ВЕСТНИК ТОМСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА
2019 Управление, вычислительная техника и информатика № 47

ИНФОРМАТИКА И ПРОГРАММИРОВАНИЕ

УДК 004.75
DOI: 10.17223/19988605/47/11

Е.Н. Перышкова, М.Г. Курносов

УЧЕТ КОНКУРЕНТНОГО РАЗДЕЛЕНИЯ КАНАЛОВ СВЯЗИ ПРИ ФОРМИРОВАНИИ ПОДСИСТЕМ В ВЫЧИСЛИТЕЛЬНЫХ КЛАСТЕРАХ НА БАЗЕ МНОГОПРОЦЕССОРНЫХ УЗЛОВ

Работа выполнена в рамках государственного задания № 0306-2019-0019, а также при частичной финансовой поддержке фонда РФФИ, грант № 18-07-00624.

Выполнена реализация тестовых программ для оценки времени передачи сообщений при разделении каналов связи на уровне стандарта MPI. Проведен экспериментальный анализ падения производительности коммуникационной сети при образовании очередей передачи сообщений для вычислительных систем с SMP/NUMA-архитектурой вычислительных узлов. Разработана система прогнозирования времени выполнения операции All-to-all на заданной подсистеме процессорных ядер при одновременном использовании канала связи множеством процессов.

Ключевые слова: параллельное мультипрограммирование; организация функционирования; вычислительные системы.

Одним из важнейших архитектурных свойств современных вычислительных систем (ВС) с распределенной памятью является глубокая иерархия средств доступа к оперативной памяти процессорных ядер. Коммуникационные сети большинства высокопроизводительных систем списка Top500 имеют как минимум двухуровневую организацию. Первый уровень – коммуникационная сеть связи между элементарными машинами (ЭМ, вычислительными узлами): Cray Gemini, IBM PERCS, Fujitsu Tofu, Gigabit Ethernet, InfiniBand [1–3]; второй уровень – оперативная память, разделяемая процессорными ядрами одной ЭМ. Если принять во внимание использование коммуникационных сетей на базе составных коммутаторов (например, топология fat tree) [4], а также наличие внутрисистемных шин для объединения процессоров в ЭМ с архитектурой NUMA, то количество уровней в иерархической структуре увеличивается. В частности, в системе Sunway TaihuLight пять уровней в коммуникационной среде: оперативная память ядра – Network on Chip – Sunway network – Supernode network – Switch network. Основное назначение коммуникационной сети – реализация передачи сообщений между процессами параллельных программ. На протяжении последних 20 лет доминирующее положение среди средств разработки параллельных программ занимают стандарт MPI и библиотеки, реализующие его (MPICH, MVAPICH, Open MPI).

Топологии коммуникационных сетей, используемых в ВС, по технико-экономическим причинам не являются полносвязными, поэтому при реализации параллельными программами глобальных схем информационных обменов возникает одновременное совместное использование некоторых каналов связи (network contention) [5]. Следствием этого является образование очередей передачи сообщений в библиотеках стандарта MPI, сетевых адаптерах, коммутаторах и падение производительности коммуникационной сети [6]. В данной работе выполнена реализация тестовых программ для оценки времени передачи сообщений при разделении каналов связи на уровне стандарта MPI. Проведен экспериментальный анализ падения производительности коммуникационной сети при образовании очередей передачи сообщений для вычислительных систем с SMP/NUMA-архитектурой вычис-

лительных узлов. Рассмотрено три уровня коммуникационной среды: оперативная память одной ЭМ; внутрисистемная шина, объединяющая процессоры в ЭМ с архитектурой NUMA; сеть связи между ЭМ (InfiniBand и Gigabit Ethernet).

Множество используемых при передаче сообщений между параллельными процессами MPI-программ каналов связи определяется начальным распределением процессов по процессорным ядрам ЭМ системы. Например, на рис. 1, *a* показан пример взаимодействия MPI-процессов, размещенных на двух ядрах одного процессора SMP-узла. В этом случае обмен осуществляется через оперативную память узла. В аналогичной ситуации для NUMA-узлов два процесса выполняют обмен через оперативную память NUMA-узла, на ядрах процессора которого они выполняются. При взаимодействии ядер, размещенных на разных процессорах NUMA-узла, сообщения передаются через внутрисистемную шину, например Intel QuickPath Interconnect (QPI), как показано на рис. 1, *b*. Если взаимодействующие ядра размещены на процессорах, находящихся на разных ЭМ, обмен осуществляется через сетевой адаптер.

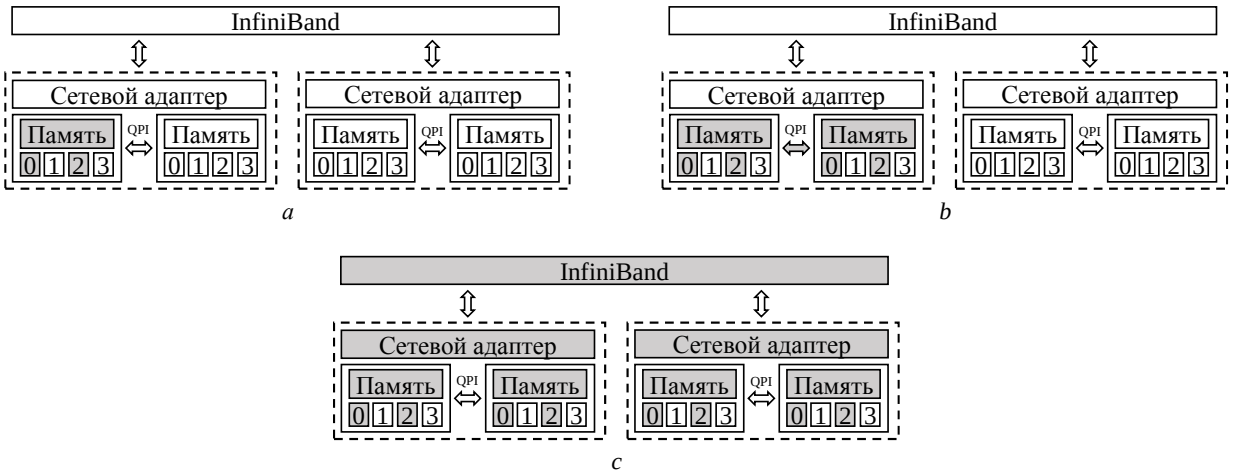


Рис. 1. Возникновение конкуренции за разделяемые ресурсы:
a – контроллер памяти; *b* – шина Intel QPI; *c* – сетевой адаптер
Fig. 1. Shared resources:
a – memory controller; *b* – Intel QPI bus; *c* – network adapter

В системах управления ресурсами ВС возникает задача формирования подсистемы из p процессорных ядер. В ВС на базе многопроцессорных узлов данная задача имеет множество решений. Например, симметричная подсистема ранга 8 может быть сформирована тремя способами: 1 вычислительный узел с 8 процессорными ядрами (1×8), два узла по 4 ядра (2×4) и четыре узла по 2 ядра (4×2). Время выполнения глобальных коммуникационных операций на этих подсистемах будет различным. Поэтому практический интерес представляет разработка алгоритмов формирования подсистем ЭМ, учитывающих структуру информационных обменов целевой программы. Для операции All-to-all выполнено экспериментальное исследование влияние конфигурации подсистемы ЭМ на время выполнения операции. Выбор операции All-to-all обусловлен ее широким распространением в пакетах суперкомпьютерного моделирования. Разработана тестовая программа для оценки времени выполнения коллективной операции All-to-all при различных начальных распределениях процессов по процессорным ядрам ЭМ. Проведено исследование зависимости времени выполнения операции All-to-all от размера передаваемых сообщений и количества процессов, одновременно разделяющих канал связи. Разработана система прогнозирования времени выполнения операции All-to-all на заданной подсистеме ЭМ по результатам предварительной экспериментальной оценки падения производительности операций MPI_Send/Recv при одновременном использовании канала связи множеством процессов. Полученные результаты будут использованы для разработки структурно-ориентированных алгоритмов формирования подсистем ЭМ.

1. Конкурентное использование каналов связи при реализации MPI-программ

Наиболее распространенные типы ЭМ современных высокопроизводительных ВС представлены на рис. 2.

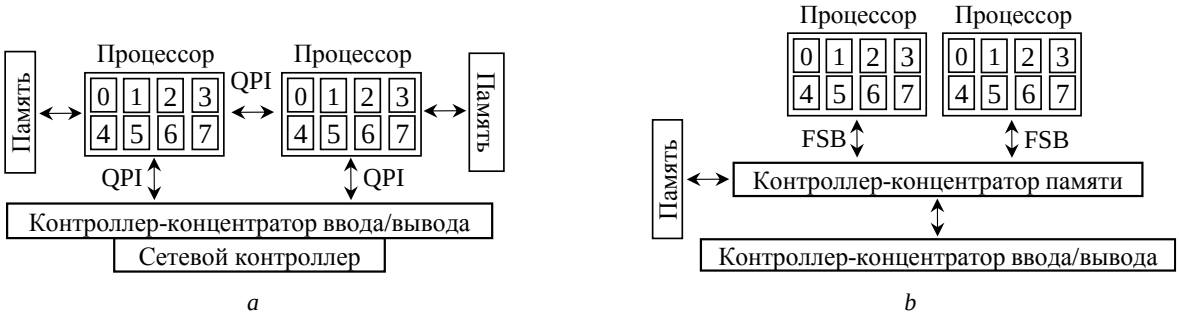


Рис. 2. Типы элементарных машин современных высокопроизводительных ВС:
a – NUMA-узел: два 8-ядерных процессора; *b* – SMP-узел: два 8-ядерных процессора
Fig. 2. Types of architectures computer nodes:
a – NUMA architecture: two 8 cores processers; *b* – SMP architecture: two 8 cores processers

На рис. 2, *a* изображена техническая реализация ЭМ NUMA-узла, включающая два 8-ядерных процессора Intel, объединенных шиной Intel QuickPath Interconnect (QPI). На рис. 2, *b* представлен пример технической реализации ЭМ SMP-узла, состоящей из двух 8-ядерных процессоров Intel, объединенных системной шиной Intel Front Side Bus (FSB). Можно выделить три уровня коммуникационной среды, на которых возникает одновременное совместное использование каналов связи: контроллер памяти, внутрисистемная шина, объединяющая процессоры в ЭМ с архитектурой NUMA, и сетевой контроллер.

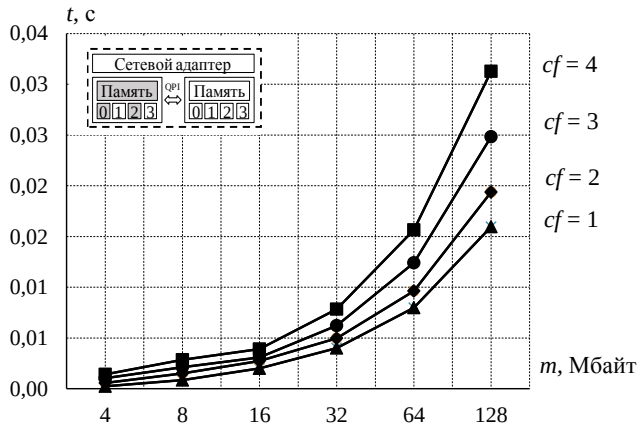
Для определения влияния конкуренции за сетевые ресурсы на время выполнения информационных обменов разработана тестовая MPI-программа, реализующая вызов каждым процессом операции MPI_Recv и MPI_Send. Ниже приведен его псевдокод. Время выполнения операции обмена оценивается путем измерения среднего времени *n* выполнений операций MPI_Recv и MPI_Send в цикле. На каждой итерации цикла реализуется ожидание завершения обменов всех ветвей параллельной программы. За время *t* выполнения информационного обмена принимается среднее время одного запуска.

```
function Benchmark(sbuf, rbuf, size, nruns)
   Irecv(rbuf, size, reqarray[0])              /* Инициализация */
   Isend(sbuf, size, reqarray[1])
   Waitall(2, reqarray)
   for i = 1 to nruns do
      t -= wtime()
      Irecv(rbuf, size, reqarray[0])
      Isend(sbuf, size, reqarray[1])
      Waitall(2, reqarray)
      t += wtime()
   end for
   t = t / (2 * nruns)
end function
```

Экспериментальная часть работы выполнена на вычислительных кластерах с NUMA/SMP-узлами. Кластер с NUMA-узлами укомплектован 6 вычислительными серверами на базе платформы Intel S5520UR. На каждом узле размещено два процессора Intel Xeon E5620, оперативная память – 24 Гбайт (DDR3), сетевой адаптер InfiniBand QDR Mellanox MT26428. Кластер с SMP-узлами уком-

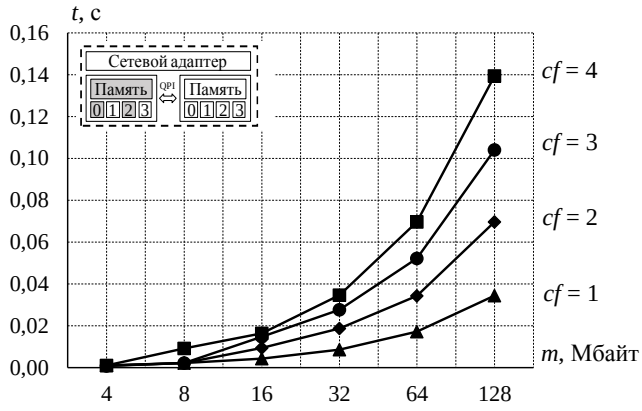
плектован 18 вычислительными серверами на базе платформы Intel SR2520SAF. На каждом узле размещено два процессора Intel Xeon E5420, оперативная память – 8 Гбайт (DDR2), сетевой адаптер Gigabit Ethernet Intel PRO/1000 EB. Вычислительные кластера функционируют под управлением операционной системы (ОС) GNU/Linux. При компиляции MPI-программы использовались коммуникационные библиотеки MPICH 3.2.1 и MVAPICH2 2.2.

Тестовая программа запускалась с разным количеством процессов для передачи информационных сообщений размером *m* Мбайт. Каждый процесс привязывался к выделенному процессорному ядру при помощи подсистемы numactl. На рис. 3–5 показаны зависимости *t*(*m*, *cf*) времени передачи сообщения размером *m* байт от количества *cf* (contention factor) процессов, одновременно разделяющих общий канал связи. В экспериментах рассматривалось три уровня коммуникационной среды: оперативная память NUMA/SMP узлов (рис. 3, *a*, *b*), внутрисистемная шина Intel QPI, объединяющая процессоры NUMA-узлов (рис. 4) и сеть связи между ЭМ (адаптеры InfiniBand QDR на рис. 5, *a* и Gigabit Ethernet на рис. 5, *b*).



m, Мбайт	Коэффициент падения производительности *t*(*m*, *cf*)/*t*(*m*, 1)			
	cf = 1	*cf* = 2	*cf* = 3	*cf* = 4
128	1	1,21	1,55	1,96
64	1	1,20	1,55	1,96
32	1	1,25	1,56	1,96
16	1	1,36	1,55	1,95
8	1	1,75	2,47	3,31
4	1	2,48	4,42	6,05

a



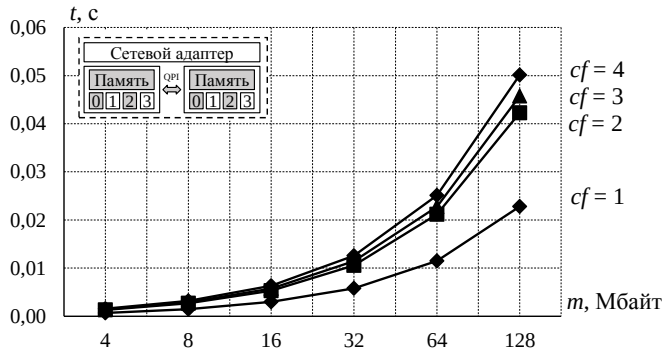
m, Мбайт	Коэффициент падения производительности *t*(*m*, *cf*)/*t*(*m*, 1)			
	cf = 1	*cf* = 2	*cf* = 3	*cf* = 4
128	1	2,03	3,03	4,06
64	1	2,00	3,04	4,06
32	1	2,17	3,23	4,03
16	1	2,17	3,43	3,80
8	1	1,01	1,01	4,27
4	1	0,86	0,82	0,96

b

Рис. 3. Измерение времени *t* передачи сообщения *m* и количества *cf* процессов MPI-программы, разделяющих общий канал связи: *a* – контроллер памяти NUMA-узла; *b* – контроллер памяти SMP-узла

Fig. 3. Measured transmission time of *m*-byte message and the number *cf* of processes sharing the common communication channel: *a* – NUMA node memory controller; *b* – SMP memory controller

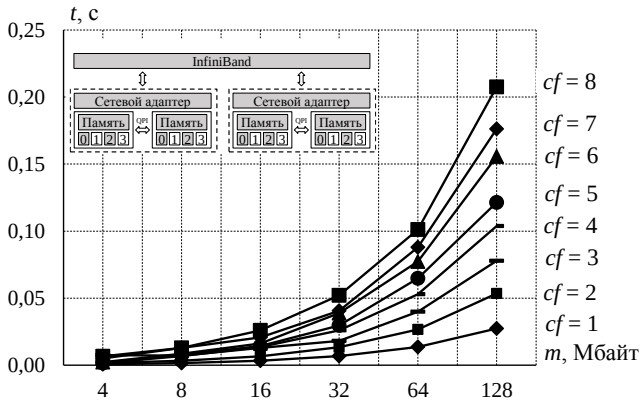
В таблицах серым фоном отмечены установленные комбинации размеров сообщений и числа процессов *cf*, при которых наблюдается резкое падение производительности канала связи (более *cf* раз). Такие значения *m* и *cf* могут быть использованы для определения оптимального числа процессов, запускаемых на одном вычислительном узле ВС при формировании подсистем ЭМ с учетом структуры информационных обменов целевой программы.



m, Мбайт	Коэффициент падения производительности $t(m, cf)/t(m, 1)$			
	cf = 1	*cf* = 2	*cf* = 3	*cf* = 4
128	1	1,85	2,01	2,20
64	1	1,85	1,98	2,19
32	1	1,82	1,97	2,16
16	1	1,78	1,91	2,12
8	1	1,83	1,96	2,16
4	1	1,99	2,09	2,38

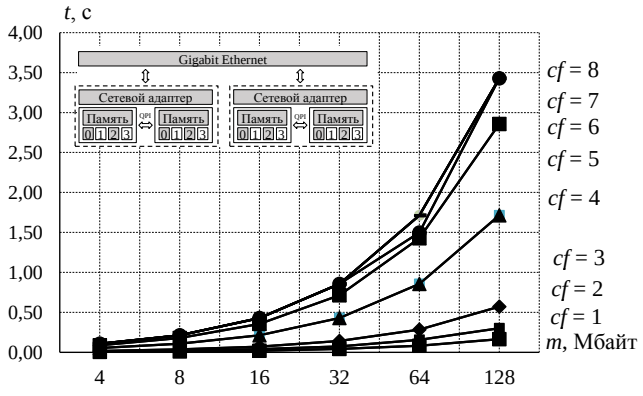
Рис. 4. Измерение времени *t* передачи сообщения *m* и количества *cf* процессов MPI-программы, разделяющих внутрисистемную шину Intel QPI

Fig. 4. Measured transmission time of *m*-byte message and the number *cf* of processes sharing the processor interconnect Intel QPI in NUMA node



m, Мбайт	Коэффициент падения производительности $t(m, cf)/t(m, 1)$							
	1	2	3	4	5	6	7	8
128	1	2,0	2,8	3,8	4,4	5,7	6,4	7,6
64	1	2,0	2,9	3,9	4,8	5,7	6,5	7,4
32	1	2,0	2,7	3,8	4,4	5,7	6,0	7,7
16	1	2,0	3,8	4,0	4,2	4,8	6,1	7,6
8	1	2,0	4,3	4,1	4,7	5,0	7,6	7,6
4	1	2,1	9,8	2,2	2,1	3,0	6,5	8,5

a



m, Мбайт	Коэффициент падения производительности $t(m, cf)/t(m, 1)$							
	1	2	3	4	5	6	7	8
128	1	0,3	3,0	0,5	5,0	6,0	6,0	6,0
64	1	0,3	3,0	0,5	5,0	5,2	6,0	6,0
32	1	0,3	3,0	0,5	5,0	6,0	6,0	6,0
16	1	0,3	3,0	0,6	5,0	6,0	6,0	6,0
8	1	0,3	3,0	0,6	5,0	6,0	6,0	5,9
4	1	0,3	3,0	0,5	4,9	5,9	5,9	5,9

b

Рис. 5. Измерение времени *t* передачи сообщения *m* и количества *cf* процессов MPI-программы, разделяющих общий канал связи: *a* – сетевой контроллер InfiniBand; *b* – сетевой контроллер Gigabit Ethernet

Fig. 5. Measured transmission time of *m*-byte message and the number *cf* of processes sharing the common communication channel: *a* – InfiniBand QDR; *b* – Gigabit Ethernet

2. Формирование подсистемы ЭМ с учетом деградации каналов связи

В системах управления ресурсами ВС возникает задача формирования подсистемы из *p* процессорных ядер. В ВС на базе многопроцессорных узлов данная задача имеет множество решений. Известные системы управления ресурсами IBM LoadLeveler, Altair PBS Pro, SLURM, TORQUE используют различные модели и методы управления очередями задачи и формирования подсистем

ВУ [7]. Широко используются методы приоритетного обслуживания задач, алгоритмы внеочередного выполнения задач (backfiling), приоритетное обслуживание с вытеснением задач (job preemption). Рассмотрим несколько эвристических алгоритмов формирования подсистем ЭМ: формирование подсистемы из минимального числа ЭМ («жадный» алгоритм), формирование подсистемы из максимально возможного числа ЭМ. Время выполнения глобальных коммуникационных операций на сформированных подсистемах будет различным. Существующие СУР не учитывают возможного падения производительности сетевой подсистемы при одновременном использовании ее компонентов параллельными процессами, поэтому практический интерес представляет разработка алгоритмов формирования подсистем ЭМ, учитывающих структуру информационных обменов целевой программы.

Авторами разработана система прогнозирования времени выполнения коллективной операции All-to-all на заданной подсистеме ЭМ. Функциональная структура системы прогнозирования представлена на рис. 6.

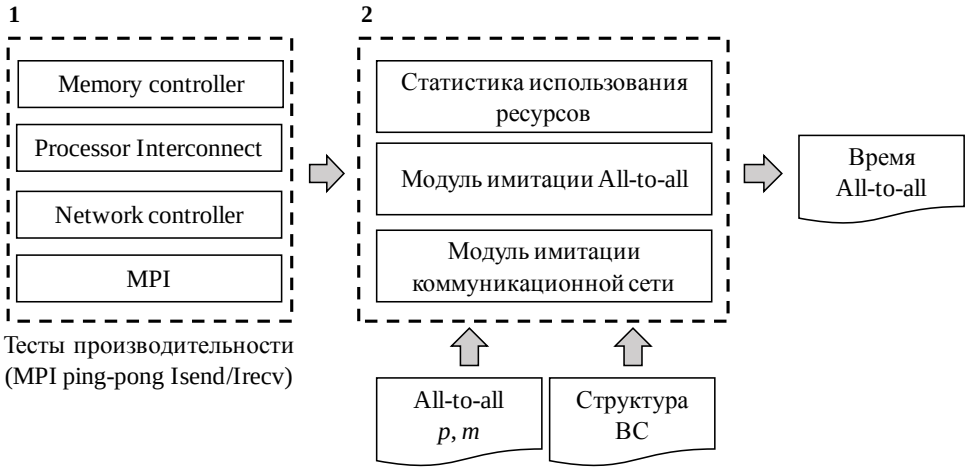


Рис. 6. Функциональная структура системы прогнозирования
Fig. 6. Prediction system

Модуль 1 служит для тестирования производительности подсистемы ЭМ при различном количестве одновременных взаимодействий по каналу связи. Модуль реализован в виде параллельной программы в стандарте MPI и основан на обменах с использованием операций MPI_Send и MPI_Recv. В состав модуля 1 входит подсистема запуска тестов с различным уровнем *cf* одновременного использования заданного канал связи. Результатом работы модуля являются таблицы с оценкой времени передачи сообщений для различных значения *cf* и размеров *m* сообщений. Построенные таблицы в дальнейшем используются для динамического построения оценок времени реализации алгоритмов.

Модуль 2 состоит из трех блоков: модуль сбора статистики одновременного использования ресурсов, модуль имитации блочного алгоритма All-to-all и модуль имитации иерархической коммуникационной сети. Модуль сбора статистики предназначен для подсчета числа *cf* одновременных использований каналов связи при реализации одного шага конкретного алгоритма (шаблона) информационных обменов параллельной программы (например, All-to-all, One-to-all, All-to-one). В текущей версии реализован модуль имитации блочного алгоритма коллективной операции All-to-all, которая наиболее часто встречается в параллельных программах, требовательных к производительности коммуникационной сети, например в MPI-реализациях алгоритмов на графах (Graph500), реализациях быстрого преобразования Фурье (HPCC FFT) и методах параллельного решения из первых принципов (ab initio) задач квантовой химии (Quantum Espresso). Ниже приведен его псевдокод. Каждый процесс выполняет *p* операций передачи и приема сообщений, где *p* – число процессов в программе. При этом операции send/recv группируются в блоки из *block* коммуникационных операций для сокращения накладных расходов. Входными параметрами модуля являются размер *m* передаваемого сообщения и требуемое количество *p* процессов.

```
function AllToAll(sbuf, scount, rbuf, rcount, block)
   for ii = 0 to p do
      ss = block
      if p – ii < block then
          ss = p - ii
      for i = 0 to ss do
         dst = (rank + i + ii) % p
         Irecv(rbuf + dst * rcount, rcount, dst, reqarray[i])
      end for
      for i = 0 to ss do
         dst = (rank – i – ii + p) % p
         Isend(sbuf + dst * scount, scount, dst, reqarray[i + ss])
      end for
      Waitall(2 * ss, reqarray)                    /* Запуск 2 * ss  опе-
раций */
      ii = ii + block
   end for
end function
```

Модуль имитации иерархической коммуникационной среды логически реализует коммуникационные уровни ВС, задает нумерацию ЭМ и распределение процессов программы по ним (cpu affinity). Результатом работы системы моделирования является оценка времени выполнения коллективной операции All-to-all на заданной конфигурации подсистемы ЭМ. После получения оценки времени выполнения коллективной операции All-to-all для различных конфигураций подсистем ЭМ одного ранга устанавливается отношение порядка выбора подсистемы ЭМ исходя из минимума времени реализации информационных обменов. Система моделирования может быть дополнена шаблонами информационных обменов параллельной программы в стандарте MPI. В таблице представлены результаты работы системы моделирования и время выполнения операции All-to-all в зависимости от выбора подсистемы ВС. Размер передаваемого сообщения равен 1 Мбайт.

Время выполнения операции All-to-all (MVAPICH, InfiniBand QDR) и оценка времени системой прогнозирования

Ранг подсистемы	**Время выполнения операции All-to-all, с**		
2	**1 ВУ, 2 ядра**	**2 ВУ, 1 ядро**	
Прогноз (с)	0,00016	0,00033	
Экспериментальный запуск (с)	0,00044	0,00061	
Установленный порядок	1	2	
4	**1 ВУ, 4 ядра**	**2 ВУ, 2 ядра**	**4 ВУ, 1 ядро**
Прогноз (с)	0,0019	0,0021	0,0018
Экспериментальный запуск (с)	0,0031	0,0036	0,0029
Установленный порядок	2	3	1
8	**1 ВУ, 8 ядер**	**2 ВУ, 4 ядер**	**4 ВУ, 2 ядер**
Прогноз (с)	0,00384	0,19	0,0058
Экспериментальный запуск (с)	0,00754	0,09	0,0076
Установленный порядок	1	3	2

Заметим, что система моделирования не дает точного совпадения времени с результатами выполнения All-to-all на реальной ВС. Последнее обусловлено простотой модели и наличием асинхронных событий в ЭМ ВС. Важно отметить, что точного совпадения не требуется, основная задача – установить отношение порядка на множестве подсистем одного ранга. Из таблицы видно, что для рассмотренного примера данная задача решается успешно.

Заключение

В данной работе выполнена реализация тестовых программ для оценки времени передачи сообщений при разделении каналов связи на уровне стандарта MPI. Проведен экспериментальный анализ падения производительности коммуникационной сети при образовании очередей передачи сообщений для вычислительных систем с SMP/NUMA-архитектурой вычислительных узлов. Рассмотрено три уровня коммуникационной среды: оперативная память одной ЭМ, внутрисистемная шина, объединяющая процессоры в ЭМ с архитектурой NUMA, сеть связи между ЭМ (InfiniBand и Gigabit Ethernet).

Разработана система прогнозирования времени выполнения операции All-to-all на заданной подсистеме ЭМ по результатам предварительной экспериментальной оценки падения производительности операций MPI_Send / MPI_Recv при одновременном использовании канала связи множеством процессов. Полученные результаты будут использованы для разработки структурно-ориентированных алгоритмов формирования подсистем ЭМ.

ЛИТЕРАТУРА

1. Alverson R., Roweth D., Kaplan L. The Gemini System Interconnect // Proc. 18th IEEE Symposium on High Performance Interconnects. Washington, DC : IEEE Press, 2010. P. 83–87.
2. Chen D., Eisley N.A., Heidelberger P., Senger R. et al. The IBM Blue Gene/Q interconnection network and message unit // Proc. 2011 International Conference for High Performance Computing, Networking, Storage and Analysis. New York : ACM Press, 2011. DOI: 10.1145/2063384.2063419.
3. Ajima Y., Inoue T., Hiramoto S., Shimizu T., Takagi Y. The tofu interconnect // IEEE Micro 32(1). 2012. P. 21–31.
4. Корнеев В.В. Вычислительные системы. М. : Гелиос АРВ, 2004. 512 с.
5. Prisacari B., Rodriguez G., Minkenberg C., Hoefler T. Bandwidth-optimal all-to-all exchanges in fat tree networks // Proc. 27th international ACM conference on International conference on supercomputing, June 10–14, Eugene, Oregon,. USA. 2013.
6. Steffenel L.A. Modeling network contention effects on all-to-all operations. IEEE Press, 2006.
7. Hovestadt M., Kao O., Keller A., Streit A. Scheduling in HPC resource management systems: Queuing vs. Planning // Proc. 9th Workshop on Job Scheduling Strategies for Parallel Processing (JSSPP), LNCS #2862, 2003. P. 1–20.

Поступила в редакцию 10 мая 2018 г.

Peryshkova E.N., Kurnosov M.G. (2018) MODELING NETWORK CONTENTION EFFECTS ON PROCESS ALLOCATION IN COMPUTER SYSTEMS. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie vychislitelnaja tehnika i informatika* [Tomsk State University Journal of Control and Computer Science]. 47. pp. 93–101

DOI: 10.17223/19988605/47/11

Interconnection networks of modern high-performance distributed computer systems have at least a two-level hierarchical organization. The first level of the communication network is formed by the switch-based network (InfiniBand, Ethernet). The second level is represented by a shared memory of SMP/NUMA-computer nodes. In such systems a communication time between processors depends on their replacement in the system.

In this paper, we present a benchmark for estimating the message passing time when MPI-processes share the communication channels. We analyze the degradation of the communication network performance when message passing queues are formed for computer systems with NUMA/SMP computer nodes. We consider three levels of communication environment: shared memory of a computer node, processor interconnect in NUMA nodes, network interconnect between nodes (InfiniBand and Gigabit Ethernet).

The Resource and Jobs Management Systems (RJMS) form a subsystem of *p* processor cores. If computer systems consist of multiprocessor nodes, this problem has many solutions. For example, a symmetric set of nodes that has the rank equal to eight can be formed in three ways: one computational node with eight processor cores (1x8), two nodes with four cores (2x4) and four nodes with two cores (4x2). A completion time of collective communication operations on these subsystems will be different. Therefore, the development of algorithms that determine nodes allocation taking into account a message passing structure of the target program has practical interest.

Authors have developed a software for predicting the execution time of the All-to-all operation on the given subsystem of nodes. A software uses the results of an experimental estimate of the performance degradation for the MPI_Send/MPI_Recv operations during simultaneous use of the communication channel by a set of processes.

Keywords: collective communications; network contention; computer clusters.

PERYSHKOVA Eugene Nikolaevna (Senior teacher, Siberian State University of Telecommunications and Information; Engineer, Rzhanov Institute of Semiconductor Physics of SB RAS, Novosibirsk, Russian Federation).
E-mail: e.peryshkova@gmail.com

KURNOSOV Mikhail Georgievich (Doctor of Technical Sciences, Professor, Siberian State University of Telecommunications and Information Sciences; Senior Research Scientist, Rzhanov Institute of Semiconductor Physics of SB RAS, Novosibirsk, Russian Federation).
E-mail: mkurnosov@gmail.com

REFERENCES

1. Alverson, R., Roweth, D. & Kaplan, L. (2010) The gemini system interconnect. *Proc. 18th IEEE Symposium on High Performance Interconnects.* Washington, DC: IEEE Press. pp. 83–87. DOI: 10.1109/HOTI.2010.23
2. Chen, D., Eisley, N.A., Heidelberger, P., Senger, R. et al. (2011) The IBM Blue Gene/Q interconnection network and message unit. *Proc. 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*. New York: ACM Press. pp. 1–2. DOI 10.1145/2063384.2063419
3. Ajima, Y., Inoue, T., Hiramoto, S., Shimizu, T. & Takagi, Y. (2012) The tofu interconnect. *IEEE Micro 32(1)*. pp. 21–31. DOI: 10.1109/MM.2011.98
4. Korneev, V. (2004) *Vychislitel'nye sistemy* [Computer systems]. Moscow: Gelios ARB.
5. Prisacari, B., Rodriguez, G., Minkenberg, C. & Hoefler, T. (2013) Bandwidth-optimal all-to-all exchanges in fat tree networks. *Proc. 27th international ACM conference on International conference on supercomputing*. New York: ACM Press. pp. 139–148. DOI: 10.1145/2464996.2465434
6. Luiz, A.S. (2006) Modeling network contention effects on all-to-all operations. *IEEE Press.*
7. Hovestadt, M., Kao, O., Keller, A. & Streit, A. (2003) Scheduling in HPC resource management systems: Queuing vs. Planning. *Proc. 9th Workshop on Job Scheduling Strategies for Parallel Processing (JSSPP)*. pp. 1–20. DOI: 10.1007/10968987_1