Розділ 3. Науково-технологічна безпека

УДК 621.3

ОБ ОЦЕНКЕ ЭФФЕКТИВНОСТИ МНОГОПРОЦЕССОРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

С. К. Полумиенко, д-р физ.-матем. наук;
Д. А. Федюков
(Институт телекоммуникаций и глобального информационного пространства НАН Украины)

В работе проводится общий обзор методов построения кластерных систем, выделяются основные направления их формирования, а также задачи, возникающие в процессе повышения эффективности или оптимизации их функционирования.

У роботі проводиться загальний огляд методів побудови кластерних систем, виділяються основні напрями їх формування, а також завдання, що виникають в процесі підвищення ефективності або оптимізації їх функціонування.

The general review of methods of construction of the cluster systems is in-process conducted, basic directions of their forming, and also tasks, arising up in the process of increase of efficiency or optimization of their functioning, are selected.

Исторически сложилось, что высокопроизводительные вычислительные системы, суперЭВМ воспринимаются как отражение позиций того или иного исследовательского коллектива, компании и даже целого государства в мировом «табеле о рангах» в сфере создания вычислительной техники. Академик РАН В. С. Бурцев считает [1], что суперкомпьютеры — один из

краеугольных камней в экономической независимости и национальной безопасности государства. Что, в частности, подтверждается его работой над вычислительными системами для противоракетной обороны, зенитно-ракетных комплексов в бывшем СССР, прежде всего над многопроцессорным комплексом «Эльбрус».

Военное применение осталось актуальным и сегодня, хотя суперкомпьютеры эксплуатируются уже и во многих других сферах:

- предсказания погоды, климата и глобальных изменений в атмосфере, геоинформационные системы;
- различные сферы материаловедения, физики, астрономии;
- генетика и биология;
- динамика жидкостей и газов;
- распознавание и синтез речи, изображений;
- широкий спектр экономических задач, связанных, прежде всего, с задачами линейного программирования, транспортные задачи и т. д.

С другой стороны, в работе [2] указывается, что сегодня имеет место проблема загрузки суперкомпьютеров, имеющая более важное значение, чем «выжимание» производительности из каждого заложенного в них терафлопса (Terra Floating Point Operations per Second).

Вначале остановимся на архитектуре современных суперкомпьютеров, при анализе которых и, прежде всего, выполняемых ими высокопроизводительных или высокопродуктивных вычислениях, необходимо рассматривать следующие факторы [2].

Во-первых, это многоядерные процессоры, без которых современные высокопродуктивные вычисления невозможны в принципе.

Во-вторых, архитектура больших вычислительных систем, построенных на базе многоядерных процессоров. Сегодня такие системы масштабируются до тысяч и десятков тысяч и процессоров, и ядер. Такое их количество переходит в определённое качество, что в свою очередь, выражается в сложности сопроводительных систем. Это становится отдельной проблемой. Возникают также проблемы, связанные с инфраструктурой потребления, охлаждения, мониторинга, уже упоминавшейся эффективности работы и множество иных задач.

В-третьих, системное и прикладное программное обеспечение, которое так или иначе упирается в параллельные алгоритмы.

На сайте [3] дается подробных анализ архитектуры многопроцессорных вычислительных систем, остановимся на их ключевых характеристиках, структурах и способах построения.

В 1966 году М.Флинном была предложена система классификации архитектур вычислительных систем, основанная на числе потоков инструкций и данных и включающая четыре архитектурных класса:

SISD = Single Instruction Single Data;

MISD = Multiple Instruction Single Data;

SIMD = Single Instruction Multiple Data;

MIMD = Multiple Instruction Multiple Data.

SISD (single instruction stream/single data stream) — одиночный поток команд и одиночный поток данных. К этому классу относятся уходящие в историю последовательные компьютерные системы, имеющие один центральный процессор, способный обрабатывать только один поток последовательно исполняемых инструкций. В случае векторных систем векторный поток данных следует рассматривать как поток из одиночных неделимых векторов. Примерами компьютеров с архитектурой SISD являются большинство рабочих станций Compaq, Hewlett-Packard и Sun Microsystems.

MISD (multiple instruction stream / single data stream) — множественный поток команд и одиночный поток данных. До сих пор ни одной реальной машины, попадающей в данный класс, не было создано.

SIMD (single instruction stream / multiple data stream) — одиночный поток команд и множественный поток данных. Эти системы обычно имеют большое количество процессоров, в пределах от 1024 до 16384, которые могут выполнять одну и ту же инструкцию относительно разных данных в жесткой конфигурации. Единственная инструкция параллельно выполняется над многими элементами данных. Примерами SIMD машин являются системы CPP DAP, Gamma II и Quadrics Apemille. Другим подклассом SIMD-систем являются векторные компьютеры. Векторные компьютеры манипулируют массивами сходных данных подобно тому, как скалярные машины обрабаты-

вают отдельные элементы таких массивов. Примерами систем подобного типа является, например, компьютеры Hitachi S3600.

MIMD (multiple instruction stream / multiple data stream) — множественный поток команд и множественный поток данных. команды и данные связаны и представляют различные части одной и той же выполняемой задачи. Наличие большого разнообразия систем данного класса, делает классификацию Флинна не полностью адекватной и заставляет использовать следующий подход к классификации. Множественный поток команд может быть обработан двумя способами: либо одним конвейерным устройством обработки, работающем в режиме разделения времени для отдельных потоков, либо каждый поток обрабатывается своим собственным устройством. Первая возможность используется в MIMD-компьютерах, которые обычно называют конвейерными или векторными, вторая — в параллельных компьютерах. В основе векторных компьютеров лежит концепция конвейеризации, т.е. явного сегментирования арифметического устройства на отдельные части. В основе параллельного компьютера лежит идея использования для решения одной задачи нескольких процессоров, работающих сообща, причем процессоры могут быть как скалярными, так и векторными.

SMP-архитектура (symmetric multiprocessing) — симметричная многопроцессорная архитектура. Главной особенностью систем с архитектурой SMP является наличие общей физической памяти, разделяемой всеми процессорами и используемой для передачи сообщений между процессорами. При этом все вычислительные устройства при обращении к ней имеют равные права и одну и ту же адресацию для всех ячеек памяти. Поэтому SMP-архитектура называется симметричной. Последнее обстоятельство позволяет очень эффективно обмениваться данными с другими вычислительными устройствами. SMP-система строится на основе высокоскоростной системной шины (SGI PowerPath, Sun Gigaplane, DEC TurboLaser), к слотам которой подключаются функциональные блоки трех типов: процессоры (ЦП), операционная система (ОП) и подсистема ввода/вывода (I/O). Для подсоединения к модулям I/O используются уже более медленные шины (PCI, VME64).

Наиболее известными SMP-системами являются SMP-сервера и рабочие станции на базе процессоров Intel (IBM, HP, Compaq, Dell, ALR, Unisys, DG, Fujitsu и др.) Вся система работает под управлением единой ОС (обычно UNIX-подобной, но для Intel-платформ поддерживается Windows NT).

Основные преимущества SMP-систем:

- простота и универсальность для программирования, использование общей памяти увеличивает скорость межпроцессорного обмена, пользователь также имеет доступ сразу ко всему объему памяти, существуют сравнительно эффективные средства автоматического распараллеливания;
- легкость в эксплуатации. Как правило, SMP-системы используют систему охлаждения, основанную на воздушном кондиционировании, что облегчает их техническое обслуживание.
- относительно невысокая цена.

Но SMP-системы плохо масштабируемы. Причины этого в том, что в данный момент шина способна обрабатывать только одну транзакцию, вследствие чего возникают проблемы разрешения конфликтов при одновременном обращении нескольких процессоров к одним и тем же областям общей физической памяти. В настоящее время конфликты могут происходить при наличии 8-24-х процессоров. Кроме того, системная шина имеет ограниченную (хоть и высокую) пропускную способность (ПС) и ограниченное число слотов. Для построения масштабируемых систем на базе SMP используются кластерные или NUMA-архитектуры.

MPP-архитектура (massive parallel processing) — массивно-параллельная архитектура. Главная особенность — память физически разделена. В этом случае система строится из отдельных модулей, содержащих процессор, локальный банк операционной памяти (ОП), два коммуникационных процессора (рутера) или сетевой адаптер, иногда — жесткие диски и/или другие устройства ввода/вывода. Один рутер используется для передачи команд, другой — для передачи данных. Доступ к банку ОП из такого модуля имеют только процессоры из этого же модуля. Модули соединяются коммуникационными каналами. Пользователь может определить логический номер процессора, к которому он подключен, и организовать обмен сооб-

щениями с другими процессорами. Используются два варианта работы операционной системы (ОС) на машинах MPP-архитектуры. В одном полноценная ОС работает только на управляющей машине (front-end), на каждом отдельном модуле работает сильно урезанный вариант ОС, обеспечивающий только работу расположенной в нем ветви параллельного приложения. Во втором варианте на каждом модуле работает полноценная UNIX-подобная ОС, устанавливаемая отдельно на каждом модуле.

Главным преимуществом систем с раздельной памятью является хорошая в отличие от SMP-систем масштабируемость: в машинах с раздельной памятью каждый процессор имеет доступ только к своей локальной памяти, в связи с чем не возникает необходимости в их потактовой синхронизации. Практически все рекорды по производительности на сегодняшний день устанавливаются на машинах именно такой архитектуры.

Недостатки:

- отсутствие общей памяти заметно снижает скорость межпроцессорного обмена, поскольку нет общей среды для обмена между процессорами;
- каждый процессор может использовать только ограниченный объем локального банка памяти.
- вследствие этого требуются значительные усилия для того, чтобы максимально использовать системные ресурсы.

Системами с раздельной памятью являются суперкомпьютеры MBC-1000, IBM RS/6000 SP, SGI/CRAY T3E, системы ASCI, Hitachi SR8000, системы Parsytec. Машины последней серии CRAY T3E от SGI, основанные на базе процессоров Dec Alpha 21164 с пиковой производительностью 1200 Мфлопс/с (CRAY T3E-1200), способны масштабироваться до 2048 процессоров.

При работе с MPP-системами используют Massive Passing Programming Paradigm — парадигму программирования с передачей данных (MPI, PVM, BSPlib).

Гибридная архитектура **NUMA** (nonuniform memory access). Главной ее особенностью является неоднородный доступ к памяти. Гибридная архитектура воплощает в себе удобства систем с общей памятью и относительную дешевизну систем с раздельной памятью, а именно: память является физически распределенной по

различным частям системы, но логически разделяемой, так что пользователь видит единое адресное пространство. Система состоит из однородных базовых модулей, объединенных с помощью высокоскоростного коммутатора. По существу архитектура NUMA является MPP-архитектурой, где в качестве отдельных вычислительных элементов берутся SMP-узлы. Впервые идею гибридной архитектуры предложил С. Воллох и воплотил в системах серии Exemplar. Фирма HP купила идею и реализовала на суперкомпьютерах серии SPP. Идею подхватил С. Крей и добавил новый элемент — когерентный кэш, создав архитектуру **cc-NUMA** (Cache Coherent Non-Uniform Memory Access), которая расшифровывается как «неоднородный доступ к памяти с обеспечением когерентности кэшей». Он ее реализовал на системах Origin.

Наиболее известными системами архитектуры cc-NUMA являются: HP 9000 V-class в SCA-конфигурациях, SGI Origin3000, Sun HPC 15000, IBM/Sequent NUMA-Q 2000. На настоящий момент максимальное число процессоров в cc-NUMA-системах может превышать 1000 (серия Origin3000). Обычно вся система работает под управлением единой ОС, как в SMP. Возможны также варианты динамического «подразделения» системы, когда отдельные «разделы» системы работают под управлением разных ОС.

PVP-архитектура (Parallel Vector Process) — параллельная архитектура с векторными процессорами. Основным признаком PVP-систем является наличие специальных векторно-конвейерных процессоров, в которых предусмотрены команды однотипной обработки векторов независимых данных. Поскольку передача данных в векторном формате осуществляется намного быстрее, чем в скалярном, то проблема взаимодействия между потоками данных при распараллеливании становится несущественной. И то, что плохо распараллеливается на скалярных машинах, хорошо распараллеливается на векторных. Таким образом, системы PVP-архитектуры могут являться машинами общего назначения (general purpose systems). Однако, поскольку векторные процессоры весьма дороги, эти машины вряд ли станут общедоступными.

Наиболее популярны следующие 3 машины PVP-архитектуры.

- CRAY SV-2, SMP-архитектура [4]. Пиковая производительность системы в стандартной конфигурации может составлять десятки терафлопс;
- NEC SX-6, NUMA-архитектура [5]. Пиковая производительность системы может достигать 8 Тфлопс, производительность 1 процессора составляет 8 Гфлопс, система масштабируется до 128 узлов;
- Fujitsu-VPP5000 (vector parallel processing), MPP-архитектура [6]. Производительность 1 процессора составляет 9.6 Гфлопс, пиковая производительность системы может достигать 1249 Гфлопс, максимальная емкость памяти — 8 Тб. Система масштабируется до 512 узлов.

Кластерная архитектура. Кластер представляет собой два или больше компьютеров, часто называемых узлами, объединяемых при помощи сетевых технологий на базе шинной архитектуры или коммутатора и предстающих перед пользователями в качестве единого информационно-вычислительного ресурса. В качестве узлов могут быть выбраны серверы, рабочие станции и даже обычные персональные компьютеры. Преимущество кластеризации для повышения работоспособности становится очевидным в случае сбоя какого-либо узла: когда другой узел кластера может взять на себя нагрузку неисправного узла, и пользователи не заметят прерывания в доступе. Возможности масштабируемости кластеров позволяют многократно увеличивать производительность приложений для большего числа пользователей технологий Fast/Gigabit Ethernet [7], Myrinet [8]. Такие суперкомпьютерные системы являются самыми дешевыми, поскольку собираются на базе стандартных комплектующих элементов процессоров, коммутаторов, дисков и внешних устройств. Кластеризация может быть осуществлена на разных уровнях компьютерной системы, включая аппаратное обеспечение, операционные системы, программы-утилиты, системы управления и приложения. Вариирование построения кластера, выбор набора элементов позволяет существенно изменять его стоимость, а также доступность комплектующих.

Очевидно, что разработка кластеров является сложным процессом, требующим согласований таких вопросов как инстал-

ляция, эксплуатация и одновременное управление большим числом компьютеров, технические требования параллельного и высокопроизводительного доступа к одному и тому же системному файлу (или файлам) и межпроцессорная связь между узлами и координация работы в параллельном режиме. Эти проблемы проще всего решаются при обеспечении единого образа операционной системы для всего кластера. Однако реализовать подобную схему удаётся далеко не всегда и обычно она применяется лишь для не слишком больших систем.

При этом следует согласиться с [2], что архитектура кластерной системы в большей степени определяет ее производительность, чем тип используемых в ней процессоров. Критическим параметром, влияющим на величину производительности такой системы, является расстояние между процессорами. Так, соединив вместе 10 персональных компьютеров, мы получим систему для проведения высокопроизводительных вычислений, проблема, однако, будет состоять в нахождении наиболее эффективного способа соединения их друг с другом, поскольку при увеличении производительности каждого процессора в 10 раз производительность системы в целом в 10 раз не увеличится. В частности, может оказаться [2] более рентабельным создать систему из большего числа дешевых компьютеров, чем из меньшего числа дорогих.

В кластерах, как правило, используются стандартные операционные системы, чаще всего, свободно распространяемые — Linux [9], FreeBSD [10], вместе со специальными средствами поддержки параллельного программирования и балансировки нагрузки.

Принципы построения коммуникационных сред [3]. В последнее время, исходя из производительности и скорости передачи данных, а также из стоимости необходимого оборудования, учета свойств масштабируемости и других параметров при создании многопроцессорных вычислительных систем часто используются технологиям SCI, Myrinet или Raceway.

Технология Myrinet основана на использовании многопортовых коммутаторов при ограниченных несколькими метрами длинах связей узлов с портами коммутатора. Узлы в Myrinet соединяются друг с другом через коммутатор (до 16 портов). Как коммутируемая сеть, аналогичная по структуре сегментам Ethernet, соединен-

ным с помощью коммутаторов, Myrinet может одновременно передавать несколько пакетов, каждый из которых идет со скоростью, близкой к 2 Гбит/с. В отличие от некоммутированных Ethernet и FDDI-сетей совокупная пропускная способность сети Myrinet возрастает с увеличением количества машин.

Myrinet чаще всего используют как локальную сеть (LAN) сравнительно небольшого размера, связывая вместе компьютеры внутри комнаты или здания. Из-за своей высокой скорости, малого времени задержки, прямой коммутации и умеренной стоимости, Myrinet особенно популярен для объединения компьютеров в кластеры. Myrinet также используется как системная сеть (System Area Network, SAN), которая может объединять компьютеры в кластер внутри стойки с той же производительностью, но с более низкой стоимостью, чем Myrinet LAN. Пакеты Myrinet могут иметь любую длину и могут включать другие типы пакетов, включая IP-пакеты.

Myrinet является открытым стандартом, компания Myricom предлагает широкий выбор сетевого оборудования по сравнительно невысоким ценам. Реализована программная поддержка драйверов для Linux (Alpha, x86, PowerPC, UltraSPARC), Windows NT (x86), Solaris (x86, UltraSPARC) и Tru64 UNIX.

Коммуникационная среда Raceway обеспечивает пропускную способность на уровне 1 Гбайт/с; создается с помощью коммутатора Cypress и соответствующих сетевых адаптеров. Коммутатор имеет 6 портов, пропускная способность каждого порта составляет 160 Мбайт/с. Структуры вычислительных систем, создаваемых при помощи Raceway, аналогичны тем, которые применяются при использования Myrinet или коммутаторов и адаптеров SCI. Разница заключается в количестве портов коммутаторов, форматах передаваемых пакетов и в протоколах. Raceway принята в качестве стандарта ANSI/VINA 5-1994.

Требования к многопроцессорным вычислительным системам [3].

Отношение стоимость/производительность. Добиться повышения производительности в МВС тяжелее, чем произвести масштабирование внутри узла. Основным барьером является организации эффективных межузловых связей. Коммуникации между узлами, должны быть устойчивы к большим задер-

жкам программно поддерживаемой когерентности. Приложения с большим количеством взаимодействующих процессов работают лучше на основе SMP-узлов, в которых коммуникационные связи более быстрые. В кластерах, как и в MPP-системах, масштабирование приложений более эффективно при уменьшении объема коммуникаций между процессами, работающими в разных узлах. Это обычно достигается путем разбиения данных.

Именно такой подход используется в наиболее известном приложении на основе кластеров OPS (Oracle Parallel Server).

Масштабируемость. Возможность масштабирования системы определяется не только архитектурой аппаратных средств, но зависит от программного обеспечения. Его масштабируемость затрагивает все уровни от простых механизмов передачи сообщений до работы с такими сложными объектами как мониторы транзакций и вся среда прикладной системы. В частности, программное обеспечение должно минимизировать трафик межпроцессорного обмена, который может препятствовать линейному росту производительности системы. В этом смысле, аппаратные средства являются только частью масштабируемой архитектуры и простой переход на более мощный процессор может привести к перегрузке других компонентов системы.

Возможность масштабирования, например, кластера ограничена значением отношения скорости процессора к скорости связи, которое не должно быть слишком большим, хотя, последние 10 лет показывают, что разрыв по скорости между ними все увеличивается. Добавление каждого нового процессора в действительно масштабируемой системе должно давать прогнозируемое увеличение производительности и пропускной способности при приемлемых затратах. Тем самым, одной из основных задач при построении масштабируемых систем является минимизация стоимости расширения компьютера и упрощение планирования. В действительности реальное увеличение производительности трудно оценить заранее, поскольку оно в значительной степени зависит от прикладных задач. Другими словами получаем, что действительно масштабируемая система должна быть сбалансирована по всем параметрам — стоимость, производительность, решаемые прикладные задачи.

Суперкомпьютеры Cray Research и высокопроизводительные мейнфреймы IBM относятся к наиболее дорогим категориям компьютеров. Крайним примером, где производительность принесена в жертву для достижения низкой стоимости, являются персональные компьютеры IBM PC. Между этими двумя границами находятся иные конструкции, основанные на балансе стоимости и производительности, кот орый учитывает и требования, следующие из задач, решаемых с помощью МВС — научных, экономических и т.п., непосредственно влияющих на ее архитектуру.

Совместимость и мобильность программного обеспечения. В настоящее время одной из тенденций информационных технологий является ориентация на рынок прикладных программных средств, так как для конечного пользователя, в конце концов, важно программное обеспечение, позволяющее решить его задачи, а не выбор той или иной аппаратной платформы. Переход к неоднородным сетям компьютеров в корне изменил и точку зрения на саму сеть — она превратилась в средство интеграции отдельных распределенных ресурсов, каждый из которых соответствует требованиям конкретной прикладной задачи. Этот переход выдвинул ряд новых требований.

Прежде всего, такая вычислительная среда должна позволять гибко менять количество и состав аппаратных средств и программного обеспечения в соответствии с меняющимися требованиями решаемых задач. Во-вторых, она должна обеспечивать возможность запуска одних и тех же программных систем на различных аппаратных платформах. В-третьих, эта среда должна гарантировать возможность применения одних и тех же человеко-машинных интерфейсов на всех компьютерах, входящих в неоднородную сеть. Это привело к созданию совокупности (открытых) стандартов на компоненты вычислительной среды для обеспечения мобильности программных средств в рамках неоднородной, распределенной вычислительной системы. Одним из вариантов моделей открытой среды является модель OSE (Open System Environment), предложенная комитетом IEEE POSIX.

Надежность и отказоустойчивость МВС. Главной целью является целостность хранимых и обрабатываемых данных. Повышение надежности основано на применении схем и ком-

понентов с высокой и сверхвысокой степенью интеграции, снижении уровня помех, обеспечении облегченных, в том числе тепловых, режимов работы аппаратуры. Надежность относится и к используемому программному обеспечению, которое должно позволять организовать эффективную службу сопровождения и мониторинга решений.

Повышение надежности и отказоустойчивости так или иначе предполагает наличие избыточного аппаратного и программного обеспечения. Здесь впрямую выражены концепции параллельности вычислительных систем, на которых достигается как наиболее высокая производительность, так и, во многих случаях, очень высокая надежность. Имеющиеся ресурсы избыточности в параллельных системах могут использоваться как для повышения производительности, так и для повышения надежности. В то же время эта избыточность ведет к увеличению, прежде всего, эксплуатационных энергетических затрат, которые весьма значительны для МВС и их снижение — одна из ключевых задач эффективной работы системы.

Кластеры являются, пожалуй, оптимальной схемой повышения надёжности — благодаря единому представлению, отдельные узлы или компоненты кластера могут подменять элементы, обеспечивая непрерывность и безотказную работу.

Таким образом, основные требования к МВС образуются исходя из потребностей решаемых на них прикладных задач, а также включают общие для всех требования масштабируемости, надежности, соотношения стоимости/производительности. К этим требованиям следует добавить эксплуатационные требования, вытекающие из значительных энергозатрат МВС в процессе их функционирования. Последние также обусловлены известными проблемами мониторинга и восстановления вычислительного процесса, в частности, в ситуациях, когда система решали задачу в течение длительного промежутка времени и требует значительных временных затрат для сохранения «слепка» ее состояния в ситуации останова, вызванного аварийными или иными обстоятельствами. Такая ситуация помимо вычислительных проблем, затрат времени неизбежно ведет и к простому выбрасыванию средств, затраченных на работу вычислительного комплекса в случае повторного запуска вычислений.

Эти и близкие проблемы рассматриваются в рамках направления «зеленых» вычислений. Остановимся на их общей характеристике.

«Зеленые вычисления». В стремлении к повышению производительности создатели суперкомпьютеров игнорируют побочные эффекты в виде чрезмерного энергопотребления и необходимости дополнительного отвода тепла, что влечет за собой ограничение этой самой производительности [11]. С 1992 года производительность суперкомпьютеров выросла в 10 тыс. раз, хотя в пересчете на единицу потребляемой мощности за то же самое время она увеличилась лишь в 300 раз, а в пересчете на единицу занимаемой площади — в 65 раз. В результате приходится строить новые машинные залы, а то и новые здания.

Например, потребляемая мощность наиболее производительных суперкомпьютеров из списка Top500 [12] достигает 10 мегаватт [11]. Этого достаточно, чтобы поддерживать жизнедеятельность города с населением в 40 тыс. человек. И это несмотря на то, что системы строятся из компонентов с низким энергопотреблением, общая потребляемая мощность все равно измеряется мегаваттами. Стоимость эксплуатации компьютеров в пересчете на мегаватт потребляемой мощности составляет от 200 тыс. до 1,2 млн долл. в год тоже весьма существенна. Тем самым, потребляемая мощность стала фактором, приведшим к переосмысливанию архитектуры МВС.

Первым доступным суперкомпьютером с эффективным энергопотреблением стал Green Destiny (2002 г., Лос-Аламос), который представлял собой 240-процессорный кластер и занимал около 0,5 кв. м. При бездисковой загрузке [13, 14] его потребляемая мощность составляла 3,2 кВт. С рейтингом Linpack [15] в 101 GFLOPS машина хотя и занимала на тот момент 393-ю строчку в рейтинге Top500, но привлекла к себе широкое внимание. Можно сказать, что она стала первым существенным примером и стимулом снижения энергопотребления, в чем-то вместе с программой США Energy Star [16] инициировав название «зеленых вычислений». В то же время, параметры чистой производительности, хотя и оказали существенное влияние на конструкцию современных высокопроизводительных компьютерных систем, не имеют никакого отношения к энергетической эффективности, что привело к задаче выработки

новых параметров сравнения МВС и их сравнения как по производительности, так и по эффективности их энергопотребления — формирование рейтингов Green500 [17] и Top500.

В [18] представлены результаты сравнения рейтингов Green500 и Top500 восьми суперкомпьютеров, оценки их производительности и максимального энергопотребления, а также значения параметров, отражающих соотношение производительности к пиковой потребляемой мощности (эффективность максимального энергопотребления) и результаты выполнения тестов Linpack (эффективность среднего энергопотребления).

Несмотря на приведенные оценки, вопрос оценивания и сравнения МВС в целом остается неоднозначным. Это связано, в частности, с различными системами и тестами измерения — повсеместно принятой, несмотря на отмечаемые ее недостатки, чисто «производительной» характеристикой Linpack, тесты SPEChpc [19] и HPC Challenge [20], и другие, расширяющие методологию Green500. Помимо этого возникают вопросы оценки суммарных мощностей различных системных узлов, систем охлаждения и т.д., которые в целом позволяют более точно определить операционные расходы в связи с надежностью систем, ежегодные затраты на потребляемую электроэнергию которых приближаются к затратам на закупку самих суперкомпьютеров.

Есть достаточно известных способов [21] снизить потребление электроэнергии центрами обработки данных (ЦОД) и настольными ПК — обесточивание неиспользуемого оборудования, управление питанием с помощью функций операционных систем и аппаратуры, дедупликация данных и т.д. Регулирование мощности процессоров было реализовано еще в начале нынешнего десятилетия, но на практике применяется нечасто, что, возможно, связано с поддержкой этой технологии на уровне аппаратуры и операционной системы. Но регулирование мощности процессоров позволяет сэкономить 14% энергии, а вложенные средства дают наибольший доход. Как показало исследование компании Enterprise Management Associates, виртуализация является наиболее популярной технологией «зеленых» ИТ. Проекты консолидации ЦОД с применением технологии гипервизора производства VMware, Microsoft, Citrix Systems, Red Hat, Novell или Xen уже показали

возможности экономии на оборудовании и повышения эффективности использования ресурсов ИТ.

Но виртуальные ресурсы все равно требуют физическое оборудование, которое нуждается в энергии и охлаждении. Рост числа виртуальных машин и их неконтролируемое использование могут привести к тому, что добиться экономии энергии не удастся. В частности, надо сразу же удалять неиспользуемые виртуальные машины и сформулировать строгие правила управления жизненным циклом виртуальных машин. Без эффективного планирования и управления виртуализация может стать причиной того, что расходы на электроэнергию останутся на прежнем уровне или даже возрастут [21].

Тем самым, «зеленые» ИТ требуют целостного подхода к управлению, увязывающего воедино потребности бизнеса и планирование ресурсов. Эта целостность упирается и в способы измерения эффективности, к которым, к примеру, относится и цена производительности на 1 Вт, определение которой требует отдельного исследования, являясь и наукой, и искусством. Например, компания Green Grid [21, 22] опубликовала «Критерии оценки продуктивности расходования электроэнергии в ЦОД», где рассматриваются доводы «за» и «против» использования процессорной нагрузки при определении объема вычислений на 1 Вт.

Таким образом, к требованиям к МВС следует относить:

- масштабируемость;
- совместимость и мобильность программного обеспечения;
- надежность МВС;
- отношение стоимость/производительность, к которому будем добавлять эффективность энергопотребления, включая остальные эксплуатационные расходы.

Реализация этих требований, как видим, предполагает проведение системного исследования архитектуры МВС еще на этапе предварительного проведения, построения и использования соответствующих моделей, методов, алгоритмических и программных средств. Детальные постановки задач построения таких средств, их реализация будут рассмотрены в последующих работах автора.

* * *

1. http://oss.mexmat.sgu.ru/article-burcev.
2. http://ru.intel.com/business/community/index.php?automodule=blog&blogid=182&showentry=421.
3. http://www.informika.ru/text/teach/topolog/2.htm.
4. http://www.cray.com/products/systems/sv2.
5. http://www.nec.com.au/hpcsd/vector.htm.
6. http://www.fse.fujitsu.com/products/vpp5000.html.
7. http://parallel.ru/computers/interconnects.html#fast_ethernet.
8. http://www.myri.com.
9. http://www.linux.org.
10. http://www.freebsd.org.
11. http://www.osp.ru/os/2008/01/4839411.
12. http:/www.top500.org
13. Feng W., Making a Case for Efficient Supercomputing. ACM Queue, Oct. 2003.
14. Feng W., The Importance of Being Low Power in High-Performance Computing. Cyberinfrastructure Technology Watch, Aug. 2005.
15. http://www.netlib.org/linpack.
16. http://www.energystar.gov.
17. http://www.green500.org.
18. http://www.osp.ru/data/866/405/1239/041-b.gif.
19.M. Mueller, Overview of SPEC HPC Benchmarks. BOF presentation, ACM/IEEE SC Conf., 2006.
20.J. Dongarra, P. Luszczek, Introduction to the HPC Challenge Benchmark Suite. Tech. report, Univ. Tennessee, 2004; www.cs.utk.edu/~luszczek/pubs/hpcc-challenge-intro.pdf.
21.http://www.pcweek.ru/themes/detail.php?ID=116466&THEME_ID =13878.
22. http://thegreengrid.org.

Отримано: 18.09.2009 р.