

**УЧЕБНО-ЭКСПЕРИМЕНТАЛЬНЫЙ
ВЫЧИСЛИТЕЛЬНЫЙ КЛАСТЕР.****Ч. 1. ИНСТРУМЕНТАРИЙ И ВОЗМОЖНОСТИ**

Представлен учебно-экспериментальный вычислительный кластер кафедры “Прикладная математика” МГТУ им. Н.Э. Баумана. Для описания кластера и моделей высокопроизводительных вычислений введены понятия логической и аппаратной структуры вычислительной системы. Приведены основные технические характеристики созданной системы, а также результаты оценки производительности вычислительного кластера с помощью теста Linpack Benchmark. Рассмотрены варианты применения системы в учебной и научно-исследовательской деятельности.

E-mail: vvlukin@gmail.com; iliamarchevsky@mail.ru

Ключевые слова: параллельные вычисления, вычислительный кластер, математическое моделирование, архитектура ЭВМ, тест Linpack Benchmark.

Понятие вычислительного эксперимента появилось в середине XX в., что связано с бурным ростом возможностей вычислительной техники, появлением больших ЭВМ, позволивших проводить исследования ряда математических моделей, востребованных прежде всего в оборонной и космической отраслях. При этом программирование выделяется в специфический род деятельности, направленный в первую очередь на реализацию в рамках ЭВМ математических алгоритмов численного решения задач, продиктованных исследовательской практикой; в 1957 г. был разработан один из первых языков программирования высокого уровня — Фортран. Именно на этот период приходится становление прикладной математики как науки о математическом моделировании и вычислительном эксперименте [1].

С появлением на рубеже 1970–80-х годов персональных ЭВМ программирование перестало быть узкоспециальной деятельностью, появились новые языки высокого уровня и новые сферы приложения возможностей ЭВМ — от офисной и коммерческой деятельности до компьютерных игр. К 1990-м годам возможности вычислительного эксперимента стали доступны сравнительно широкому кругу специалистов инженерной сферы, получили распространение универсальные пакеты прикладных программ, позволяющие решать типичные задачи математического моделирования.

Параллельно развивалась и суперкомпьютерная техника, задействованная в решении более сложных задач. Под термином “суперкомпьютер” обычно понимают вычислительные машины, заметно превосходящие по своим характеристикам большинство существующих компьютеров. Это достигается за счет наличия в системе большого числа

вычислительных ядер, способных одновременно выполнять расчеты в рамках решения одной задачи. Выделение независимых частей алгоритма, исполняемых на различных ядрах, называется распараллеливанием. К настоящему времени получили развитие два подхода к организации взаимодействия вычислительных ядер на аппаратном уровне, реализованные в системах с общей и распределенной памятью [2].

Развитие компьютерных сетей привело к появлению в конце 1970-х годов систем с распределенной памятью — кластеров. Кластер — это группа компьютеров, объединенных высокоскоростными каналами связи и представляющих с точки зрения пользователя единый вычислительный ресурс [3]. С этого момента начинает развиваться новая сфера профессиональной деятельности, направленная на решение задач математического моделирования на таких системах — параллельное программирование, главная задача которого — достижение максимальной производительности вычислительного алгоритма, в частности, путем параллельного выполнения операций над векторами и матрицами. Формируется международное сообщество специалистов, занимающихся высокопроизводительными вычислениями, в 1993 г. запущен проект Top-500, дважды в год публикующий список самых мощных компьютерных систем мира [4].

На современном этапе параллельное программирование — как в свое время программирование последовательное — перестает быть “уделом избранных”. Достаточно указать на тот факт, что параллельные программы нужны теперь для проведения вычислений не только на суперкомпьютерах, но и на вполне обычных персональных компьютерах: современные процессоры являются многоядерными, и проводить расчеты на них с максимальной производительностью можно лишь используя параллельные вычислительные алгоритмы. К настоящему времени получили развитие такие технологии разработки параллельных программ, как MPI [5] (с целым набором свободных реализаций — OpenMPI, MPICH, LamMPI), OpenMP [6], shmem и др.

Использование параллельных вычислительных технологий становится неотъемлемой частью деятельности современного специалиста в области прикладной математики, программиста, инженера, исследователя. Для разработки параллельных программ специалист должен обладать знаниями о современном уровне развития программного и аппаратного инструментария и способностью применять эти знания на практике. Все звенья классической триады “модель–алгоритм–программа” тесно связаны между собой, поэтому наиболее эффективны те программы, которые изначально создаются с учетом их параллельного исполнения, что во многом меняет подход к разработке

методов и алгоритмов решения задач, и даже может влиять на выбор математической модели.

Осваивать подобного рода деятельность необходимо уже на уровне вузовского образования, включая практикум параллельного программирования в систему профессионального цикла дисциплин подготовки специалистов в области математического моделирования. Для этого сегодня имеются все возможности: стоимость компьютеров неуклонно снижается, сетевое оборудование также становится доступнее, что уже открыло возможность для научных групп использовать кластерные системы из десятков и даже сотен вычислительных ядер. В простейшем случае кластерная система может быть построена на базе персональных ЭВМ и “бытового” сетевого оборудования. В этом смысле идеальным “полигоном” для создания вычислительных кластеров являются компьютерные классы вузов, которые к тому же часто укомплектованы однотипными компьютерами.

Авторами статьи на базе компьютерного класса кафедры ФН2 “Прикладная математика” МГТУ им. Н.Э.Баумана создан учебно-экспериментальный кластер. Ниже рассмотрены его вычислительные возможности и применение в научно-исследовательской и образовательной деятельности кафедры и университета в целом.

Цель настоящей работы — исследование возможностей и способов применения и адаптации параллельных вычислительных технологий для решения широкого класса инженерных и научных задач. Современное развитие вычислительной техники, с одной стороны, и достигнутый к настоящему времени уровень абстракции, обеспечиваемый параллельными технологиями, с другой — дают возможность их массового применения, которая сейчас ограничивается отсутствием единой методологической базы и учебного курса.

Терминология. К настоящему времени издан ряд русскоязычных пособий, посвященных параллельным вычислениям, среди которых следует выделить [2, 3], однако единая общепринятая терминология применительно к вычислительным кластерным системам, по существу, отсутствует. Часто в разных источниках одни и те же термины рассматриваются в различных контекстах и имеют разный смысл. Это связано с тем, что базовое понятие вычислительной машины включает в себя как аппаратные, так и программные (логические) средства, которые, будучи согласованными между собой, позволяют тем не менее по-разному представлять архитектуру и возможности вычислительной системы. Поэтому представляется целесообразным разделить логический и аппаратный подходы к описанию кластера. Введем основные определения, которыми будем пользоваться в рамках данной статьи.

Кластер — объединение логической и аппаратной структуры единой с точки зрения пользователя вычислительной системы. Логическая структура кластера включает в себя вычислительные, управляющие и сервисные модули, между которыми возможен обмен информацией. Аппаратная структура представляется узлами кластера, соединенными между собой активным и пассивным сетевым оборудованием.

Вычислительное ядро — элементарный вычислитель, способный исполнять последовательную программу. В наиболее распространенном варианте понятия вычислительного ядра и ядра процессора совпадают, однако возможны и другие реализации; к примеру, технология Hyper-threading от Intel позволяет логически представить каждое процессорное ядро как два вычислительных ядра. Отметим, что применяемые для ускорения расчетов графические ускорители (технология nVidia CUDA) вычислительными ядрами, согласно принимаемой классификации, не являются. В случае их использования под вычислительным ядром следует понимать совокупность процессорного ядра и графического ускорителя.

Вычислительный модуль — одно или несколько вычислительных ядер, работающих с общей памятью. Если модуль объединяет несколько ядер, то он представляет собой симметричный мультипроцессор (SMP-машину).

Функции управляющих и сервисных модулей могут быть возложены на вычислительные ядра, как участвующие в расчетах, так и несущие исключительно вспомогательные функции.

Узел кластера на аппаратном уровне объединяет один или несколько модулей в сервер, имеющий общий для всех модулей набор сетевых интерфейсов (число интерфейсов может быть различным в зависимости от способа реализации каналов связи).

Набор узлов кластера фиксирован, в то время как его модульная структура может меняться в зависимости от используемой схемы вычислений. Она определяется аппаратной реализацией кластера, используемыми технологиями параллельного программирования и способом организации конкретного вычислительного эксперимента.

Отметим, что в рамках введенного понятия логической структуры кластер является вычислительной машиной массивно-параллельной архитектуры (MPP-машиной), реализация вычислительных модулей которой может быть различной. Здесь мы несколько расширяем исходно чисто аппаратное понятие MPP-архитектуры с тем, чтобы иметь возможность корректно и единообразно описывать современные многопроцессорные и многоядерные вычислительные системы. Если логически кластер состоит из модулей, в каждый из которых включено

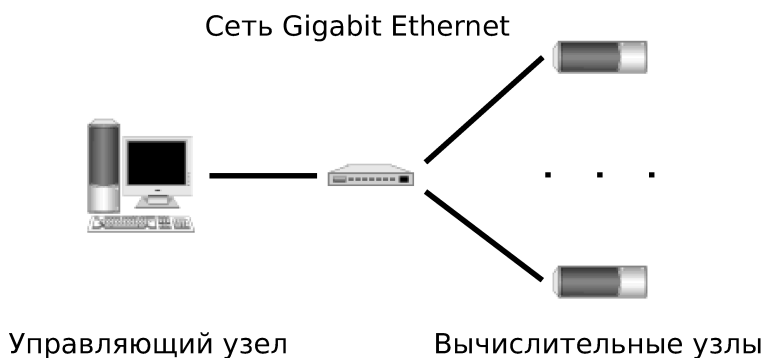


Рис. 1. Топология кластера

только одно ядро, то будем говорить о “чистой” MPP-машине, в противном случае — о смешанной MPP-машине с SMP-модулями.

Учебно-экспериментальный вычислительный кластер. *Аппаратная структура.* Учебно-экспериментальный вычислительный кластер построен на базе персональных компьютеров, оснащенных бытовыми процессорами Intel Core i7. Система состоит из девяти связанных сетью Gigabit Ethernet узлов, из которых восемь участвуют в вычислениях и один выполняет вспомогательные (управляющие и сервисные) функции. Аппаратная топология сети представлена на рис. 1. (топология “звезда”). Вычислительные узлы имеют следующие характеристики:

процессор — четырехъядерный Intel Core i7-920 с тактовой частотой каждого ядра 2,66 ГГц (производительность, по оценкам Intel, 42,56 Гфлопс [7]);

системная плата — MSI X58 PRO-E на базе чипсета Intel X58+ICH10R с частотой системной шины 4800 МТ/с;

оперативная память — 12 Гбайт трехканальной памяти стандарта DDR3, работающей на частоте 1333 МГц;

жесткий диск — SATA II Seagate емкостью 500 Гбайт;

сетевой интерфейс — сетевая карта Gigabit Ethernet на базе процессора Realtek 8111С.

Каждый вычислительный узел можно рассматривать по-разному. С одной стороны, это четыре независимых вычислительных модуля, каждый из которых работает со своей областью памяти, выделяемой из общего объема ОЗУ данного узла средствами операционной системы. При этом обмен данными между модулями внутри узла является быстрым, поскольку осуществляется по внутренней шине без использования локальной сети. В этом случае вычислительная часть кластера логически представляет собой 32-модульную систему. С другой стороны, каждый узел кластера может рассматриваться как машина с

общей памятью и представлять собой один вычислительный модуль с четырьмя ядрами. Тогда вычислительная часть кластера — это смешанная восьмимодульная система, где каждый модуль — SMP-машина.

Управляющий узел (напрямую не задействованный в вычислениях) имеет характеристики:

процессор — четырехъядерный Intel Core 2 Quad 8400 с тактовой частотой каждого ядра 2,66 МГц (производительность по оценкам Intel 42,56 Гфлопс);

системная плата — Gigabyte GA-EP45-UD3LR на базе чипсета Intel P45+ICH10R;

оперативная память — 4 Гбайт двухканальной памяти стандарта DDR2;

жесткие диски — SATA II Seagate емкостью 500 Гбайт для операционной системы и два диска SATA II Seagate емкостью 1 Тбайт, объединенных в Raid-0 массив, используемый в качестве хранилища данных, доступного по сети вычислительным узлам;

сетевой интерфейс — сетевая карта Gigabit Ethernet на базе процессора Realtek 8111C.

Коммуникации осуществляются с использованием сетевого коммутатора D-Link Gigabit Switch DGS-1016D на 16 портов.

Программное обеспечение кластера. В качестве программной платформы системы выбран кластерный дистрибутив Rocks Clusters, основанный на открытой свободно распространяемой операционной системе CentOS. Unix-подобные операционные системы чаще всего становятся платформой, на которой разворачиваются кластеры (как вычислительные, так и кластеры высокой доступности), что связано прежде всего с зарекомендовавшим себя уровнем поддержки сетевых сервисов. Кроме того, Linux-кластеры являются гораздо более гибкими, масштабируемыми и доступными по цене, чем аналогичные решения, например, от Microsoft. Указанный кластерный дистрибутив Rocks является открытым свободно распространяемым продуктом. Он включает в себя основные автоматически настраиваемые сервисы для увязки узлов системы в единый вычислительный кластер. В числе прочего он обеспечивает:

— возможность использования автоматизированных средств расширения и контроля состояния кластера (в том числе система Ganglia);

— возможность адаптации логической структуры кластера к конкретному вычислительному эксперименту;

— использование единого набора компиляторов и библиотек как на управляющем, так и на вычислительных узлах, а также возможность переключения между различными реализациями библиотек (система Module);

- автоматическое монтирование файловой системы управляющего узла на вычислительных узлах, что обеспечивает простоту доступа и единство обрабатываемых данных;
- беспарольный доступ на каждый вычислительный узел с помощью технологий ssh и rsh;
- автоматизированное создание учетных записей пользователей системы с отдельными правами доступа;
- обработку очереди и управление выполнением заданий (система Torque).

Существуют и другие кластерные платформы, например системы MOSIX, OSCAR, PelicanHPC и другие, обладающие схожими возможностями. Кластерный дистрибутив Rocks Clusters был выбран исходя из нескольких соображений. Во-первых, в отличие от системы OSCAR он не требует наличия на узлах кластера предустановленной операционной системы, что упрощает процесс добавления в кластер вычислительных узлов. Во-вторых, эта система ориентирована на относительно небольшие вычислительные установки (а в данном случае речь идет о кластере “кафедрального” уровня) и не требует столь существенных затрат на поддержку и системное администрирование, как, например, система MOSIX, предназначенная для суперкомпьютеров промышленного масштаба. В то же время дистрибутив Rocks Clusters позволяет провести настройку основных параметров кластера и использовать имеющееся оборудование намного эффективнее по сравнению с системой PelicanHPC, которая не требует предварительной установки и загружается с внешнего (DVD или USB) носителя в заранее выбранной производителем стандартной конфигурации.

Дополнительно к указанным системам на кластере установлен пакет компиляторов Intel для языков C/C++ и Fortran. Этот пакет поддерживает технологию распараллеливания для систем с общей памятью OpenMP и включает собственную реализацию библиотеки линейной алгебры BLAS (Basic Linear Algebra Subroutines). В качестве библиотеки для вычислений с распределенной памятью выбрана библиотека OpenMPI, скомпилированная с поддержкой компиляторов Intel. Также пользователям доступны открытые компиляторы GCC и GFortran с соответствующим вариантом библиотеки OpenMPI. Все перечисленное программное обеспечение является свободно распространяемым (компиляторы Intel — в рамках лицензии для образовательных и научных учреждений).

Кроме того, на управляющем узле установлено стандартное математическое программное обеспечение, включающее системы разработки ПО, системы визуализации полученных результатов и т.д.

Отметим, что кластер построен на базе стандартного оборудования компьютерного класса. Это, с одной стороны, существенно снизило стоимость всей установки по сравнению с приобретением готового кластерного решения от одного из ведущих производителей такого типа техники (Hewlett-Packard, IBM, Т-Платформы и др.). С другой стороны, опыт создания и настройки кластера сам по себе является ценным как с методической, так и с практической точки зрения: он позволяет при проведении вычислительного эксперимента учитывать более тонкие особенности системы, чем это доступно при использовании готовых (и в этом смысле “закрытых”) коммерческих решений.

Настройка подобной кластерной установки является нетривиальной задачей, но вполне доступна специалисту, имеющему инженерное образование и знакомому с операционными системами Linux.

Оценка производительности кластера. *Описание теста Linpack Benchmark.* Производительность кластера оценена с помощью стандартного тестового пакета Linpack Benchmark [8], скомпилированного при помощи компилятора Intel и оптимизированного под архитектуру Intel Nehalem, лежащую в основе используемых процессоров Intel Core i7. Этот тест состоит в решении системы линейных алгебраических уравнений вида $Ax = b$, где A — полностью заполненная матрица размера $N \times N$, методом LU -факторизации с частичным выбором главного элемента. При этом в процессе факторизации матрицы L и U хранятся в памяти на месте исходной матрицы A .

При параллельном расчете на вычислительном кластере исходная матрица разделяется на элементарные блоки размерностью $N_B \times N_B$. Полученные $n^2 = (N/N_B)^2$ блоков распределяются на вычислительные модули по схеме $P \times Q$ (рис. 2), причем $P \cdot Q = n_m$, где n_m — число вычислительных модулей, задействованных в эксперименте. На каждом шаге расчета происходит LU -разложение N_B столбцов матрицы A , пересылка данных между модулями с использованием одного из шести коммуникационных алгоритмов и обновление оставшейся части расширенной матрицы $(A|b)$. При таком алгоритме все малые блоки, входящие в крупный блок, состоящий из $P \times Q$ малых, обрабатываются одновременно, причем число n задает число шагов расчета, а размер малого блока N_B определяет размер

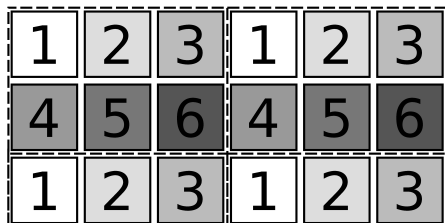


Рис. 2. Распределение блоков, состоящих из $N_B \times N_B$ элементов, по модулям при $P = 3$ и $Q = 2$. Цифрами обозначены соответствующие номера модулей, пунктиром выделены крупные $P \times Q$ -блоки

массива данных, пересылаемого между вычислительными модулями на каждом шаге.

После разложения последовательно решаются две системы уравнений: $Ly = b$, $Ux = y$. По окончании расчета матрица A и правая часть b системы воссоздаются и вычисляется невязка полученного решения $f = Ax - b$. Тест считается выполненным, если $r_N < \alpha$, $r_1 < \alpha$ и $r_\infty < \alpha$, где α — константа порядка единицы, ε — точность представления чисел с плавающей точкой,

$$r_N = \|f\|_\infty / (\|A\|_1 \cdot N \cdot \varepsilon); \quad r_1 = \|f\|_\infty / (\|A\|_1 \cdot \|x\|_1 \cdot \varepsilon);$$

$$r_\infty = \|f\|_\infty / (\|A\|_\infty \cdot \|x\|_\infty \cdot \varepsilon).$$

Коэффициенты P и Q выбираются в зависимости от структуры кластера. Например, если кластер состоит из восьми вычислительных модулей, возможными являются комбинации 1×8 , 2×4 , 4×2 , 8×1 , 2×3 и т.п. Если размерность решаемой системы N достаточно велика, то максимальная производительность достигается в одном из тех случаев, когда $P \cdot Q = 8$, т.е. задействованы все вычислительные модули кластера.

Производительность кластера R определяется по времени, затрачиваемому на выполнение $\frac{2}{3}N^3 + 2N^2$ операций с плавающей точкой, требуемых для решения системы. Производительность принято указывать в гигафлопсах (Гфлопс) — миллиардах операций с плавающей точкой в секунду.

Результаты анализа производительности кластера. В силу нелинейного влияния на эффективность распараллеливания различных характеристик решаемой задачи (размера матрицы, размеров пересылаемых блоков данных, способа разбиения матрицы на блоки и т.д.) для получения максимальной производительности на каждой конкретной кластерной системе приходится подбирать собственные оптимальные значения этих параметров.

Зависимость оценки производительности от размера решаемой системы достаточно хорошо исследована [3], и чем большую долю занимает вычислительная составляющая расчета по сравнению с обменом данными, тем большую производительность показывает кластер. Для демонстрации этой закономерности были проведены расчеты для систем линейных уравнений различной размерности (табл. 1).

Таблица 1

Производительность кластера в зависимости от размерности системы ($P = 4$, $Q = 2$)

Размерность системы N	30 000	50 000	70 000	90 000	110 000
Производительность R_8 , Гфлопс	152,1	198,1	223,7	239,5	250,8

Дальнейшее увеличение размерности системы приводит к тому, что размер хранимых данных превышает объем доступной оперативной памяти (суммарно 96 Гбайт на все вычислительные модули) и часть данных сохраняется в виртуальной памяти на жестком диске, что в несколько раз снижает производительность системы в целом. Представленный пример убедительно доказывает, что для оценки производительности кластера в дальнейшем достаточно проводить расчеты с матрицами наибольшего доступного размера.

Приведенные данные относятся к расчетам, в которых кластерная система конфигурировалась как смешанная MPP-машина с восемью SMP-модулями. Подобный подход, использующий технологию OpenMP для систем с общей памятью, по-видимому, позволяет оптимально использовать вычислительные ресурсы, сосредоточенные на одном узле кластера.

В табл. 2 приведена производительность системы для случая, когда кластер конфигурировался как “чистая” MPP-машина из 32 вычислительных модулей. Хотя на оценку производительности существенно влияет способ распределения блоков матрицы между вычислительными модулями (числа P и Q), во всех вариантах наблюдается значительное снижение производительности, что связано с возрастанием числа и сложности организации обменов между вычислительными модулями. Кроме того, такая логическая структура кластера приводит к дополнительным затратам оперативной памяти на хранение служебной информации, поэтому максимальный размер обрабатываемой матрицы становится несколько меньше. В табл. 2 приведены данные для $N = 90\,000$ в сравнении с соответствующим результатом из табл. 1.

Таблица 2

Производительность структуры из 32 MPP-модулей

$P \times Q$	1×32	2×16	4×8	8×4	16×2	32×1
R_{32} , Гфлопс	187,7	200,9	198,0	174,6	121,8	68,1
$R_{32}/R_8 \cdot 100\%$	78,4	83,9	82,7	72,9	50,9	28,4

Зависимость производительности кластера от соотношения $P \times Q$ проявляется и для случая восьмимодульного кластера с SMP-машинами. В табл. 3 приведены результаты расчетов для $N = 110\,000$.

Таблица 3

Производительность структуры из 8 SMP-модулей

$P \times Q$	1×8	2×4	4×2	8×1
R_8 , Гфлопс	240,3	258,7	250,5	210,6

К более тонким настройкам теста Linpack Benchmark относится выбор схемы пересылки информации между вычислительными модулями, варианта алгоритма LU -разложения, способа хранения матриц в памяти и некоторые другие. Расчеты показывают, что наибольшее влияние на производительность оказывает выбор схемы пересылок (табл. 4), в то время как остальные параметры изменяют оценку производительности кластера не более чем на 1 %. Их оптимальные значения были подобраны экспериментально.

Таблица 4

Производительность в зависимости от схемы пересылок данных, $P = 2, Q = 4$

Схема пересылок	Increasing ring	Increasing ring (mod.)	Increasing 2 ring	Increasing 2 ring (mod.)	Long	Long (mod.)
R_8 , Гфлопс	262,2	259,1	260,2	261,4	258,9	249,4

Таким образом, максимальная производительность кластера, измеренная по тесту Linpack Benchmark, составила 262,2 Гфлопс. С такой производительностью кластер решает систему из 110 000 линейных уравнений за 3383,9 с.

Для оценки эффективности использования ядер была проведена оценка пиковой производительности кластера. Она была получена исходя из оценки производительности каждой SMP-машины, на которой исполнялся тот же тест Linpack Benchmark для системы из $N = 39\,000$ линейных уравнений. В процессе расчета обмен данными по локальной сети отсутствовал. Производительность каждого узла составила $R_1^* = 41,5$ Гфлопс, следовательно, пиковая производительность кластера составляет $R_8^* = 8 \cdot R_1^* = 332,0$ Гфлопс, а максимальная производительность составляет 79,0 % от пиковой. Отметим, что измеренная пиковая производительность каждого модуля отличается от заявленной фирмой-производителем процессора лишь на 2,6 %.

Как видно, производительность кластера зависит от решаемой задачи и способа ее распараллеливания. Поэтому оценка, полученная с помощью теста Linpack Benchmark, является достаточно условной, однако именно этот тест де-факто стал международным стандартом. В реальных приложениях производительность кластера может существенно отличаться от этой оценки как в меньшую, так и в большую сторону.

Перспективы использования кластера. Деятельность современного инженера-вычислителя требует от специалиста в области прикладной математики глубокого теоретического владения методами построения и анализа математических моделей, а также практических

навыков их использования, в том числе в рамках стандартных пакетов. Вместе с тем актуальные на сегодня задачи не только наукоемки, но и ресурсоемки, поэтому подготовка специалистов должна сочетать как фундаментальное физико-математическое образование, так и базовую подготовку в области высокопроизводительных (параллельных) вычислений.

Учебно-экспериментальный вычислительный кластер может использоваться для проведения ресурсоемких расчетов при решении различных учебных и исследовательских задач математического моделирования физических процессов и технических систем, а также для обучения студентов основам параллельного программирования в рамках специализированного вычислительного практикума. Для этого могут применяться как стандартные пакеты прикладных программ, так и программы собственной разработки.

К настоящему времени разработан целый ряд пакетов для решения задач механики сплошной среды, имеющих реализации для параллельных вычислительных систем. Многие из них стали де-факто стандартами при проведении расчетов в соответствующих отраслях промышленности. Значительная часть таких пакетов является свободно распространяемой, а зачастую и открытой. Это позволяет широко использовать их возможности, а при необходимости и модифицировать, дополняя новыми математическими моделями и численными методами. Один из них — OpenFOAM (Open Source Field Operation And Manipulation CFD ToolBox) — установлен на кластере и позволяет решать комплексные междисциплинарные задачи математического моделирования. В частности, на кафедре “Прикладная математика” МГТУ им. Н.Э. Баумана данный пакет используется для решения задач аэродинамики, а также связанных теплогидравлических задач.

Еще одной сферой применения кластера является проведение исследований по ряду научных направлений, развиваемых на кафедре “Прикладная математика” МГТУ им. Н.Э. Баумана. В рамках этих направлений разрабатываются авторские пакеты программ, изначально предусматривающие использование параллельных вычислительных технологий. Отладка таких программ на относительно небольшой вычислительной системе является важным этапом и хорошим подспорьем при подготовке вычислительных экспериментов на больших вычислительных машинах класса МВС-100К МСЦ РАН, К-100 ИПМ им. М.В. Келдыша РАН, СКИФ-МГУ “Чебышев” и др. Это связано с тем, что программное обеспечение и логическая структура учебно-экспериментального кластера максимально приближены к характеристикам указанных выше суперкомпьютерных систем и имеется возможность получать исчерпывающую информацию о прохождении

расчета для ее использования в процессе оптимизации и отладки разрабатываемого программного кода.

Одно из направлений исследований — математическое моделирование формирования астрофизических релятивистских струйных выбросов в окрестности компактных объектов — нейтронных звезд и черных дыр. Особенностью таких процессов являются высокие коллимация струи и энергетика вещества выброса: угол раствора конуса, в который происходит выброс, чаще всего не превышает 10° , а скорость вещества достигает субсветовой (в галактике M87 скорость вещества достигает $0,9c$, где c — скорость света). Модель включает в себя систему уравнений идеальной магнитной гидродинамики (МГД), описывающую движение аккрецирующей на гравитирующий объект плазмы, и уравнение переноса излучения (УПИ) тонкого аккреционного диска, ускоряющего вещество выброса. Уравнения рассматриваются в двумерной осесимметричной постановке и решаются с помощью конечно-разностного метода для МГД-системы [9, 10] и метода длинных характеристик для УПИ [11].

Задача является связанной и предъявляет высокие требования к вычислительным ресурсам, прежде всего для учета рассеяния при моделировании излучения. Для решения уравнений математической модели разработан программный комплекс PLASMA, использующий технологии параллельного программирования для систем с общей и распределенной памятью. Комплекс основан на явных вычислительных схемах, что обеспечивает высокую эффективность распараллеливания МГД-части задачи с использованием библиотеки MPI (рис. 3). Работа с общей памятью в рамках SMP-машин приводит к существенному ускорению расчетов УПИ по методу длинных характеристик, по-

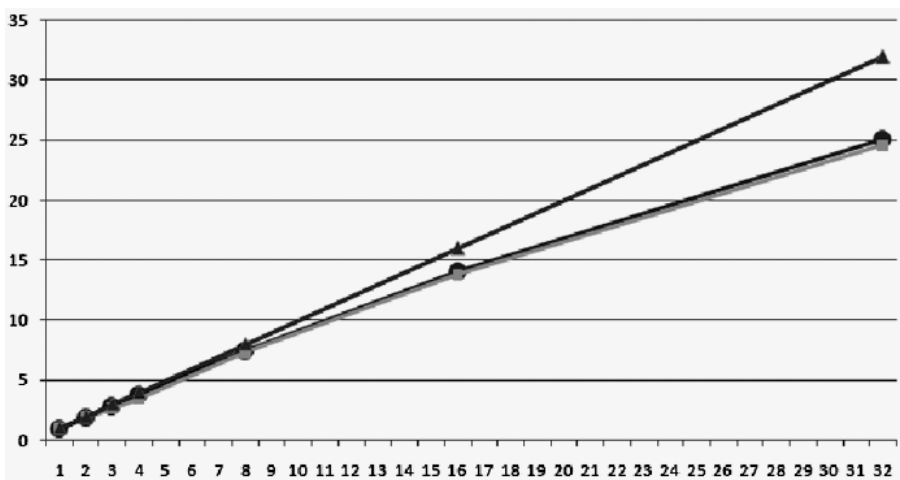


Рис. 3. Ускорение МГД-расчетов на кластере для задачи об ускорении замагниченного выброса из окрестностей компактного объекта:

● — закон Амдала; ■ — MPI; ▲ — линейное ускорение

зволюя избежать пересылок большого объема данных, описывающих излучение при выбранной сетке направлений. К примеру, использование с помощью технологии OpenMP в рамках вычислительного модуля четырех ядер процессора вместо одного позволяет ускорить эту часть вычислений в 3,8 раза.

Другим направлением исследований является математическое моделирование течения несжимаемой жидкости и решение связанных задач аэрогидроупругости с использованием бессеточных лагранжевых методов. В рамках данной тематики рассматриваются задачи моделирования обтекания двумерных профилей и трехмерных тел, исследования динамики вихревых структур в идеальной и вязкой жидкости, а также сопряженные задачи, когда обтекаемое тело является упругозакрепленным или деформируемым. В основу математической модели течения среды положены уравнения Навье–Стокса, решаемые методом вихревых элементов (МВЭ). При этом непрерывное распределение завихренности в области течения моделируется большим числом изолированных вихревых элементов (вихревых нитей, вихревых рамок, вихревых сгустков, вихревых рамок, вихревых нитей, вихревых рамок, вихревых сгустков, вортонов) [12]. Эволюция вихревых элементов описывается системой обыкновенных дифференциальных уравнений, решаемых явным методом Эйлера или Рунге–Кутты. Использование явного метода позволяет существенно ускорить решение системы ОДУ, используя технологию MPI. Для остальных этапов решения задачи также разработаны параллельные алгоритмы, что позволило довести долю параллельно исполняемого кода в программном комплексе MVE до 99,5 % [13].

Определение аэродинамических характеристик профилей предполагает решение серии однотипных независимых задач, различающихся только параметрами расчетной схемы. Для этого был разработан параллельный программный комплекс POLARA, позволяющий также оптимизировать использование имеющихся вычислительных ресурсов [14]. В силу независимости решаемых задач ускорение расчета практически линейно зависит от числа задействованных вычислительных модулей (рис. 4).

Таким образом, учебно-экспериментальный вычислительный кластер представляет ценность и как самостоятельный вычислительный ресурс кафедры, и как инструмент моделирования вычислительного эксперимента на больших суперкомпьютерных системах. Более подробно полученный опыт использования кластера, а также области его приложения будут рассмотрены во второй части настоящей работы.

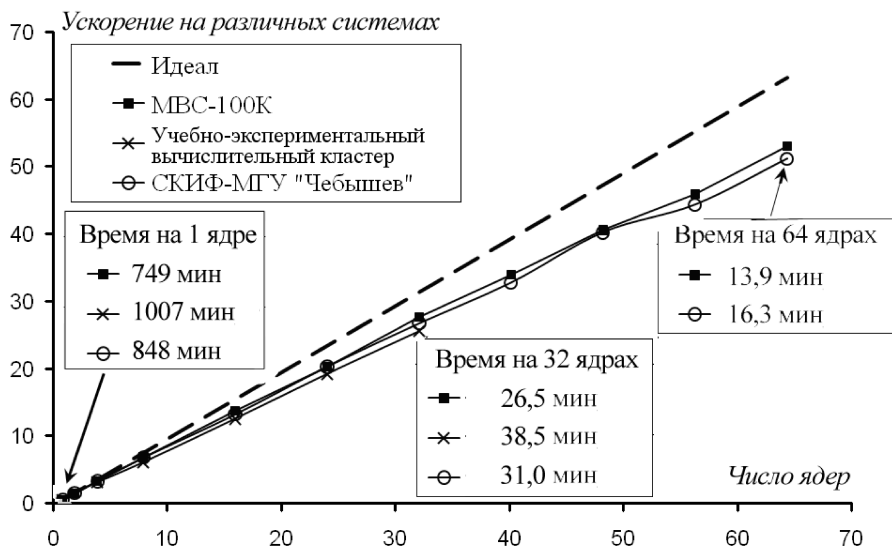


Рис. 4. Ускорение расчетов при использовании программного комплекса POLARA

Работа выполнена при частичной финансовой поддержке РФФИ (проект № 09-01-00151-а) и в рамках программы “Университетский кластер”.

СПИСОК ЛИТЕРАТУРЫ

- Самарский А. А., Михайлов А. П. Математическое моделирование: Идеи. Методы. Примеры. – М.: Физматлит, 2002. – 320 с.
- Воеводин В. В., Воеводин В. Л. Параллельные вычисления. – СПб.: БХВ-Петербург, 2004. – 608 с.
- Гергель В. П. Теория и практика параллельных вычислений. – М.: БИНОМ. Лаборатория знаний, 2007. – 423 с.
- Top 500 Supercomputing Sites. 2000. URL: <http://www.top500.org> (дата обращения 20.05.2011).
- Корнеев В. Д. Параллельное программирование в MPI. – Новосибирск: Изд-во СО РАН, 2000. – 220 с.
- Chandra R., Dagum L., Kohr D. et al. Parallel programming in OpenMP. – Morgan Kaufmann, 2001.
- Processors – Intel microprocessor export compliance metrics // Intel Corp. Site. 2006. URL: <http://www.intel.com/support/processors/sb/cs-023143.htm> (дата обращения 20.05.2011).
- Dongarra Jack J., Luszczek Piotr, Petitet Antoine. The LINPACK benchmark: Past, present and future // Concurrency and Computation: Practice and Experience. – 2003. – Vol. 15, no. 9. – P. 803–820.
- Галанин М. П., Лукин В. В. Разностная схема для решения двумерных задач идеальной МГД на неструктурированных сетках. – 2007. – 29 с. Препр. Инст. прикл. матем. им. М.В. Келдыша РАН. №50.
- Галанин М. П., Лукин В. В., Чечеткин В. М. Математическое моделирование струйных выбросов в окрестности компактных объектов // Астрономический журнал. – 2009. – Т. 86, № 4. – С. 331–344.

11. Галанин М. П., Лукин В. В., Чететкин В. М. Методы решения уравнения переноса излучения для астрофизических моделей. – 2010. – 30 с. Препр. Инст. прикл. матем. им. М.В.Келдыша РАН. № 59.
12. Марчевский И. К., Щеглов Г. А. Модель симметричного вортон-отрезка для численного моделирования пространственных течений идеальной несжимаемой среды // Вестник МГТУ им. Н.Э. Баумана. Сер. Естественные науки. – 2008. – № 4. – С. 62–71.
13. Марчевский И. К., Щеглов Г. А. Применение параллельных алгоритмов при решении задач гидродинамики методом вихревых элементов // Вычислительные методы и программирование. – 2010. – Т. 11. – С. 105–110.
14. Марчевский И. К., Морева В. С. Высокопроизводительный программный комплекс POLARA для определения аэродинамических характеристик профилей // Параллельные вычислительные технологии (ПАВТ'2010): Тр. Междунар. конф. – Челябинск, 2010. – С. 533–538.

Статья поступила в редакцию 23.05.2011

Владимир Владимирович Лукин родился в 1985 г., окончил МГТУ им. Н.Э. Баумана в 2008 г. Аспирант Института прикладной математики им. М.В. Келдыша РАН, ассистент кафедры “Прикладная математика” МГТУ им. Н.Э. Баумана. Автор 12 научных работ в области математического моделирования астрофизических процессов, численных методов магнитной гидродинамики и теории переноса излучения.

V.V. Lukin (b. 1985) graduated from the Bauman Moscow State Technical University in 2008. Post-graduate of the Keldysh Institute for Applied Mathematics, RAS; assistant lecturer of “Applied Mathematics” department of the Bauman Moscow State Technical University. Author of 12 publications in the field of mathematical simulation of astrophysical processes, numerical methods of magnetohydrodynamics and radiation transport theory.



Илья Константинович Марчевский родился в 1983 г., окончил МГТУ им. Н.Э. Баумана в 2005 г. Канд. физ.-мат. наук, доцент кафедры “Прикладная математика” МГТУ им. Н.Э. Баумана. Автор 35 научных и учебно-методических работ в области исследования движения и устойчивости конструкций в потоке среды, вычислительной гидродинамики, высокопроизводительных вычислений, элементарной математики.

I.K. Marchevskii (b. 1983) graduated from the Bauman Moscow State Technical University in 2005. Ph. D. (Phys.-Math.), assoc. professor of “Applied Mathematics” department of the Bauman Moscow State Technical University. Author of 35 publications in the field of study of motion and stability of constructions in a flow of medium, computational fluid dynamics, high-efficient computing and elementary mathematics.

