ТЕМП ВЫДАЧИ СООБЩЕНИЙ КАК МЕРА КАЧЕСТВА КОММУНИКАЦИОННОЙ СЕТИ

Ю.А. Климов, А.Ю. Орлов, А.Б. Шворин

Введение

Основными параметрами сети являются *пропускная способность* (bandwidth) и *коммуникационная задержка* (latency). Значения этих параметров публикуются на сайтах разработчиков сетей как наиболее важные; кроме того, существует множество тестов, предназначенных для их измерения [5-7]. Действительно, и пропускная способность, и задержка во многом определяют эффективность сети на реальных задачах. Однако есть класс задач, где требуется послать сразу очень много небольших сообщений, возможно разным адресатам. В этом случае на первый план выходит такая характеристика сети как *темп выдачи сообщений* (message rate).

Темп имеет значение!

В данной работе проведено сравнение по темпу выдачи сообщений коммуникационных сетей суперкомпьютера СКИФ-Аврора [1,4,11]: специализированной сети 3D-тор, разработанной авторами данных тезисов, и универсальной сети InfiniBand QDR.

Темп выдачи сообщений

Темп выдачи определяется как количество сообщений, которое узел может выдать в сеть в единицу времени. При этом нужно проводить усреднение по достаточно длительному промежутку времени, иначе мы измерим скорость заполнения входных буферов. Очевидно, значение данной величины зависит от размера сообщения. Также можно заметить, что значение темпа выдачи принципиально ограничено пропускной способностью сети. Действительно, сообщение длины L не может быть передано быстрее, чем за время L/B (где B — пиковая пропускная способность канала), и, таким образом, максимальный темп выдачи на сообщениях длины L составляет B/L сообщений в секунду. В реальности же большинству сетей на малых сообщениях редко удается приблизиться к этому значению. Причина кроется в накладных расходах на посылку каждого сообщения.

Почему значение темпа выдачи важно с точки зрения эффективности приложения? При низком темпе выдачи ядро процессора вынуждено будет часто простаивать в ожидании возможности отправить готовое сообщение.

Как можно повысить темп выдачи? Самый простой способ — это завести огромный буфер на входе в сеть и накапливать в нем мелкие сообщения, агрегируя их в более крупные. Ведь, как известно, реальная пропускная способность большинства сетей приближается к пиковой на достаточно больших сообщениях. Однако этот способ, как правило, не годится, поскольку увеличивает задержку до неприемлемых значений. Для эффективных параллельных вычислений требуются высокореактивные сети, обладающие как малой задержкой, так и высоким темпом выдачи. Достижение этой цели потребует определенных усилий, даже InfiniBand демонстрирует довольно посредственные значения темпа выдачи на малых сообщениях.

MPI

Для написания многих параллельных программ для кластеров традиционно используется интерфейс MPI [9]. Он неплохо подходит для ситуаций, когда сообщения нужно посылать редко. Однако в некоторых классах задач, таких как вычисления на нерегулярных адаптивных сетках (например, UA из набора тестов NPB [6]), требуется посылать много мелких сообщений подряд. MPI изначально не разрабатывался для эффективной поддержки большого числа посылок, поэтому для таких задач целесообразнее использовать более эффективные в этом плане библиотеки, например, SHMEM [10].

Тем не менее, до сих пор для сравнения различных сетей принято использовать тесты, написанные на MPI. Для демонстрации упускаемых в связи с использованием MPI возможностей, на графиках ниже показаны также результаты выполнения тестов, реализованных на нижнем уровне сети 3D-тор. Мы утверждаем, что используя вместо MPI более подходящий интерфейс, это преимущество может быть сохранено, однако обсуждение этого аспекта выходит за рамки данной работы.

Измерение темпа выдачи сообщений

Для измерения характеристик сети мы использовали тесты [8], входящие в поставку библиотеки ScaliMPI. Тест Ping-Pong измеряет задержку сети, а для измерения темпа выдачи сообщений применялся тест под названием «Ping-Ping», который не имеет отношения к известному тесту Ping-Ping из набора IMB [5] и поэтому здесь фигурирует как MsgRate.

Тест MsgRate устроен следующим образом:

```
MPI_Barrier()
for (i = 0; i < КОЛИЧЕСТВО_ИТЕРАЦИЙ_N; i++)
   if (my_id == 0)
      MPI_SEND(СООБЩЕНИЕ_ДЛИНЫ_L, 1 /* получатель */);
   else if (my_id == 1)
      MPI_RECV(СООБЩЕНИЕ_ДЛИНЫ_L, 0 /* отправитель */);
MPI_Barrier()
```

Здесь один процесс (с номером 0) только отправляет сообщения, а другой (с номером 1) только принимает сообщения. Измеряется общее время выполнение теста. Полученное значение, деленное на количество итераций, — это время выполнения одной посылки, то есть как раз темп выдачи сообщений. Размер сообщения по отношению ко времени выполнения одной итерации — это пропускная способность сети для сообщений данного размера.

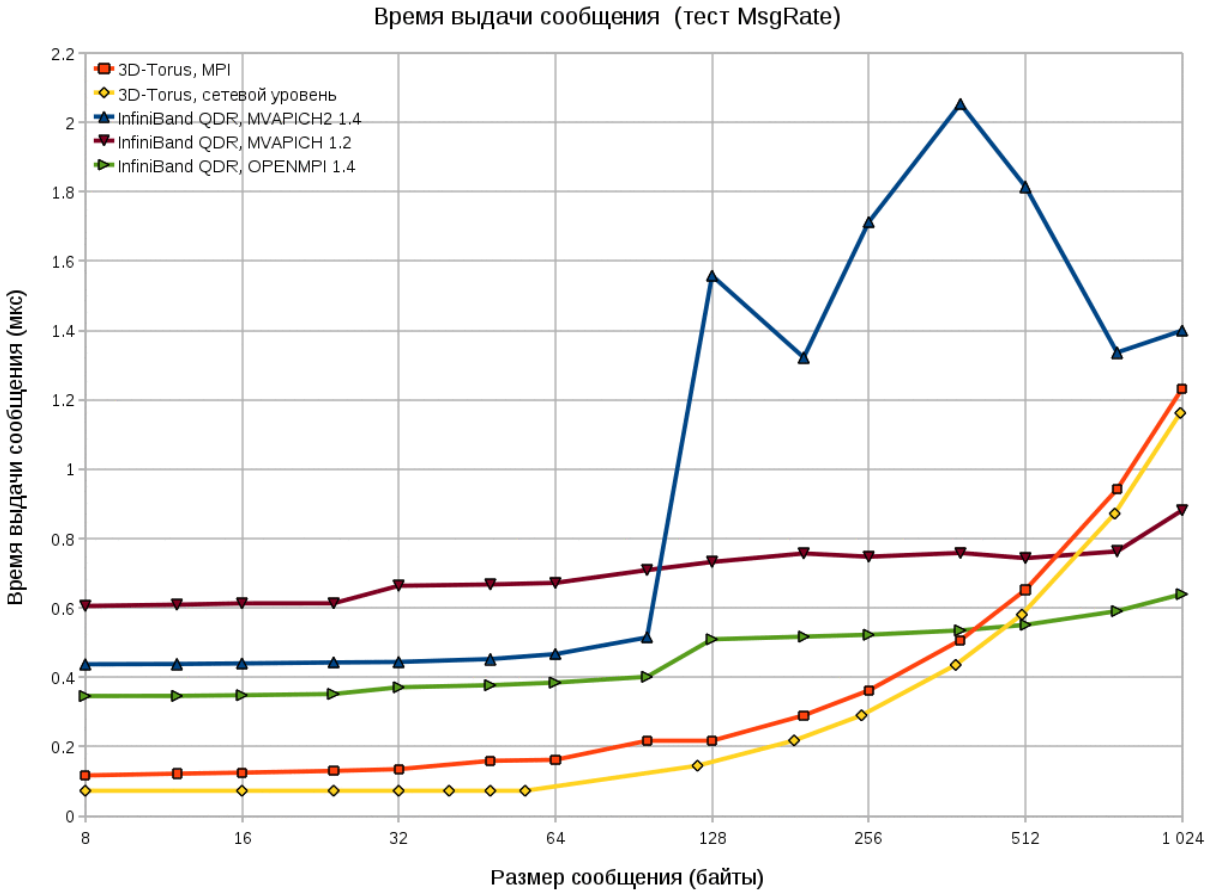

Рис 1. Время выдачи сообщения (меньше — лучше).

Заметим, что если характеристики сети (пропускная способность, темп, задержка) ниже соответствующих характеристик памяти на узлах (а это верно для всех современных суперкомпьютеров), то данный тест фактически превращается в ставший уже классическим тест RandomAccess (GUPS) из HPC Challenge Benchmark [7]. Тест RandomAccess приобретает все большую популярность, так как результаты его выполнения позволяют оценить пригодность суперкомпьютера для решения задач с нерегулярной интенсивной коммуникационной нагрузкой. Таким образом, высокие показатели темпа выдачи коротких сообщений чрезвычайно важны для оценки качества машины.

На приведенных графиках представлены результаты измерений для коммуникационных сетей 3D-тор и InfiniBand QDR суперкомпьютера СКИФ-Аврора. Рис. 1 показывает темп выдачи сообщений (точнее, обратную величину — время выдачи одного сообщения), а рис. 2 — реальную пропускную способность на данном тесте.

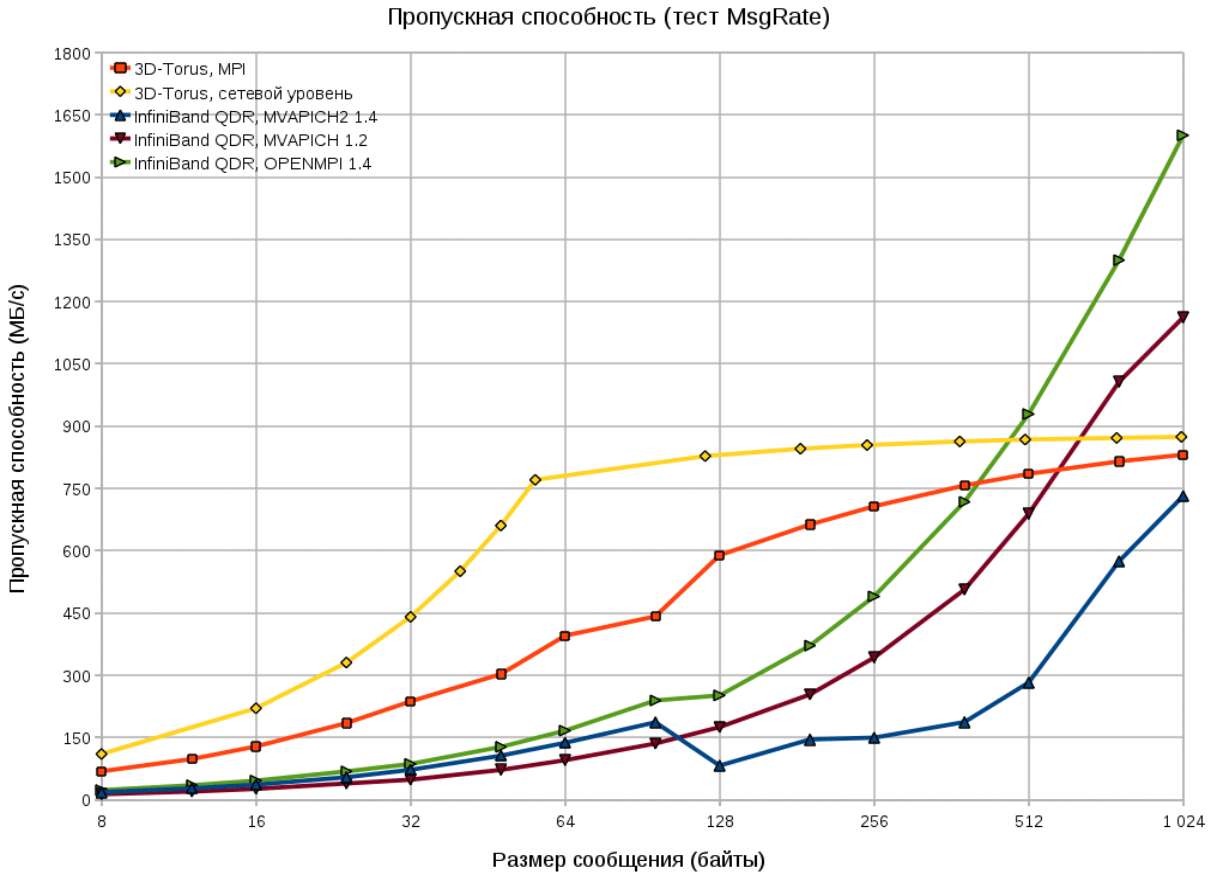

Рис 2. Пропускная способность сети на тесте MsgRate (больше — лучше).

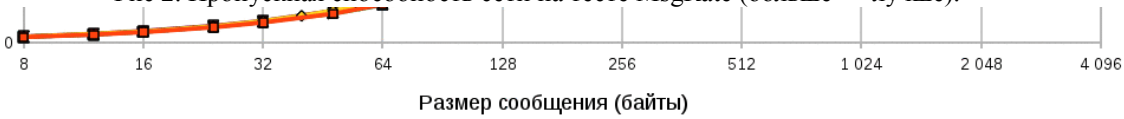

Рис 4. Пропускная способность сети на тесте Ping-Pong (больше — лучше).

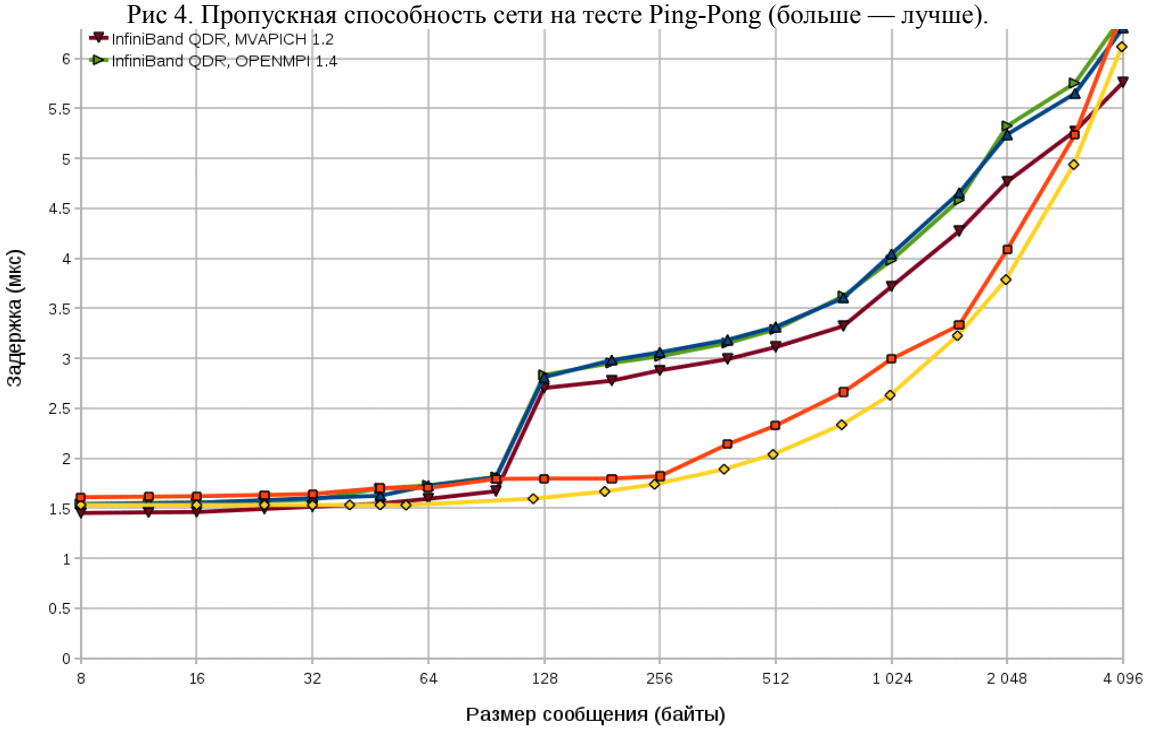

Рис 3. Задержка (меньше — лучше).

Что касается задержки (графики на рис. 3 и 4), то обе сети показывают близкие результаты на коротких сообщениях (8-64 байта). На сообщениях средней длины (128-2048 байтов) сеть 3D-тор заметно лучше.

Результаты тестирования приведены для соединения точка-точка двух узлов суперкомпьютера СКИФ-Аврора. Для сообщений каждого размера тесты проводились в течение достаточного времени для насыщения всех имеющихся входных, выходных и внутрисетевых буферов.

Технические характеристики узла таковы:

- Два 4-х ядерных процессора Intel Xeon CPU X5570 2.93GHz
- Пиковая производительность узла: 93.76 Гфлоп/с
- Интерконнект:
- Сеть 3D-тор, реализованная авторами данной работы в ПЛИС [4], — 6 линков с пиковой пропускной способностью 10 Гбит/с каждый.
- Сеть InfiniBand QDR с пиковой пропускной способностью 40 Гбит/с.

Выводы

Сеть InfiniBand QDR является «ширпотребом» (в хорошем смысле этого слова), то есть универсальным и коммерчески доступным решением, показывающим высокие результаты в большинстве случаев. Однако существует ряд важных задач высокой коммуникационной сложности, на которых традиционные сети, включая InfiniBand, несмотря на высокую пиковую пропускную способность, становятся узким местом суперкомпьютера. Одной из причин снижения эффективности может являться слишком низкий темп выдачи сообщений. В этих случаях рациональней использовать машины на основе специализированных сетей. К такого рода машинам относится суперкомпьютер СКИФ-Аврора, обладающий сетью топологии 3D-тор с исключительно высоким темпом выдачи сообщений и довольно низкой задержкой. Также к этому классу машин можно отнести, к примеру, IBM Blue Gene/P с его сетью топологии 3D-тор (см. результаты масштабирования теста NPB UA для него в [2]) и машину МВС-Экспресс, разработанную в ИПМ им. Келдыша совместно с НИИ «Квант», включающую оригинальную сеть на основе прямой коммутации PCI-Express [3].

Работы выполняются по научно-технической программе Союзного государства «СКИФ-ГРИД» [11], а также при поддержке РФФИ по проекту № 09-07-13598-офи-ц.

ЛИТЕРАТУРА:

1. С.М. Абрамов, В.Ф. Заднепровский, А.Б. Шмелев, А.А. Московский "Супер ЭВМ ряда 4 семейства СКИФ: штурм вершины суперкомпьютерных технологий" // Параллельные вычислительные технологии: Труды Международной научной конференции (30 марта - 3 апреля 2009 г., г. Нижний Новгород). Изд. Нижегородского государственного университета имени Н.И. Лобачевского. С. 5-16.
2. А.А. Корж "Результаты масштабирования бенчмарка NPB UA на тысячи ядер суперкомпьютера Blue Gene/P с помощью PGAS-расширения OpenMP" // Вычислительные методы и программирование. 2010. Том 11. Раздел 2. С. 31-41.
3. А.О. Лацис "Вычислительная система МВС-Экспресс" // URL: http://www.kiam.ru/MVS/research/mvs_express.html (дата обращения: 01.06.2010).
4. А.Ю. Орлов, А.Б. Шворин "О реализации в ПЛИС маршрутизатора высокопроизводительной сети" // Научный сервис в сети Интернет: масштабируемость, параллельность, эффективность: Труды Всероссийской научной конференции (21-26 сентября 2009 г., г. Новороссийск). М.: Изд-во МГУ, 2009. С. 208-210.
5. Набор тестов Intel MPI Benchmarks (IMB) // URL: http://software.intel.com/en-us/articles/intel-mpi-benchmarks/ (дата обращения: 01.06.2010).
6. Набор тестов NAS Parallel Benchmarks (NPB) // URL: http://www.nas.nasa.gov/Resources/Software/npb.html (дата обращения: 01.06.2010).
7. Набор тестов HPC Challenge Benchmark // URL: http://icl.cs.utk.edu/hpcc/ (дата обращения: 01.06.2010).
8. Тест Bandwidth // URL: http://botik.ru/~klimov/bandwidth.tgz (дата обращения: 01.06.2010).
9. Message Passing Interface (MPI) // URL: http://www.mpi-forum.org/ (дата обращения: 01.06.2010).
10. SHMEM application programming interface // URL: http://www.shmem.org/ (дата обращения: 01.06.2010).
11. Суперкомпьютерная программа Союзного государства «СКИФ-ГРИД» (2007-2010 гг.) // URL: http://skif-grid.botik.ru/ (дата обращения: 01.06.2010).