УДК 681.324

П.Н. Филиппенко, А.А. Шашелов, С.В. Сеитова

ПОСТРОЕНИЕ СИСТЕМ, ОБРАБАТЫВАЮЩИХ БОЛЬШИЕ ВЫЧИСЛЕНИЯ: ПРОБЛЕМЫ И ТЕНДЕНЦИИ

Рассмотрены проблемы оптимизации больших вычислений. Приводится краткая история развития систем больших вычислений, показаны результаты изменения значимости положения занимаемого кластерными системами в ряду лучших современных вычислительных систем за последние годы. Рассмотрены проблемы построения мощных вычислительных систем и способы решения этих проблем. Проведен анализ тенденций развития систем обработки больших вычислений.

Многопроцессорная система; кластер; архитектура; параллельные вычисления; производительность.

P.N. Filippenko, A.A. Shashelov, S.V. Seitova

CONSTRUCTION OF SYSTEM ARE PROCESSING THE BIG CALCULATIONS: PROBLEMS AND TENDENCIES

In this article are considered problems of optimization of the big computing calculations. There are short history of clusters systems, results of modification the importance position that cluster's systems occupied among the best modern computing systems during the last year. Appointment, constructions, problems of powerful computing systems and ways of their decision are considered here. The progress analysis is carried out also in this article.

Multiprocessing system; cluster; architecture; parallel calculations; productivity.

Введение. Круг задач, требующих для своего решения применения мощных вычислительных ресурсов, постоянно расширяется. Вследствие широкого внедрения вычислительной техники, усилилось внимание к численному моделированию и численному эксперименту. Численное моделирование, заполняя промежуток между физическими экспериментами и аналитическими подходами, позволило изучать явления, которые являются либо сложными для исследования аналитическими методами, либо дорогостоящими или опасными для экспериментального изучения. При этом численный эксперимент позволил удешевить процесс научного и технологического поиска. Стало возможным моделировать в реальном времени процессы интенсивных физико-химических и ядерных реакций, глобальные атмосферные процессы, процессы экономического и промышленного развития регионов и т.д. Очевидно, что решение таких масштабных задач требует значительных вычислительных ресурсов. Мощности современных процессоров вполне достаточно для решения элементарных шагов большинства задач, а объединение нескольких десятков таких процессоров позволяет быстро и эффективно решать многие поставленные задачи, не прибегая к помощи мэйнфреймов и суперкомпьютеров [1-3].

Развитие систем автоматизации больших вычислений. Слова «кластер» и «суперкомпьютер» в значительной степени синонимы, но прежде чем об этом стало можно с уверенностью говорить, аппаратные средства прошли длительный цикл эволюции. В течение первых 30 лет с момента появления компьютеров, вплоть до середины 1980-х гг., под «суперкомпьютерными» технологиями понимали исключительно производство специализированных особо мощных процессоров. Однако появление однокристального микропроцессора практически стерло разницу между «массовыми» и «особо мощными» процессорами, и с этого момента единственным способом создания суперкомпьютера стал путь объединения процессоров для параллельного решения одной задачи. Основное направление развития высокопроизводительных компьютерных технологий до середины 1990х годов было связано с построением специализированных многопроцессорных систем из массовых микросхем. Один из сформировавшихся подходов построение систем – SMP (Symmetric Multi Processing), подразумевал объединение многих процессоров с использованием общей памяти, что предъявляло высокие требования к быстродействию памяти, хотя и упрощало процесс программирования. Было практически невозможно сохранить быстродействие системы при даже небольшом увеличении количества узлов. Подход SMP оказался дорогим в аппаратной реализации. На порядок дешевле и практически бесконечно масштабируемым оказался подход MPP (Massively Parallel Processing), при котором независимые специализированные вычислительные модули объединялись специализированными каналами связи, причем, и те, и другие создавались под конкретный суперкомпьютер и ни в каких других целях не применялись [1-3].

Идея создания, так называемого, кластера рабочих станций фактически явилась развитием подхода MPP, ведь логически MPP-система не сильно отличалась от обычной локальной сети. Локальная сеть стандартных персональных компьютеров, при соответствующем программном обеспечении (ПО) использовавшаяся как многопроцессорный суперкомпьютер, и стала прародительницей современного кластера. Полноценную MPP-систему можно создать из стандартных серийных компьютеров при помощи серийных коммуникационных технологий, причем такая система обходилась дешевле в среднем на два порядка[2].

Самые знаменитые компьютеры с кластерной архитектурой «первого поколения»: Beowulf (1994, NASA Goddard Space Flight Center) – 16-процессорный кластер на процессорах Intel 486DX4/100 МГц; Avalon (1998, Лос-Аламосская национальная лаборатория) – Linux-кластер на базе процессоров Alpha 21164A/533 МГц [6].

Одной из основных характеристик многопроцессорной систем является производительность. Различают пиковую и реальную производительность. Пиковая производительность многопроцессорной системы (кластера, SMP-системы и т. д.) это теоретическое значение, недостижимое на практике. Оно получается умножением пиковой производительности процессора на число процессоров в системе. Пиковая производительность, в общем случае, рассчитывается путем умножения тактовой частоты ЦП на максимальное число операций, выполняемых за один такт. Реальная производительность многопроцессорной системы это производительность, полученная при решении реальной задачи (академической или промышленной). Например, системы в рейтинге самых производительных вычислительных систем ранжируются по результатам теста LINPACK, который является реальной академической задачей обеспечивающей решение системы линейных уравнений [5].

Бурное развитие кластерных технологий за последние годы хорошо видно из анализа списка самых производительных вычислительных систем: с 2000 по 2010 г. доля кластеров в списке увеличилась с 2,2 до 83,0 % [6] (рис. 1).

Самый мощный кластер в списке самых производительных вычислительных систем находится на 3м месте списка: это Roadrunner Dawning TC3600 Blade System Dawning Cluster включает 120640 ядер (процессоры Intel EM64T Xeon X56xx (Westmere-EP) 2660 MHz). По рейтингу на ноябрь 2010 г. на 62-ом месте находится самая мощнейшая из российских кластерная платформа MVS-100K – Cluster Platform 3000 BL460c/BL2x220, Xeon 54xx 3 Ghz, Infiniband производства Hewlett-Packard. Это решение находится в Межведомственном суперкомьютерном центр Российской академии наук включает 2920 процессоров Intel Xeon x5400 (11680 ядра) и работает под Linux [6].

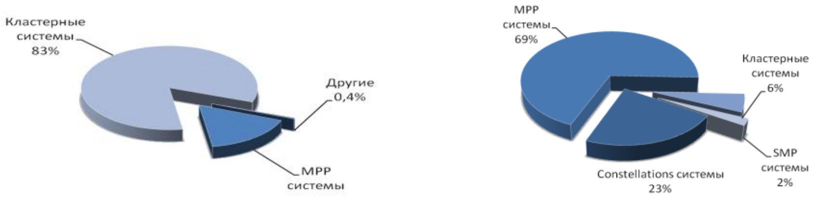

Рис. 1. Доля кластеров в списке лучших суперкомпьютеров данные на ноябрь 2010 г [6]

Кластерные системы успешно применяются для решения любых задач – от расчетов для науки и промышленности до управления базами данных. Практически любые приложения, требующие высокопроизводительных вычислений, имеют сейчас “параллельные” версии, которые позволяют разбивать задачу на фрагменты и обсчитывать ее параллельно на многих узлах кластера. Например, для инженерных расчетов (прочностные расчеты, аэромеханика, гидро- и газодинамика) традиционно применяются так называемые сеточные методы, когда область вычислений разбивается на ячейки, каждая из которых становится отдельной единицей вычислений. Эти ячейки обсчитываются независимо на разных узлах кластера, а для получения общей картины на каждом шаге вычислений происходит обмен данными, распространенными в пограничных областях [6].

Проблемы построения и способы их решения. Архитектура кластера должна обеспечивать масштабируемость ПО при увеличении количества узлов, т.е. прирост производительности при добавлении новых вычислительных модулей. Для этого важно правильно выбрать конфигурацию кластера в зависимости от профиля обмена данными между экземплярами программы, запущенными на разных узлах. Здесь нужно учитывать общий объем пересылаемых данных, распределение длин сообщений, использование групповых операций и т.п. Архитектура таких вычислительных систем представляет собой множество однородных по аппаратуре и настройкам системного ПО модулей, объединенных специальной высокопроизводительной коммуникационной средой, имеющих единый центр доступа, администрирования, мониторинга и работающих под управление специального системного программного обеспечения [1-4]. Благодаря открытости и модульности архитектуры, использования распространенных на рынке компонент (процессоров, оперативной памяти и т.д.), свободно распространяемого ПО такие вычислительные системы имеют рад преимуществ: высокая производительность, эффективность, простота наращиваемости мощностей, простота эксплуатации и обслуживания и т.д. Для достижения необходимой производительности возможно объединение в единые вычислительные системы компьютеров самого разного типа, начиная от персональных компьютеров и заканчивая мощными суперкомпьютерами [6]. Прирост производительности при добавлении новых вычислительных модулей обеспечивается правильным составлением конфигурации кластера в зависимости от профиля обмена данными между экземплярами программы, запущенными на разных узлах. Здесь нужно учитывать общий объем пересылаемых данных, распределение длин сообщений, использование групповых операций и т.п. Привлекательным свойством кластерных вычислительных систем является оптимальное соотношения цена/производительность [3-4].

Задача построения кластера не ограничивается объединением большого количества процессоров в один сегмент. Для того чтобы на практике получить описанные преимущества при использовании вычислительной системы для решения задач конкретной прикладной области на конкретном программном комплексе необходимо решить ряд задач еще на этапе проектирования вычислительной системы [7]. Так архитектура вычислительного кластера должна учитывать особенности, используемой прикладным пакетом модели вычислительной системы, параметры аппаратуры кластера должны быть согласованны с требованиями прикладного пакета к производительности узлов и характеристикам коммуникационного оборудования, используемые системные библиотеки, установленные на кластере должны поддерживать оптимизации прикладного пакета и т.д.

Параметры аппаратуры кластера должны быть согласованны с требованиями прикладного пакета к производительности узлов и характеристикам коммуникационного оборудования. Для некоторых хорошо распараллеливаемых задач (таких, как рендеринг независимых сюжетов в видеофрагменте) основной фактор быстродействия – мощный процессор, и производительность интерконнекта не играет основной роли. В то же время, для некоторых других задач важна производительность системной сети, а увеличение числа узлов в кластере будет мало влиять на скорость решения задачи [7-8].

Чтобы задействовать вычислительные мощности кластера нужно либо запускать множество однопроцессорных задач либо использовать параллельные программы. Использовать однопроцессорные задачи может быть разумным вариантом, если нужно провести множество независимых вычислительных экспериментов с разными входными данными, причем срок проведения каждого отдельного расчета не имеет значения, а все данные размещаются в объеме памяти, доступном одному процессу. Второй способ – запускать готовые параллельные программы или создавать собственные параллельные программы [9]. Это трудоемкий, но универсальный способ.

Для ускорения проведения вычислений на кластере необходимо максимально ускорить вычисления на одном процессоре, для чего применяются следующие оптимизационные решения [9-10]:

- Подбор опций оптимизации компилятора.
- Использование оптимизированных библиотек. Если некоторые стандартные действия, такие как умножение матриц, занимают значительную долю времени работы программы, то имеет смысл использовать готовые оптимизированные процедуры, выполняющие эти действия, а не программировать их самостоятельно.
- Исключение свопинга (автоматического сброса данных из памяти на диск). Каждый процесс должен хранить не больше данных, чем для него доступно оперативной памяти.
- Оптимальное использование кэш-памяти. В случае возможности изменять последовательность действий программы, нужно модифицировать программу так, чтобы действия над одними и теми же, или подряд расположенными данными выполнялись также подряд.
- Оптимальная работа с временными файлами. Например, если программа создает временные файлы в текущем каталоге, то разумнее будет перейти на использование локальных дисков на узлах.

Для ускорения работы программного обеспечения основанного на принципах параллельных вычислений стоит принять меры для снижения накладных расходов на синхронизацию и обмены данными. Возможно, приемлемым подходом окажется совмещение асинхронных пересылок и вычислений. Для исключения простоя отдельных процессоров нужно равномерно распределить вычисления между процессами, причем в некоторых случаях может понадобиться динамическая балансировка. Важным показателем, который говорит о том, эффективно ли в программе реализован параллелизм, является загрузка вычислительных узлов, на которых работает программа. Если загрузка на всех или на части узлов далека от 85%, то это значит, что программа неэффективно использует вычислительные ресурсы, т.е. создает большие накладные расходы на обмены данными или неравномерно распределяет вычисления между процессами [9-10].

Тенденции развития вычислительных систем. На сегодняшний день, как отечественными, так и зарубежными производителями продолжается деятельность по исследованию и созданию перспективных кластерных систем с повышенной эффективностью на широком классе задач.

Исследования ведутся по следующим направлениям [6,11-14]:

- выделение типовых проблем, возникающих при решении задач на современных вычислительных системах и обуславливающие их низкую реальную производительность, поиски путей их решения за счет использования мультитредовых архитектур и соответствующих программ;
- разработка принципов организации мультитредовых процессоров с разной организацией (архитектура с управлением потоком данных – DF; мультитредовая архитектура – MT; параллельная мультитредовая архитектура – SMT, мультитредовая архитектура с управлением потоком данных – MT/DF или SMT/DF; чип-мультитредовая архитектура – CMP; архитектура процессоров внутри кристалла памяти – PIM);
- разработка принципов организации коммуникационных сред с высокой пропускной способностью и малой задержкой передачи сообщений для мультитредовых вычислительных систем;
- разработка принципов организации мультитредовых систем с распределенной разделяемой памятью и динамической балансировкой загрузки процессоров;
- разработка принципов организации компиляторов языков программирования для мультитредовых процессоров и систем, обеспечивающих статическое и динамическое автоматическое распараллеливание;
- разработка принципов организации исполняемых мультитредовых программ;
- разработка принципов построения систем предобработки больших объемов сигнальной информации в реальном времени.

Исследования предложенных архитектур ведутся на параллельных имитационных моделях. Необходимость их усовершенствования возникает при проработке вопросов: создания операционной системы, повышении возможностей эффективного межтредового взаимодействия (это необходимо, например, при решении задачи эмуляции на этих машинах архитектур других систем), аппаратной поддержке обнаружения информационной зависимости параллельно выполняемых участков программ при использовании современных методов динамического распараллеливания и др. Особое место занимают вопросы обеспечения отказоустойчивости, снижения потребляемой энергии [10].

Заключение. Разработка и исследование систем обработки больших вычислений, в частности перспективных в этом направлении – кластерных систем, требует: понимания внутренней архитектуры прикладного пакета; использования вычислительных параллельных алгоритмов; исследования современных технологий (многоядерных микропроцессоров, многопроцессорных платформ, высокопроизводительных сетевых технологий и т.д.); разработки аналитических моделей позволяющих прогнозировать характеристики вычислительной системы для конкретного прикладного продукта; практического опыта построения таких систем. Кластерные системы, являясь альтернативой дорогим SMP-серверам, обладают рядом конкурентных преимуществ: лучший показатель цена/производительность, высокая вычислительная производительность, возможность параллельных вычислений, хорошая гибкость и масштабируемость системы, возможность сочетания в системе узлов с разной частотой процессоров (что не допустимо в SMP-системах). Такие системы имеют перспективы развития и использования для вычислительных и параллельных задач в научных, а также инженерных расчетах. Системы обработки больших вычислений перспективны как с точки зрения интересов и задач пользователей, так и с точки зрения развития технологической базы таких систем.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Воеводин В.В., Жуматий С.А.* Вычислительное дело и кластерные системы. – М.: Изд-во МГУ, 2007. – С. 35-42.
2. *Букатов А.А., Дацюк В.Н., Жегуло А.И.* Программирование многопроцессорных вычислительных систем. – Ростов-на-Дону: Изд-во ООО «ЦВВР», 2003. – С. 123-127.
3. *Филиппенко П.Н.* Обзор в области построения и использования кластерных систем // Информатика, вычислительная техника и инженерное образование. – Таганрог: Изд-во ТТИ ЮФУ. – 2010. – № 1. – С. 19-27.
4. *Воеводин В.В., Воеводин В.В.* Параллельные вычисления. – СПб.: БХВ-Петербург, 2002. – С. 155-160.
5. Top 500 supercomputers sites [электронных ресурс]. – URL: http://www.top500.org/ (дата обращения: 2.12.2010).
6. Суперкомпьютерные технологии в науке, образовании и промышленности / Под редакцией: академика В.А. Садовничего, академика Г.И. Савина. – М.: Изд-во МГУ, 2009. – С. 232-234.
7. *Бройдо В.Л., Ильина О.П.* Архитектура ЭВМ и систем: Учебник для вузов. 2-е изд. – СПб.: Изд-во Питер, 2009. – С. 154-157.
8. *Таненбаум Э.* Архитектура компьютера. 5-е изд. – СПб.: Изд-во Питер, 2010. – С. 34-42.
9. *Шпаковский Г.И., Серикова Н.В.* Программирование для многопроцессорных систем в стандарте MPI: Пособие, – Минск: БГУ, 2002. – С. 87-92.
10. *Антонов А.С.* Параллельное программирование с использованием технологии OpenMP: Учебное пособие. – М.: Изд-во МГУ, 2009. – С. 54-56.
11. *Антонов А.С.* Параллельное программирование с использованием технологии MPI: Учебное пособие. – М.: Изд-во МГУ, 2004. – С. 66-73.
12. *Олифер В., Олифер Н.* Компьютерные сети. Принципы, технологии, протоколы. – 4-е издание. – СПб.: Изд-во Питер, 2010. – С. 310-321.
13. *Максимов Н.В., Попов И.И.* Компьютерные сети: Учебное пособие. – 4-е издание. – СПб.: Изд-во: Форум, 2010. – С. 237-239.
14. *Курейчик В.М., Писаренко В.И., Кравченко Ю.А.* Инновационные образовательные технологии в построении систем поддержки принятия групповых решений // Известия ЮФУ. Технические науки. – 2008. – № 4 (81). – С. 216-221.

Филиппенко Петр Николаевич

Технологический институт федерального государственного автономного образовательного учреждения высшего профессионального образования «Южный федеральный университет» в г. Таганроге.

E-mail: peterpain@mail.ru.

347939, г. Таганрог, ул. Чехова, д. 363, кв. 185.

Тел.: +79094167387.

Кафедра систем автоматизированного проектирования; аспирант.

Шашелов Артем Андреевич

E-mail: temant@mail.ru

347900, г. Таганрог, Октябрьская д. 11а, кв. 9.

Тел.: +79185185701.

Кафедра систем автоматизированного проектирования; аспирант.

Сеитова Светлана Владимировна
E-mail: bits@mail.ru
347942, г. Таганрог, 1-й Новый 16, кв. 84.
Тел.: +79064189974.
Кафедра высшей математики; аспирант.

Filippenko Petr Nikolaevich
Taganrog Institute of Technology – Federal State-Owned Autonomy Educational Establishment of Higher Vocational Education "Southern Federal University".
E-mail: peterpain@mail.ru.
363, ap., 185, Chehov Street, Taganrog, 347939, Russia.
Phone: +79094167387.
The Department of Computer Aided Design; Postgraduate Student.

Shashelov Artem Andreevich
11a, ap.9. Oktabrskaya Street, Taganrog, 347900, Russia.
E-mail: temant@mail.ru.
Phone: +79185185701.
The Department of Computer Aided Design; Postgraduate Student.

Seitova Svetlana Vladimirovna
E-mail: bits@mail.ru.
16, ap.84. 1-st. New street, Taganrog, 347942, Russia.
Phone: +79064189974.
The Department of Mathematics; Postgraduate Student.