УДК 004.272.43

В.В. Щербинин, Ив.А. Шмаков, Ю.А. Баранчугов, Иг.А. Шмаков

Настольный параллельный компьютер с архитектурой MPI для научных расчетов и образовательного процесса

V.V. Scherbinin, Iv.A. Shmakov, Yu.A. Baranchugov, Ig.A. Shmakov

Desktop Parallel Computer with MPI Architecture for Science and Education

Описан настольный вычислительный MPI-кластер, предназначенный для решения вычислительных задач и обучения параллельному программированию. Кластер реализован на шести материнских платах форм-фактора mini-ITX в корпусе mini-tower ATX. В качестве программной платформы используется ОС Debian GNU/Linux с библиотекой OpenMPI. Приведены результаты тестирования производительности разработанной системы.

Ключевые слова: параллельные компьютеры, параллельные вычисления, технология MPI, GNU/Linux, OpenMPI.

In this paper a desktop MPI-cluster has been described. This device intended for scientific computing and parallel programming learning. The cluster based on six mini-ITX motherboards in single mini-tower ATX case. Debian GNU/Linux with OpenMPI library is the program platform of considered device. The results of performance measuring are presented.

Key words: parallel computers, HPC, MPI, GNU/Linux, OpenMPI.

Введение. Большинство задач, стоящих в настоящее время перед наукой и техникой, требуют большого объема машинных вычислений. Следствием этого является постоянно возрастающая потребность проектных и исследовательских организаций в вычислительных ресурсах.

Рост производительности однопоточных компьютеров принципиально ограничивается элементной базой – максимально достижимыми на данный момент тактовыми частотами, поэтому удовлетворение потребностей в вычислительных мощностях может быть обеспечено системами с параллельной либо распределенной обработкой данных [1, с. 12]. Мощные вычислительные кластеры довольно дороги и, кроме того, требуют специально оборудованных помещений и подготовленного обслуживающего персонала, что могут позволить себе только крупные вычислительные центры. Поэтому исследовательские и проектные группы, потенциально нуждающиеся в больших вычислительных мощностях, заинтересованы в недорогом и простом в обслуживании параллельном компьютере, обеспечивающем возможность разрабатывать и отлаживать программы с использованием технологии MPI.

К тому же довольно широкая распространенность кластерных систем требует подготовки специалистов, владеющих технологией написания параллельных прикладных программ. Для образовательных целей не требуются большие вычислительные мощности. При этом систему, используемую для обучения, желательно изолировать от рабочего кластера, применяемого для научных расчетов, что обеспечит студентам производительность независимо от загрузки вычислительного кластера. Такой комплекс позволит обучать студентов не только программированию, но и обслуживанию компьютерной системы, не ставя под угрозу функционирование узлов основного вычислительного кластера учебного заведения.

Таким образом, разработка компактного и недорогого параллельного компьютера с архитектурой, подобной архитектуре вычислительных кластеров, является достаточно актуальной задачей. Желательно, чтобы система по габаритам и стоимости не превосходила ПК и при этом превосходила бы типичный ПК по производительности.

Аппаратная реализация вычислительной системы. Возможным вариантом организации такой системы является компьютер, собранный из нескольких материнских плат малого форм-фактора в стандартном корпусе. При этом на большинстве узлов подобного настольного кластера можно обойтись без долговременного накопителя, что должно благоприятно сказаться на стоимости и энергопотреблении вычислительной системы. Загрузка операционной системы (ОС) на узлах такого кластера может осуществляться через сеть.

Материнские платы малого форм-фактора в настоящее время получили довольно широкое распространение. Наиболее часто на рынке встречаются

платы формата mini-ITX (172×172 мм), которые предназначены для малогабаритных домашних и офисных ПК, а также для компьютеров, используемых как основа для домашних кинотеатров. Кроме того, ограниченно представлены и более миниатюрные решения: nano-ITX (120×120 мм), pico-ITX (100×72 мм) и mobile-ITX (60×60 мм). Эти форм-факторы предназначены для промышленных и мобильных встраиваемых систем (банкоматов, информационных киосков и т.п.), причем первые два несут на себе минимальный набор стандартных портов ввода-вывода (VGA-разъем D-sub; 2-4 USB-порта; Ethernet и т.д.), а третий не содержит портов ввода-вывода и для их поддержки нуждается в дополнительной плате. Из числа вышеперечисленных материнские платы формата mini-ITX поддерживают наиболее широкий спектр центральных процессоров: фактически на них могут быть установлены любые выпускаемые в настоящий момент x86-совместимые процессоры, поэтому эти платы предпочтительны в качестве основы для разработки малогабаритного параллельного компьютера.

Важный эксплуатационный параметр любой компьютерной системы – уровень производимого шума. Основным источником последнего являются вентиляторы системы активного воздушного охлаждения. Таким образом, желательно конструировать настольный кластер с водяным охлаждением или на процессорах, способных работать с пассивным воздушным охлаждением. Системы водяного охлаждения малоудобны в эксплуатации, поскольку при их разгерметизации и утечке охлаждающей жидкости может произойти короткое замыкание и выход из строя отдельного узла или всей кластерной системы. Следовательно, пассивное воздушное охлаждение является предпочтительным с точки зрения удобства эксплуатации.

Наиболее производительными решениями форм-фактора mini-ITX с пассивным охлаждением являются системы, основанные на процессорах семейства CULV, производимых компанией Intel. Семейство CULV включает в себя одно- и двухъядерные процессоры с тактовыми частотами до 1.33 ГГц. К сожалению, подобные материнские платы в настоящий момент только анонсированы [2, с. 181–184] и, следовательно, практически недоступны для заказа, поэтому в качестве основы при изготовлении прототипа были выбраны материнские платы Intel D510MO (6 штук), укомплектованные процессором Intel Atom D510 (1,66 ГГц, 2 ядра, поддержка технологии Hyper-Threading) с пассивным охлаждением. Плата Intel D510MO имеет интегрированный гигабитный сетевой интерфейс.

Каждая плата в составе вычислительной системы была снабжена 4 Гбайт оперативной памяти DDR-II. К одной из плат подключены также жесткий диск и дополнительная сетевая карта. Эта плата, именуемая в дальнейшем *головной*, обеспечивает функционирование кластера в ждущем режиме, управление загрузочными образами и пользовательскими данными, хранящимися на жестком диске, а также играет роль маршрутизатора, позволяя ограничить взаимодействие между сетью организации и внутренней сетью кластера. Остальные пять материнских плат (именуемые *дочерними*) не имеют постоянных накопителей. Они включаются и выключаются по мере необходимости при проведении расчетов. Материнские платы объединены в агрегат, который установлен в корпус таким образом, что каждая из плат расположена вертикально, что улучшает отвод нагретого воздуха от радиаторов.

Функционирование внутренней сети обеспечивается гигабитным маршрутизатором 3Com Gigabit Switch 8, установленным в верхней части корпуса. Питание осуществляет блок Thermaltake Toughpower 1500W. Мощность блока питания была выбрана с большим запасом, поскольку планируется в дальнейшем увеличить количество узлов, входящих в систему. Кнопки питания и сетевые разъемы от одного из портов маршрутизатора и дополнительной сетевой карты, установленной на головной плате, выведены на переднюю панель корпуса. Там же установлен 12-сантиметровый вентилятор, предназначенный для выведения теплого воздуха из системного блока, что необходимо в жаркую погоду. Внешний вид разработанного прототипа представлен на рисунке 1.

Таким образом, разработанный кластер имеет 12 вычислительных ядер (или 24 виртуальных ядра, при использовании Hyper-Threading) и по 4 ГБ оперативной памяти на узел (часть этой памяти может использоваться как файловая система; в «штатном» режиме объем задействованной таким образом под системные нужды памяти не должен превышать 100 Мбайт).

Программная реализация вычислительной системы. Выбор программной платформы для вычислительной системы определялся рядом требований. Во-первых, система должна обладать развитым инструментарием для разработки прикладного программного обеспечения. Во-вторых, система должна быть гибкой в настройке и к тому же иметь минимально возможную стоимость. Наиболее полно предъявленным требованиям удовлетворяют свободные ОС на основе компонент, разработанных в рамках проекта GNU.

В качестве основного источника программного обеспечения был выбран вариант Debian системы GNU с ядром Linux в проходящей (на момент написания) тестирование перед официальным выпуском версии 6.0 («Squeeze») для 64-разрядной платформы AMD64. Были опробованы два варианта обеспечения загрузки: с флэш-накопителей и по сети.

Первый вариант более простой в реализации, поэтому на этапе первоначального тестирования системы оказался предпочтительным. Образы ОС, установленные на накопителях, практически иден-

тичны друг другу за исключением имен машин, IP-адресов и ключей Kerberos и SSH. Для автоматизированной индивидуализации образов была разработана программа на языке Shell. При загрузке с флэш-накопителей узлы кластера могут быть совершенно равноправны. Недостатком такого способа загрузки является крайне ограниченный объем дискового пространства – всего 2 ГБ на всех пользователей, и зависимость сохранности пользовательских данных от электропитания.

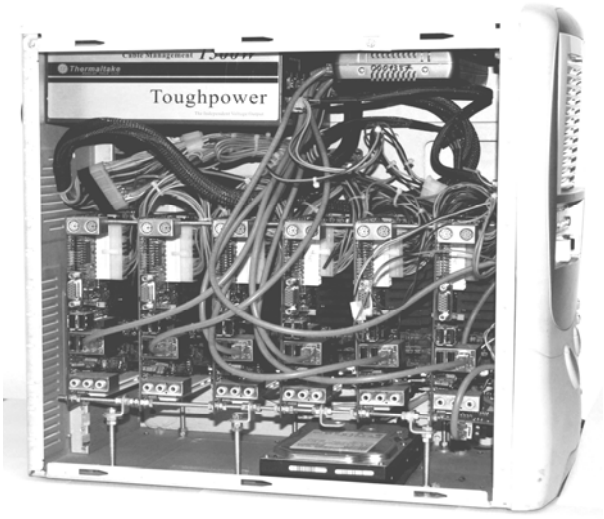

Рис. 1. Внешний вид прототипа настольного кластера

Второй вариант – с загрузкой по сети, – более удобен на практике, поскольку избавляет от необходимости клонировать образы системы, но несколько сложнее в первоначальной настройке. При этом головная плата должна выполнять роль *центрального узла*, который будет обеспечивать загрузку *дочерних*.

Основная система центрального узла. Наиболее значимым отличием конфигурации основной системы центрального узла от «минимальной» в настройках «по умолчанию» является наличие средств, обеспечивающих:

– сетевую загрузку бездисковых систем (с использованием таких средств и протоколов, как PXE, DHCP, TFTP, NFSv3/IPv4; в том числе пакетов Debian isc-dhcp-server, nfs-kernel-server, tftpd-hpa, syslinux);

– автоконфигурацию IPv6 (radvd);

– создание «образов» системы – как для целей удаленной загрузки (на основе Debian Live; пакет live-build), так и для установки *дочерних* (обеспечиваемых системным вызовом chroot) систем (пакет debootstrap);

– взаимодействие с хранимыми вне системы пользовательскими учетными записями – с использованием протоколов LDAPv3 (для открытой информации; пакет libnss-ldap) и Kerberos 5 (для аутентикации; пакеты libpam-heimdal, heimdal-clients);

– синхронизацию времени – через протокол NTP (что необходимо, в частности, для корректной работы используемого варианта протокола Kerberos).

Для организации дискового пространства используются таблица разделов на основе UUID GPT и система управления логическими томами LVM. В частности, отдельные логические тома LVM созданы для файловых систем (ФС) неизменяемых (/usr) и изменяемых (/var) системных файлов; пользовательских файлов; файлов, составляющих образ системы для сетевой загрузки; дочерних систем.

Кроме того, система включает следующие программные компоненты:

– набор обычных для GNU/Linux и других Unix-подобных систем инструментов (включая следующие пакеты: at, bc, bsd-mailx cpio, cron, dc, file, grep, info, less, lrzsz, man-db, mawk, screen, sed, tar, etc);

– текстовые редакторы (минимальный вариант редактора Vim, vim-tiny, а также mg, nano и zile);

– базовые инструменты системной (lsof, ltrace, strace) и сетевой (bind9-host, dnsutils, iputils-ping, tcpdump, whois) диагностики;

– реализацию «обычных» протоколов передачи файлов – Rsync (rsync), HTTP и FTP (wget, клиент);

– необходимую для корректной работы перенаправления X Window через SSH программу xauth.

Рабочие системы. Средства разработки, не имеющие отношения к основной задаче центрального узла – обеспечения сетевой загрузки рабочих узлов – вынесены в отдельную, *рабочую* систему, взаимодействие с которой осуществляется благодаря системному вызову chroot и пакету schroot, обеспечивающему доступ к оному непривилегированным пользователям.

Программная конфигурация рабочей системы центрального узла в основе повторяет конфигурацию основной, но имеет определенные отличия:

– отсутствие некоторых средств управления системой (программ диагностики и восстановления ФС, управления аппаратным обеспечением, обеспечения загрузки, etc.);

– наличие широкого спектра средств разработки, сборки ПО из исходного кода и отладки, в том числе компиляторы C и Fortran из комплекта GCC 4.4, универсальный сборщик GNU Make, GNU Emacs, реализации ряда высокоуровневых языков программирования, системы управления редакциями, etc.; библиотеки, в том числе OpenMPI, NetCDF, etc.;

– инструменты создания, преобразования, извлечения данных и метаданных для файлов различных форматов.

Загружаемые через сеть системы *дочерних* узлов имеют похожую конфигурацию (за исключением

наличия на них средств, обеспечивающих загрузку, в том числе специфичных для Debian Live).

Тестирование производительности. Производительность разработанного настольного кластера была протестирована на двух задачах: расчете характеристик согласования и поля излучения конечной волноводной антенной решетки с импедансным фланцем [3] и моделировании органических молекул с помощью пакета MPQC [4, с. 1214]. В качестве тестовой платформы использовался ПК с процессором Intel Core 2 Duo E6600 (2,4 ГГц, 2 ядра) и 2 ГБ ОЗУ.

Вычисления по первой тестовой задаче характеризуются близкой к кубической зависимостью времени расчета от количества элементов решетки. К примеру, решетка из 11 элементов требует приблизительно одной минуты вычислений на тестовом ПК, а из 33 элементов – уже 17 минут. Поскольку практический интерес представляют антенные решетки из десятков и сотен элементов, время вычислений может оказаться весьма значительным. При вычислениях на разработанном настольном кластере продемонстрирована заметно большая производительность, чем у тестового ПК. Отдельный интерес представляют результаты использования технологии Hyper-Threading в этой задаче, которая обеспечивает почти двукратный прирост производительности. В итоге при запуске в 24 потока (т.е. по 4 потока на узел) была достигнута производительность в 4,6–4,8 раза большая, чем у тестового ПК.

MPQC представляет собой свободно распространяемый программный пакет, предназначенный для расчета характеристик молекул с помощью решения уравнений Шредингера различными приближенными методами (в частности, методом Хартри-Фока). Пакет изначально разрабатывался для параллельных вычислительных систем как с общей памятью, так и вычислительных кластеров. Вычисление характеристик молекулы глицина продемонстрировало четырехкратный прирост производительности относительно тестовой платформы. Обращает на себя внимание тот факт, что использование технологии Hyper-Threading обеспечивает прирост производительности только на 27%.

В тесте High-Performance LINPACK [5] была получена сравнительно невысокая пиковая производительность 4.2 Гфлопс. Объяснить этот факт можно относительно низкой скоростью сетевого обмена, к которой чувствителен тест.

Заключение. Можно сделать вывод, что разработанный прототип настольного кластера в целом отвечает предъявленным требованиям. Его ограничениями являются относительно небольшой объем оперативной памяти (4 ГБ, включая объем, динамически выделяемый системой под хранение временных и рабочих файлов); отсутствие на дочерних системах носителей; относительно низкая скорость передачи информации внутренней сети. Это несколько ограничивает область применения данной системы: предпочтительно использовать ее для решения задач, связанных с большим объемом вычислений, но не требующих большого объема данных и интенсивного обмена данными между процессами.

Настольный кластер, созданный на основе разработанной модели, может быть использован как рабочими группами в исследовательских центрах, так и учебными заведениями для разработки и отладки программ с использованием технологии MPI, обучения специалистов в области параллельного программирования и обслуживания компьютерных систем.

Библиографический список

1. Немнюгин С.А., Стесик О.Л. Параллельное программирование для многопроцессорных вычислительных систем. – СПб., 2002.

2. Computex 2010: компактные платы от Zotac [Электронный ресурс] – 2010. – Режим доступа: http://www.3dnews.ru/news/Computex-2010-kompaktnie-plati-ot-Zotac/, свободный. – Загл. с экрана. – Яз. рус.

3. Щербинин В.В., Комаров С.А. Диаграмма направленности волноводной антенной решетки с импедансным фланцем // Известия АлтГУ. – 2010. – №1.

4. Valeev E.F., Janssen C.L. Second-order Moller-Plesset theory with linear R12 terms (MP2-R12) revisited: Auxiliary basis set method and massively parallel implementation // The Journal of chemical physics. – 2004. – №121.

5. Petitet A., Whaley R.C., Dongarra J., Cleary A. HPL – A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers [Электронный ресурс] – 2008. – Режим доступа: http://www.netlib.org/benchmark/hpl/, свободный. – Загл. с экрана. – Яз. англ.