7universum.com
UNIVERSUM:
ТЕХНИЧЕСКИЕ НАУКИ

ИССЛЕДОВАНИЕ ОСОБЕННОСТЕЙ АРХИТЕКТУРЫ МНОГОПРОЦЕССОРНЫХ СИСТЕМ, ВЛИЯЮЩИХ НА ЭФФЕКТИВНОСТЬ ПАРАЛЛЕЛЬНЫХ ПРОГРАММ

Голенков Евгений Александрович

канд. физ.-мат. наук, Институт автоматики и процессов управления Дальневосточного отделения Российской академии наук, 690041, Россия, г. Владивосток, ул. Радио, д. 5
E-mail: golenkov@dvo.ru

Леонтьев Денис Васильевич

аспирант, Институт автоматики и процессов управления Дальневосточного отделения Российской академии наук, 690041, Россия, г. Владивосток, ул. Радио, д. 5
E-mail: devozh@dvo.ru

Парахин Роман Валерьевич

инженер-программист, Институт автоматики и процессов управления Дальневосточного отделения Российской академии наук, 690041, Россия, г. Владивосток, ул. Радио, д. 5
E-mail: fadak@dvo.ru

Тарасов Георгий Витальевич

научный сотрудник, Институт автоматики и процессов управления Дальневосточного отделения Российской академии наук, 690041, Россия, г. Владивосток, ул. Радио, д. 5
E-mail: george@dvo.ru

Харитонов Дмитрий Иванович

канд. техн. наук, Институт автоматики и процессов управления Дальневосточного отделения Российской академии наук, 690041, Россия, г. Владивосток, ул. Радио, д. 5
E-mail: demiurg@dvo.ru

Исследование особенностей архитектуры многопроцессорных систем, влияющих на эффективность параллельных программ //
Universum: Технические науки : электрон. научн. журн. Голенков Е.А. [и др.]. 2015. № 11 (22) .
URL: http://7universum.com/ru/tech/archive/item/2762

THE STUDY OF MULTIPROCESSOR SYSTEMS ARCHITECTURE FEATURES, AFFECTING THE EFFICIENCY OF PARALLEL PROGRAMS

Evgenij Golenkov
Candidate of Physico-Mathematical Sciences, Institute of Automation and Control Processes, Far Eastern Branch Russian Academy of Sciences, 690041, Russia, Vladivostok, Radio str., 5

Denis Leontyev
Postgraduate student, Institute of Automation and Control Processes, Far Eastern Branch Russian Academy of Sciences, 690041, Russia, Vladivostok, Radio str., 5

Roman Parakhin
Software Engineer, Institute of Automation and Control Processes, Far Eastern Branch Russian Academy of Sciences, 690041, Russia, Vladivostok, Radio str., 5

Georgiy Tarasov
Research scientist, Institute of Automation and Control Processes, Far Eastern Branch Russian Academy of Sciences, 690041, Russia, Vladivostok, Radio str., 5

Dmitry Kharitonov
Candidate of Engineering Sciences, Institute of Automation and Control Processes, Far Eastern Branch Russian Academy of Sciences, 690041, Russia, Vladivostok, Radio str., 5

АННОТАЦИЯ

В работе описан программный комплекс исследования производительности как многопроцессорных вычислительных систем в целом, так и отдельных ее подсистем, обладающих различными аппаратными и программными решениями, от эффективной работы которых зависит итоговая эффективность выполнения параллельных программ. Описывается архитектура программного комплекса, общий алгоритм тестирования и структура разработанных тестов. Основой функциональности разработанного программного комплекса является набор компонент, реализующих измерение скорости работы участков кода исследуемой программы, связанных с работой отдельных подсистем многопроцессорной вычислительной системы (передача данных по сети, доступ к оперативной памяти, ввод/вывод файлов и многое другое). При анализе результатов

производительности учитываются статистические характеристики, показывающие достоверность измерений. Приводятся результаты исследования особенностей архитектуры многопроцессорных вычислительных систем, влияющих на эффективность работы параллельных программ. Подробно рассмотрены пять различных тестов. В тесте межпроцессного взаимодействия замеряется время выполнения заданной pthread-функции при определенном общем уровне загрузки вычислительной системы, формируемой созданием группы потоков. В тесте однопоточной записи в память, оценивается производительность работы памяти при последовательной записи одним потоком в общий массив памяти. В тесте многопоточной записи в память измеряется производительность при выполнении последовательной записи произвольных данных в память блоками по 500 МБ заданным количеством потоков. В тесте записи в память NUMA узлов оценивается производительность передачи данных между памятью различных NUMA-узлов. В тесте скорости передачи MPI-сообщений сравнивается производительность передачи сообщений между MPI-процессами на одном вычислительном узле и на разных узлах.

ABSTRACT

The article describes a software system for performance analysis of multiprocessor systems as a whole and its individual sub-systems with different hardware and software solutions, the effective operation of which affects the total efficiency of parallel programs. The software package architecture, general testing algorithm and the structure of the developed tests are described. Functionality of the developed software based on a set of components that implements code sections speed measurement of the programs in hand related to the work of multiprocessor computer system particular subsystems (network data transfer, memory access, file input / output and so on). Statistical characteristics, showing the reliability of the measurements, are taken into account when analyzing performance. The results of studies of the multiprocessor systems architecture peculiarities, affecting the efficiency of the parallel programs, are given. Five different tests are considered in details. The test of IPC is measuring the given pthread-function execution time at a certain level of computer system overall load, formed by creating a

group of threads. In the test of a single-threaded memory write, performance of memory is estimated when writing into a common memory array by a single thread. In a multi-threaded memory test, the performance of arbitrary data sequential write to memory in blocks of 500 MB by given number of threads is measured. In the test of NUMA nodes memory write, the performance of data transfer between different NUMA-nodes is evaluated. The MPI-messaging speed test compares the performance of MPI message passing between processes on the same computing node or on different nodes.

Ключевые слова: архитектура вычислительных систем, неоднородный доступ к памяти, межпроцессное взаимодействие, сетевая инфраструктура, производительность вычислительных систем.

Keywords: computing systems architecture, non-uniform memory access, interprocess communications, network infrastructure, computer systems performance.

Введение

Существует огромное количество средств для оценки производительности компьютерных систем. Некоторые тесты измеряют время выполнения часто встречающихся алгоритмов, другие оценивают выполнение задач, специфических для конкретной прикладной области, третьи формируют собственный синтетический показатель, основанный на измерении производительности различных компьютерных подсистем. Единицей измерения производительности суперкомпьютеров и многопроцессорных вычислительных систем считается количество операций с плавающей точкой в секунду (FLOPS) [1]. На основе этой единицы измерения составляется мировой рейтинг суперкомпьютеров Top500 и рейтинг Российских суперкомпьютеров Top50. Однако в настоящее время архитектура многопроцессорных вычислительных систем настолько разнообразна и сложна, что при программировании новых алгоритмов или распараллеливании существующих программ необходимо учитывать значительное число факторов, не описываемых показателем количества операций в секунду. Игнорирование частных архитектурных особенностей, заложенных разработчиками

в вычислительную систему, может привести к значительному падению производительности и эффективности прикладной программы. Например, в последние годы рост производительности вычислительных систем достигается наращиванием количества вычислительных ядер. Поэтому в современных вычислительных узлах подсистема оперативной памяти имеет сложную многоуровневую архитектуру, где разница между максимальным и минимальным временем доступа может составлять десятки и сотни тактов. Так, доступ к регистрам обычно происходит за 1 такт работы процессора, доступ к микросхемам памяти по разным оценкам составляет 50–100 тактов [2; 3]. Кроме того, современные вычислительные кластеры создаются из большого количества отдельных узлов, объединенных высокоскоростной сетью, позволяющей объединять от десятков до сотен тысяч вычислительных узлов. Причём разница в скорости передачи данных между узлами также может достигать нескольких раз. В результате для получения эффективной параллельной программы желательно учитывать все технические особенности вычислительной архитектуры целевой вычислительной системы. Лучшим способом оценить влияние таких особенностей на параллельную программу является получение объективных численных показателей, при помощи тестов, рассчитанных на выявление отдельных специфических деталей функционирования вычислительной системы.

Общий алгоритм тестирования

В основе общего алгоритма тестирования производительности различных архитектурных решений многопроцессорных вычислительных систем лежит следующая простая идея. Для любой архитектурной особенности системы, влияющей на производительность вычислительных программ, можно построить и запрограммировать некоторую функцию, время выполнения которой будет определяться влиянием этой особенности. Если перед вызовом этой функции и по окончании ее работы зафиксировать точное значение времени (астрономическое или относительное), то, сравнивая разницу этих значений для различных систем, можно будет определить влияние контрольной архитектурной особенности на производительность вычислительной задачи. Однако даже если

оценивается время выполнения абсолютно детерминированной функции, на практике могут получаться довольно разные результаты. Это происходит потому, что измерения проводятся в условиях реальной работы операционной системы с учетом непредсказуемых задержек, связанных с работой планировщика процессов и внутреннего состояния операционной системы – объёма задействованной памяти, размера используемого кэша файловой системы, размера выделенных таблиц дескрипторов и т. д. Одна часть непредсказуемых задержек является величиной случайной, в то время как другая появляется вследствие работы тестируемой функции. Так, если заниматься множественным выделением памяти, то рано или поздно операционной системе придётся оптимизировать карту выделения памяти, вызывая «внезапную» задержку в приложениях. Таким образом, одного единственного точного значения времени выполнения тестовой функции не существует, и, чтобы корректно учесть разброс в результатах, необходимо использовать блок статистического анализа для оценки достоверности полученных измерений. Предложено использовать следующие три критерия оценки достоверности:

1. Критерий Пирсона (Хи-квадрат), подразумевающий, что разброс в оценке измеряемой величины является случайной величиной, подчиняющейся распределению Гаусса.

2. «Кучность измерений». Разброс результатов измерений не превышает двух процентов значения измеряемой величины. Это говорит о том, что разброс в оценке измеряемой величины является, скорее всего, случайной величиной с неизвестным распределением. Если разброс времени больше, то применяется третий критерий.

3. На гистограмме частот можно выделить интервал, в котором 80 % измерений удовлетворяют критерию Пирсона. Этот критерий подразумевает, что разброс в оценке измеряемой величины имеет случайную систематическую составляющую.

Для получения статистической выборки необходимо выполнять измерения в различных режимах работы вычислительной системы с различными

нагрузками на процессор. Это позволит эмулировать работу операционной системы и вычислительных приложений в реальных условиях.

Таким образом, общий алгоритм измерения времени организован следующим образом. Задается некоторое начальное число итераций выполнения измеряемой функции и объявляется первый раунд измерений, который состоит из цикла запусков контрольной функции. По окончании раунда производится статистический анализ результатов измерений в соответствии с описанными выше критериями. Если по одному из трех критериев будет признано, что результаты измерений положительны, то процедура прекращается, и среднее значение за все итерации текущего раунда объявляется результатом измерений. В противном случае запускается следующий раунд, который состоит в увеличении числа итераций в два раза. Алгоритмически задано, что может происходить не больше шести раундов измерений. Если за все шесть раундов не удается добиться удовлетворения условий какого-либо критерия, следует повторить измерения сначала.

Для ускорения разработки различных тестов создана библиотека поддержки средств тестирования, реализующая общий алгоритм тестирования, описанный выше. Эта библиотека отвечает за первичный разбор параметров, передаваемых приложению, за измерение времени выполнения контрольной функции, обработку статистической информации и за вывод результатов тестирования. После разработки нового теста он регистрируется в основной системе, обеспечивающей единую оболочку для запуска и использования комплекса программ по тестированию.

В качестве тестовой платформы использовались ресурсы центра коллективного пользования «Дальневосточный вычислительный ресурс» ДВО РАН (ЦКП «ДВВР» ДВО РАН). В основном использовалось следующие три типа узлов:

1. Стандартный расчетный узел (AMD). Имеет четыре 12-ядерных процессора AMD Opteron 6164HE с частотой 1.7 ГГц (кэш L1 = 12 x 128 КБ (64+64), L2 = 12 x 512 КБ, L3 = 2 x 6 МБ) и 64 Гб оперативной памяти.

2. Расширенный расчетный узел (AMD). Имеет четыре 12-ядерных процессора AMD Opteron 6174 с частотой 2.2 ГГц (кэш L1 = 12 x 128 КБ (64+64), L2 = 12 x 512 КБ, L3 = 2 x 6 МБ) и 128 Гб оперативной памяти.

3. Гибридный расчетный узел (Intel). Имеет два 4-ядерных процессора Intel Xeon L5609 с частотой 1.87 ГГц (кэш L1 = 4 x 64 КБ (32+32), L2 = 4 x 256 КБ, L3 = 12 МБ), 32 Гб оперативной памяти и 2 GPU Nvidia Tesla M2050.

Тестирование межпроцессных взаимодействий

Первым направлением исследований различных архитектур является эффективность работы стандартной библиотеки pthreads (POSIX Threads), повсеместно используемой на системах с общей памятью как при разработке собственных приложений, так и в качестве базиса для построения более высокоуровневых средств реализации параллелизма в прикладных программах. Для данной библиотеки разработан тест, который замеряет время выполнения заданной pthread-функции при определенном общем уровне загрузки вычислительной системы, формируемой созданием группы потоков. Результаты измерений отображены в таблице 1.

Таблица 1.

Результаты измерений времени выполнения pthread-функций

Тест (секунда)	Узел кластера Opteron 6164HE	Узел кластера Opteron 6174	Узел кластера Xeon 5609
Порождение одного потока	0,000051	0,000047	0,000044
Порождение и ожидание одного потока	0,000069	0,000062	0,000111
Порождение и ожидание одного потока цепочкой (исп. 15000 потоков)	0,000179	0,000167	0,000095
Создание мьютекса	0,000000024	0,000000022	0,000000021
Уничтожение мьютекса	0,000000013	0,000000011	0,000000009
Блокировка мьютекса	0,000000042	0,000000033	0,000000015
Разблокировка мьютекса	0,000000040	0,000000033	0,000000018
Разблокировка заблокированного потока (исп. 100 потоков)	0,000524	0,000552	0,000514
Разблокировка заблокированного потока (исп. 200 потоков)	0,0012	0,0012	0,0009
Разблокировка заблокированного потока (исп. 500 потоков)	0,0030	0,0031	0,0024
Разблокировка заблокированного потока (исп. 1000 потоков)	0,0061	0,0069	0,0044

Анализируя полученные результаты, можно сделать следующие выводы. Наблюдается примерно 7–10 % превосходства процессоров Intel (Xeon L5609 vs. Opteron 6164HE). Разница в скорости выполнения функции pthread_create в приведенной группе процессоров составляет 3–5 мкс и коррелирует с частотой процессора. Кроме того, следует отметить несколько фактов, выявленных в результате тестирования и влияющих на скорость выполнения параллельных программ. Включение флага affinity при запуске программ может давать до 30 % сокращения времени накладных расходов за счет уменьшения затрат на переключение контекстов потоков между физическими ядрами процессоров. Запуск параллельных программ на процессорах Intel Xeon L5609 с 4 потоками оказывается более эффективным, чем с 8 потоками. В среднем время создания одного параллельного потока увеличивается до 30 раз при полной нагрузке на все 8 ядер.

Тестирование скорости записи в память

Эта группа тестов предназначена для оценки производительности работы памяти при последовательной записи одним потоком в общий массив памяти (порядка 80 % общего размера доступной памяти) блоками по k мегабайт. Результаты измерений показаны на рисунке 1 (шкала по оси OY является логарифмической; диапазон значений от 10 мкс до 1 с реального времени). Видно, что гибридный вычислительный узел в среднем на 25 % быстрее стандартного вычислительного узла. Более того, на объемах блоков, не превышающих размер кэш-памяти L3 (0.5, 1, 3 МБ), процессор Intel показал двукратное ускорение по сравнение с процессором AMD. Полученные результаты сравнивались с широко известной утилитой тестирования памяти memtest86+ [3], которая также показала, что подсистема памяти на узлах с процессором Intel работает в два раза быстрее, чем на узлах с процессором AMD. Учитывая, что конфигурация памяти на обоих узлах является практически одинаковой (у Intel модули памяти работают даже на меньшей частоте DDR3-1066 против DDR1333 у AMD), можно сделать вывод, что контроллер памяти у процессора Intel более производительный.

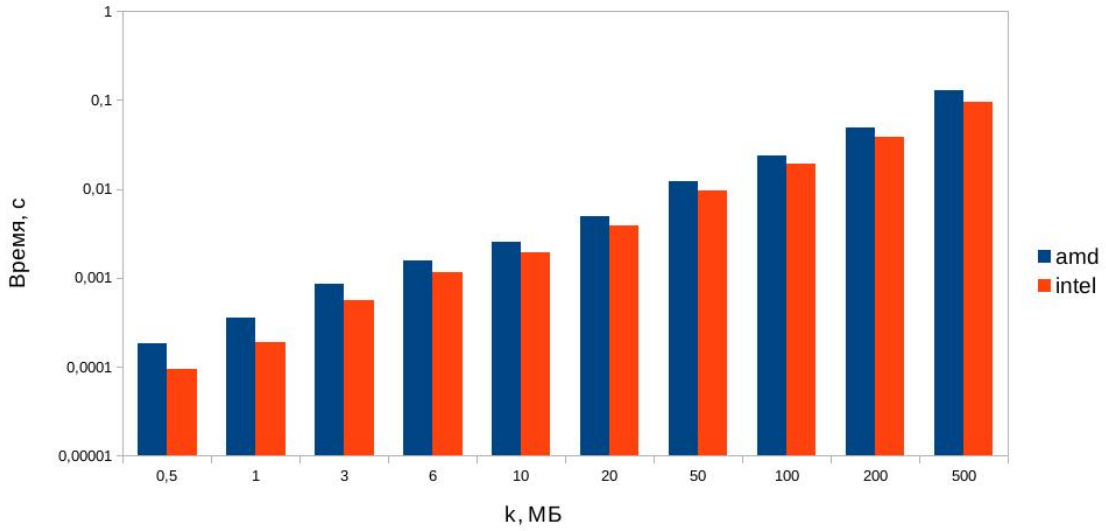


Рисунок 1. Производительность блочной записи в память

Тестирование многопоточной записи в память

При многопоточной записи в общий массив памяти интерес представляют различные режимы распределения потоков между процессами и ядрами. В этой группе тестов измерялась производительность при выполнении последовательной записи произвольных данных в память блоками по 500 МБ заданным количеством потоков при: а) отключенной привязке к ядру; б) включенной привязке потока к процессору; в) включенной привязке потока к ядру. Результаты измерений показаны на рисунке 2. Левый график отображает производительность памяти на узле с процессорами AMD, правый – на узле с процессорами Intel. Тонкой пунктирной линией отображается режим, когда привязка потока осуществляется по ядрам, то есть сначала выделялись ядра одного процессора, потом второго и т. д. Видно, что насыщение в этом режиме наступает тогда, когда полностью задействован один процессор (12 ядер). Дальнейшее увеличение числа потоков не оказывает влияния на скорость записи данных, так как каждый процессор обладает собственными каналами доступа к памяти. Толстым пунктиром показан режим, когда привязка потоков осуществляется по процессорам, то есть сначала выделяется первое ядро первого процессора, потом первое ядро второго процессора и т. д. по кругу. Видно, что насыщение в этом режиме идет плавно, ступеньками по четыре потока. То есть каждые следующие четыре потока

(на каком-либо процессоре) добавляют нагрузку на работу подсистемы памяти. И сплошной линией показан режим, когда привязка вообще не осуществлялась. Операционная система сама выбирала оптимальное распределение потоков по ядрам в соответствии с их нагрузкой. Видно, что этот график не имеет какой-либо периодичности за исключением постепенного нарастания времени с увеличением числа потоков. Интересно отметить, что при отсутствии привязки производительность работы памяти возросла (за меньшее время потоки записывают такой же объем данных). Для процессора Intel режим без привязки ядра не превысил по производительности другие режимы.

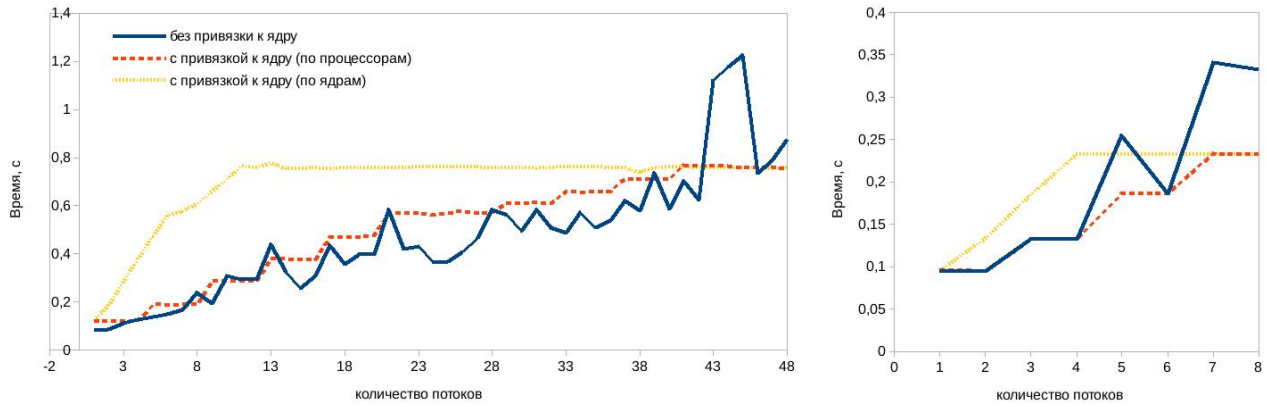
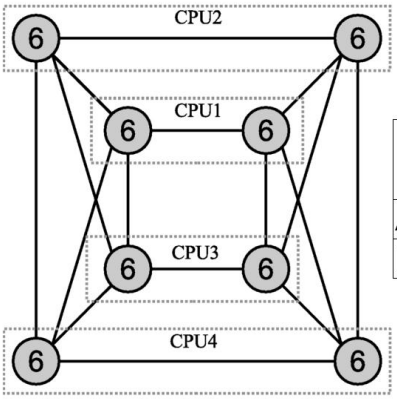


Рисунок 2. Производительность параллельной записи (AMD и Intel)

Тестирование записи в память NUMA-узлов

В этой группе тестов измерялась производительность передачи данных между памятью различных NUMA-узлов при копировании блоками по 500 МБ**.** Граф связности NUMA-узлов показан на рисунке 3: слева – вычислительный узел с процессорами AMD, справа – с Intel. Кружком показан один NUMA-узел (включая собственную память узла). Числом внутри NUMA-узла показано количество ядер данного узла. Пунктирной линией сгруппированы NUMA-узлы, которые образуют один физический процессор. В данной группе тестировались все возможные варианты передачи данных. С помощью библиотеки libnuma осуществлялась привязка кода к заданному NUMA-узлу i и выделялся блок памяти размером 500 МБ на узле j. После чего выполнялся алгоритм измерений, который состоял в копировании из памяти узла i в память

узла j блока указанного размера. Тесты были проведены для всех возможных пар вершин графа (включая петли), основные результаты сведены в таблицу на рисунке 3.



Узел на базе процессора	min время	max время
AMD	0.119 с	0.262 с
Intel	0.119 с	0.180 с

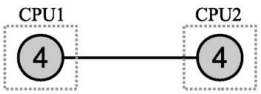


Рисунок 3. Производительность параллельной записи NUMA (AMD и Intel)

Тестирование скорости передачи сообщений

В этой группе тестов интерес представляет сравнение производительности передачи сообщений между MPI процессами на одном вычислительном узле и на разных узлах. На рисунке 4 изображены гистограммы измерений производительности передачи сообщений между двумя вычислительными процессами для двух разных размеров сообщения в случаях размещения процессов на одном вычислительном узле (фактически передача сообщений через общую память) и на двух разных узлах (передача сообщений через Infiniband).

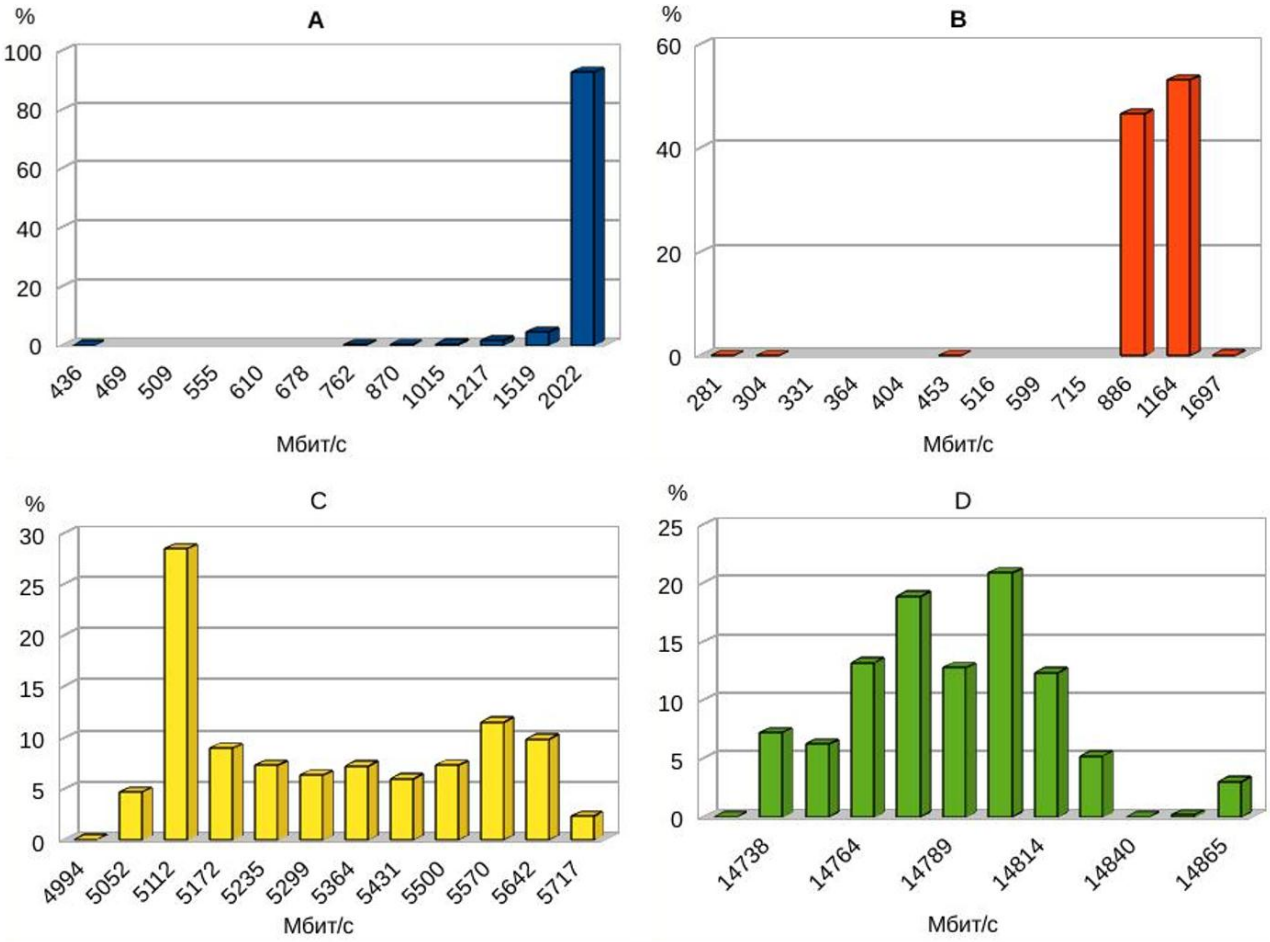


Рисунок 4. Гистограммы производительности коммуникационной подсистемы: (A) – один узел, размер сообщения 1 Кбайт; (B) – два узла, размер сообщения 1 Кбайт; (C) – один узел, размер сообщения 128 Мбайт; (D) – два узла, размер сообщения 128 Мбайт

В результате измерений производительности передачи MPI сообщений на узлах вычислительного кластера ЦКП «ДВВР» ДВО РАН был установлен следующий достаточно интересный факт. Как видно из приведённых гистограмм, производительность передачи сообщений для пакетов размером 1 Кбайт почти в четыре раза выше при использовании общей памяти, чем через Infiniband, в то время как для пакетов размером 128 Мбайт производительность передачи сообщений через Infiniband также почти в четыре раза выше, чем через общую память.

Заключение

При тестировании производительности многопроцессорных систем большое значение имеет четкая постановка задачи тестирования, которая во многом зависит от понимания элементарной базы и принципов построения системы.

При разработке тестов, представленных в настоящей статье, авторы хотели проверить собственные гипотезы о функционировании хорошо изученного ими оборудования. В некоторых случаях были получены неочевидные результаты, которые заставляют задуматься о сложности вычислительных систем. Так, при тестировании работы оперативной памяти узлов вычислительного кластера ЦКП «ДВВР» ДВО РАН обнаружено, что доступ к памяти в узлах с процессорами Intel быстрее, чем с процессорами AMD, хотя по своим внутренним характеристикам эта память медленней, а также что передача MPI-сообщений внутри одного узла может быть как быстрее, так и медленнее, чем передача сообщений между разными узлами. Такие особенности являются специфическими для тестируемого оборудования, их можно сравнивать с другими кластерами, а полученную информацию использовать при разработке новых программ. Учитывая полученный опыт, авторы рассчитывают продолжить разработку тестов, выявляющих новые особенности многопроцессорных систем.

Список литературы:

1. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. – СПб.: БХВ-Петербург, 2002. – 608 с.: ил.
2. Drepper U. What Every Programmer Should Know About Memory // Linux Weekly News / [Электронный ресурс]. – Режим доступа: URL: http://lwn.net/Articles/259710/ (дата обращения: 25.05.2013).
3. Memtest86 – And Advanced Memory Diagnostic Tool / [Электронный ресурс]. – Режим доступа: URL: http://www.memtest.org/ (дата обращения: 15.05.2013).

References:

1. Voevodin V.V., Voevodin Vl.V. Parallel computing. St. Petersburg, BHV-Peterburg Publ., 2002. 608 p.
2. Urlich Drepper. What Every Programmer Should Know About Memory. Available at: http://lwn.net/Articles/259710/ (Accessed 25 May 2013).
3. Memtest86 – And Advanced Memory Diagnostic Tool. Available at: http://www.memtest.org/ (Accessed 15 May 2013).