:

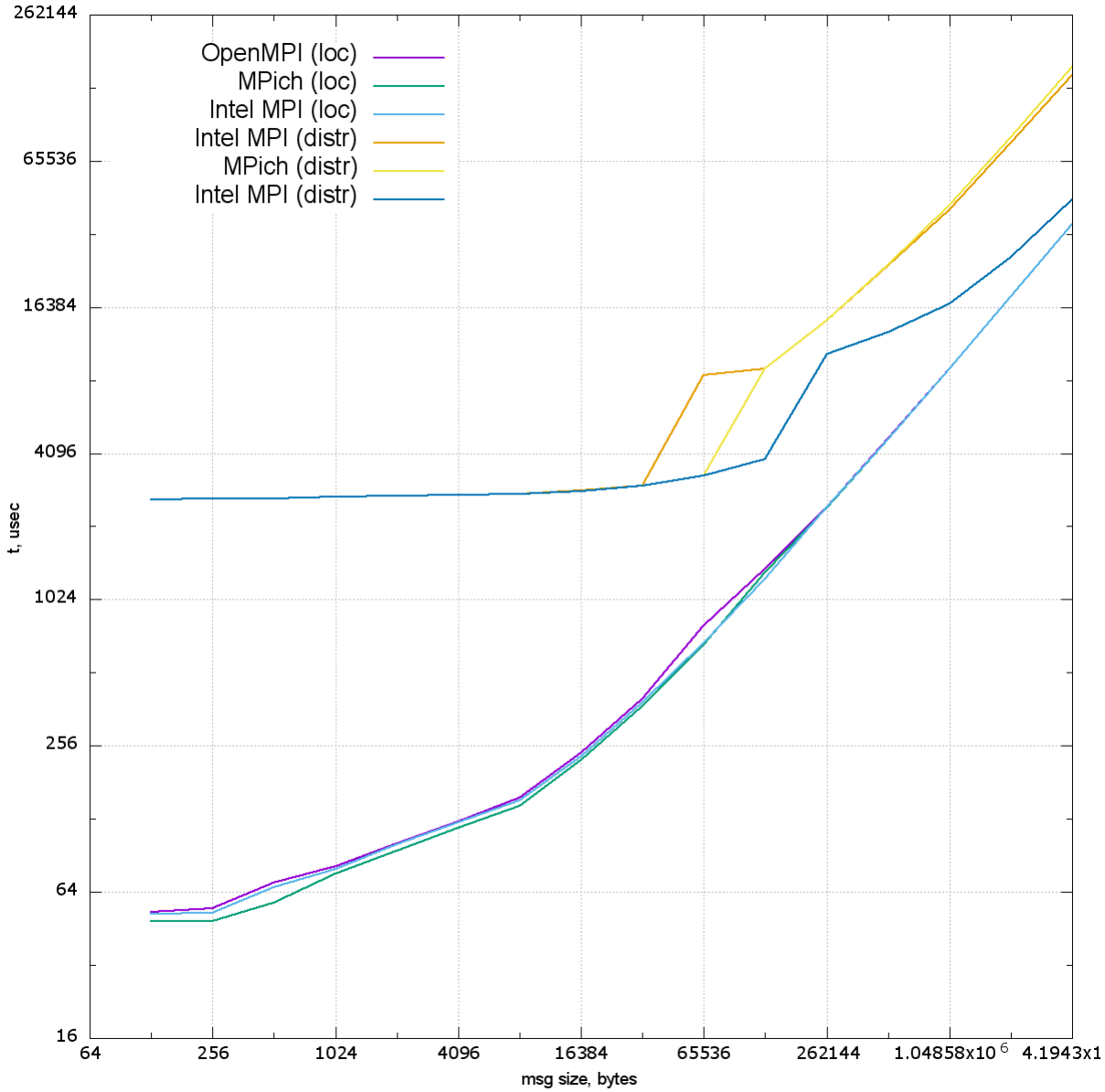

Рис. 5 Сравнение графиков из Рис.4 с тем же тестом, запущенным локально в рамках одного кластера

Можно заметить, что все три реализации используют полосу пропускания более эффективно. Реализация OpenMPI показала себя чуть менее эффективной на сообщениях средней длины.

4.2 Множественные операции точка-точка

Данный тест отличается от предыдущего тем, что используется по восемь узлов с каждой стороны. Они объединяются в группы по два узла (один узел со стороны Перми, второй со стороны Екатеринбурга) и первый узел пары посылает сообщение второму. Второй узел получает его и пересылает назад первому. Измеряется среднее время доставки сообщения и средняя пропускная способность.

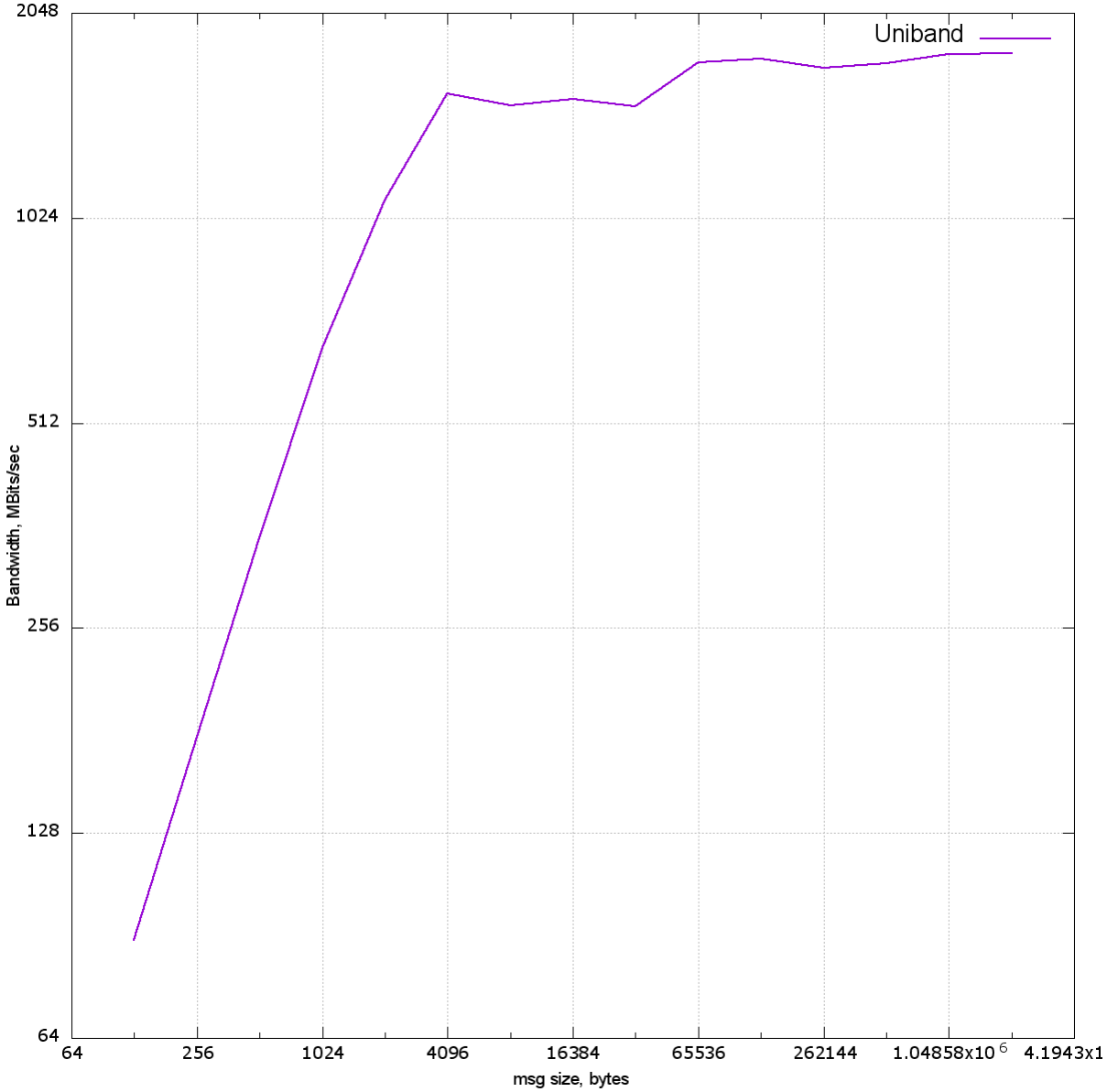

Рис. 6 Суммарная пропускная способность при передаче 8 потоков сообщений (8 узлов в Екатеринбурге обмениваются с 8 узлами в Перми). Используются операции MPI_Isend/MPI_Recv/MPI_Wait.

Средняя пропускная способность оказалась ограничена примерно 225 Мегабайтам в секунду. Это соотносится со средней пропускной способностью между данными узлами, измеренной с помощью утилиты iperf.

4.3 Групповые операции.

Исследовалась производительность 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.

На Рис.7 приведены графики выполнения операции MPI_Reduce для следующих комбинаций расположения узлов:

1) Чередование узлов Екатеринбург, Пермь, Екатеринбург, Пермь ...;
2) Один узел в Перми, пятнадцать в Екатеринбурге;
3) Восемь в Екатеринбурге, восемь в Перми;
4) Шестнадцать в Екатеринбурге.

Для остальных перечисленных операций графики имеют аналогичный характер.

Оказалось, что во всех трёх рассматриваемых реализаций 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 меньшем числе итераций.

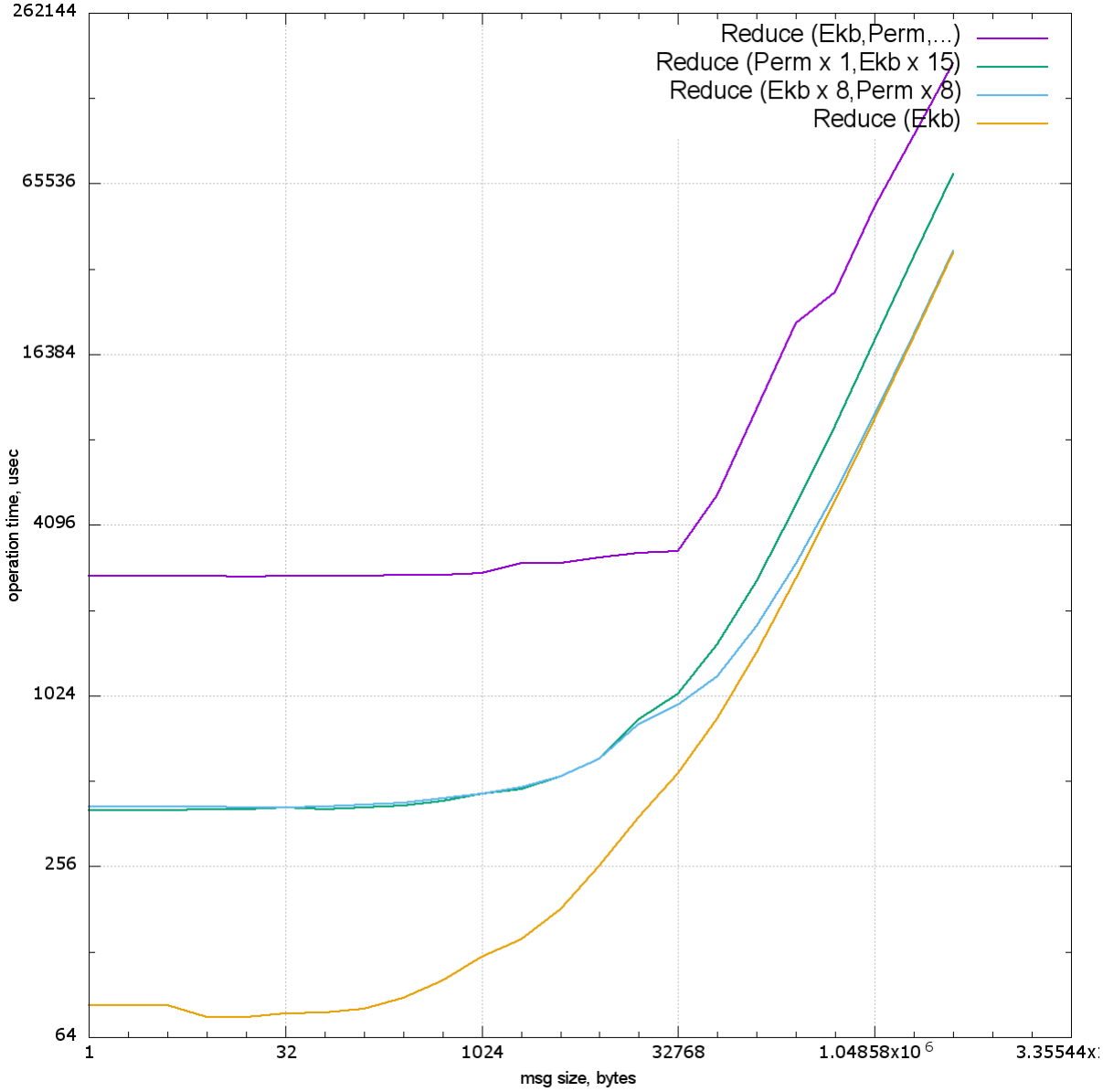

Рис. 7. Тест операции reduce на 16 узлах

Неожиданным результатом стало то, что среднее время выполнения операций MPI_Reduce, MPI_Gather, MPI_Gatherv, MPI_Scatter, MPI_Scatterv, MPI_Bcast при использовании Intel MPI и операций MPI_Gatherv, MPI_Scatter, MPI_Scatterv и MPI_Bcast при использовании MPich при размере сообщений порядка нескольких килобайт оказалось примерно равным 0,400 мс. Например, операция MPI_Bcast состоит в том, что один узел посылает данные, а 15 остальных узлов принимают их. Даже если принять, что данные в одном городе распространяются мгновенно, а в соседний город - со скоростью света в вакууме (т.е. примерно за 1,5 мс) то среднее время выполнения MPI_Bcast должно быть равным (0 * 8 + 1,5 * 8) / 16 = 0,750 мс.

Разгадка заключалась в том, что тест Intel MPI Benchmark замерял время последовательного выполнения N операций, а затем делил его на N. В операции с номером N узел с рангом N % число узлов был корнем (бродкастером). При удачном расположении рангов, а именно при таком, что четыре узла подряд находятся в Екатеринбурге, а четыре следующих - в Перми проис-

ходило следующее (для удобства изложения предположим, что в пределах одного города данные доходят за 0,05 мс, а между городами - за 2,50 мс; также предположим, что корневой узел посылает данные каждому узлу отдельно):

1) 0,00 мс: первый узел в Перми выполнял операцию MPI_Bcast. Она завершалась практически мгновенно. Данные для посылки ставились в очередь на отправку. Все остальные узлы выполняли MPI_Bcast и ждали входных данных

2) 0,05 мс: все четыре узла в Перми получали данные от MPI_Bcast и приступали к следующей итерации. Второй узел в Перми посылал данные всем остальным узлам

3) 0,10 мс, 0,15: все четыре узла в Перми получали данные и приступали к следующей операции, корневым узлом являлись третий и четвёртый узел соответственно

4) 2,50 мс: все узлы в Екатеринбурге получали данные с самой первой операции MPI_Bcast, стартовала новая итерация, ожидающая новую порцию данных

5) 2,55 мс, 2,60 мс, 2,70 мс: все узлы в Екатеринбурге получали данные со второй, третей и четвёртой операции MPI_Bcast

6) 2,70 мс: первый узел Екатеринбурга выполнял операции MPI_Bcast. Далее, шаги 1-5 повторялись для узлов Екатеринбурга

Посчитаем для первых 8 операций MPI_Bcast примерное время выполнения на каждом узле в миллисекундах и найдём среднее время их выполнения.

Таблица 2. Время выполнения MPI_Bcast

1:	0,00	0,05	0,05	0,05	2,50	2,50	2,50	2,50
2:	0,05	0,00	0,05	0,05	0,05	0,05	0,05	0,05
3:	0,05	0,05	0,00	0,05	0,05	0,05	0,05	0,05
4:	0,05	0,05	0,05	0,00	0,05	0,05	0,05	0,05
5:	2,50	2,50	2,50	2,50	0,00	0,05	0,05	0,05
6:	0,05	0,05	0,05	0,05	0,05	0,00	0,05	0,05
7:	0,05	0,05	0,05	0,05	0,05	0,05	0,00	0,05
8:	0,05	0,05	0,05	0,05	0,05	0,05	0,05	0,00
среднее	0,35	0,35	0,35	0,35	0,35	0,35	0,35	0,35

Вывод: столь малое среднее время выполнения коллективных операций обусловлено удачным расположением узлов. Если бы узлы из Перми и из Екатеринбурга чередовались, этого бы не происходило. Это было подтверждено экспериментально. Если после каждой операции MPI_Bcast выполнять MPI_Barrier, эффект так же исчезает.

6. Заключение

По результатам тестирования были сделаны следующие выводы.

Для многоголовых Ethernet адаптеров в blade-системах предпочтительным алгоритмом организации транков является алгоритм, основанный на IP-адресах. При этом IP-адреса должны назначаться способом, исключающим сильную разбалансировку трафика по линкам.

Задержка в сети порядка 2,5 мс может привести к существенному падению производительности MPI-программ, интенсивно обменивающихся сообщениями. Однако, во многих случаях производительность можно улучшить, переупорядочив узлы при запуске распределённой 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.

Литература

1. Масич Г.Ф., Масич А.Г. От «Инициативы GIGA UrB RAS» к Киберинфраструктуре УрО РАН // Вестник Пермского научного центра УрО РАН. - Пермь: ПНЦ УрО РАН, 2009. - 4: - С. 41-56.

2. Ahmad Faraj, Pitch Patarasuk, and Xin Yuan. 2007. A study of process arrival patterns for MPI collective operations. In Proceedings of the 21st annual international conference on Supercomputing (ICS '07). ACM, New York, NY, USA, P. 168-179. DOI=http://dx.doi.org/10.1145/1274971.1274996
3. J. Pjesivac-Grbovic, T. Angskun, G. Bosilca, G. E. Fagg, E. Gabriel and J. J. Dongarra, "Performance analysis of MPI collective operations," 19th IEEE International Parallel and Distributed Processing Symposium, 2005, pp. 8 pp.-. doi: 10.1109/IPDPS.2005.335
4. Aske Plaat, Henri E. Bal, Rutger F.H. Hofman, Thilo Kielmann, Sensitivity of parallel applications to large differences in bandwidth and latency in two-layer interconnects, Future Generation Computer Systems, Vol. 17, No. 6, April 2001, P. 769-782, ISSN 0167-739X, http://dx.doi.org/10.1016/S0167-739X(00)00103-5
5. H.Kredel, H.G.Kruse, S. Richling, E. Strohmaier, Performance analysis and prediction for distributed homogeneous clusters, Computer Science - Research and Development, Vol. 28, No. 2, 2013, P. 157-165, ISSN 1865-2042, http://dx.doi.org/10.1007/s00450-012-0213-5
6. Intel® MPI Benchmarks User Guide and Methodology Description // электронная документация в поставке Intel MPI Benchmark 4.1

MPI efficiency over ethernet interconnect on long distances

A.S. Igumnov[2,4], A.Y. Bersenev[2,4], A.G. Masich[1], G.F. Masich[1,3], V.A. Shchapov[1,3],

[1]ICMM UrO RAN, [2]IMM UrO RAN, [3]PNRPU, [4]UrFU

High-speed optical communication link was built between Perm (ICMM UB RAS) and Ekaterinburg (IMM UB RAS). This link operates at the network level L2 (ethernet). The link used for advanced gas and hydrodynamic studies and connects the contactless measurement unit (PIV) in Perm with a cluster "Uran" in Yekaterinburg. Series of experiments showed possibility of high-speed remote processing of the flow experimental data from PIV. After the commissioning of the cluster "Triton" in ICMM UB RAS there was a question about the appropriateness of the transfer of the computation closer to the experimental unit. This paper presents measurements showing the effect of super-long interconnect on the performance of MPI programs.

Keywords: MPI, cluster, ethernet, wide area network, supercomputing, latency, throughput, interconnect

References

1. Masich G.F., Masich A.G. Ot «Initsiativy GIGA UrB RAS» k Kiberinfrastrukture UrO RAN // Vestnik Permskogo nauchnogo tsentra UrO RAN. - Perm': PNTs UrO RAN, 2009. - 4: - S. 41-56.
2. Ahmad Faraj, Pitch Patarasuk, and Xin Yuan. 2007. A study of process arrival patterns for MPI collective operations. In Proceedings of the 21st annual international conference on Supercomputing (ICS '07). ACM, New York, NY, USA, P. 168-179. DOI=http://dx.doi.org/10.1145/1274971.1274996
3. J. Pjesivac-Grbovic, T. Angskun, G. Bosilca, G. E. Fagg, E. Gabriel and J. J. Dongarra, "Performance analysis of MPI collective operations," 19th IEEE International Parallel and Distributed Processing Symposium, 2005, pp. 8 pp.-. doi: 10.1109/IPDPS.2005.335
4. Aske Plaat, Henri E. Bal, Rutger F.H. Hofman, Thilo Kielmann, Sensitivity of parallel applications to large differences in bandwidth and latency in two-layer interconnects, Future Generation Computer Systems, Vol. 17, No. 6, April 2001, P. 769-782, ISSN 0167-739X, http://dx.doi.org/10.1016/S0167-739X(00)00103-5
5. H.Kredel, H.G.Kruse, S. Richling, E. Strohmaier, Performance analysis and prediction for distributed homogeneous clusters, Computer Science - Research and Development, Vol. 28, No. 2, 2013, P. 157-165, ISSN 1865-2042, http://dx.doi.org/10.1007/s00450-012-0213-5
6. Intel® MPI Benchmarks User Guide and Methodology Description // electronic documentation of Intel MPI Benchmark 4.1