СУПЕР КОМПЬЮТЕРЫ
зима-2013

Председатель редакционного совета
Владимир ВОЕВОДИН
Vladimir.voevodin@ supercomputers.ru

Над номером работали:
Выпускающий редактор
Игорь ЛЁВШИН
Igor.levshin@supercomputers.ru
Арт-директор
Виктория ИВАШКОВА
Victoria.ivashkova@supercomputers.ru
Корректор
Юлия ГОЛОМАЗОВА

Тексты:
Р. ВОДЕЙКО
В. ГОРБУНОВ
Е. ИВАШКО
А. КАШКОВСКИЙ
А. КОРЖ
О. КОРЖ
М. КРИВОВ
И. ЛЕВШИН
Д. МАКАГОН
Д. НИКИТЕНКО
А. ПОЗДНЯКОВ
П. СКОБЕЛЕВ
С. СОБОЛЕВ
А. СОКОЛОВ
И. ФЕДОТОВА
Ю. ЧЕРНЯВСКИЙ
А. ЧУЛЮНИН

Иллюстрации:
Владимир КАМАЕВ
Обложка: Олег ПАЩЕНКО

Учредитель
Даниэль ОРЛОВ

Издатель
ООО «Издательство СКР-Медиа»
SCR-MEDIA

Генеральный директор
Даниэль ОРЛОВ
Daniel.orlov@supercomputers.ru

Адрес редакции и Издателя:
117342, Москва, ул. Бутлерова, 17Б
www.supercomputers.ru

Издание «СУПЕРКОМПЬЮТЕРЫ» зарегистрировано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций (Роскомнадзор). Свидетельство о регистрации
СМИ ПИ №ФС77-38346 от 10.12.2009
Тираж 5000 экз.

Отпечатано
Типография ООО «Вива-Стар»
107023, Россия, Москва,
ул. Электрозаводская, д. 20, стр. 3
www.vivastar.ru

В номере

4 **О ползучих технических революциях**
а также контрреволюциях и мутациях в HPC

6 **О суперкомпьютерных списках как определяющих сознание**
HPCG пугает, но нам не страшно

8 **...зато ближе к реальным приложениям**
интервью с Джеком Донгаррой по поводу новых тестов HPCG

13 **Круглый стол в «Моряке»**
финальный аккорд конференции «Научный сервис в сети Интернет 2013»

18 **Камиль Ахметович Валиев**
Неблизкий путь к квантовым вычислениям

22 **Сложности перевода квантовых вычислений в классические**
что можно моделировать на «обычном» суперкомпьютере

26 **ПЛИС встраивается в облака**
необычные разработки «Кванта» для облачных сред

30 **О моделировании процесса обледенения линий электропередач**

34 **Моделирование спуска с орбиты**
расчеты для космоса ускоряют на GPU

38 **Использование высокоточных таймеров**
российские разработчики совершенствуют системы управления германскими ветряками

43 **От Motorola к анклавам**
интервью с Питом Бекманом, разработчиком ОС для экзаскейла

46 **Кристалл для «Ангары»**
НИЦЭВТ представляет свой высокоскоростной маршрутизатор ЕС8430

50 **HPC в Desktop Grid**
конференция в Петрозаводске по BOINC и добровольным вычислениям

54 **Суперкомпьютеры и мультиагентные технологии**
для решения сложных задач в реальном времени

60 **Рейтинг TOP50: куда крадемся?**

62 **TOP500**

64 **Суперкомпьютинг'13**
Конференция SC'2013 в Денвере

34

Моделирование спуска с орбиты на GPU

Новости

Климатические карты появятся на Amazon AWS

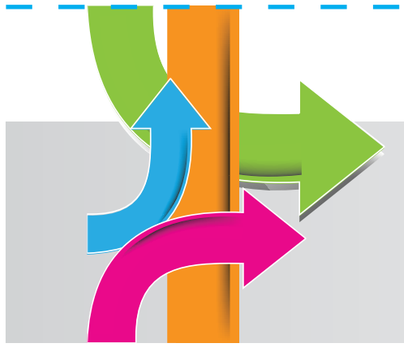
Amazon объявила о том, что NASA размещает для открытого доступа на их серверах спутниковые и метеорологические данные – температуру, осадки, площади лесных массивов. Но это сделано не только для того, чтобы желающие могли полистать картинки. Вместе с данными научному сообществу предлагаются инструменты анализа данных – часть исследовательской платформы NASA Earth Exchange (NEX), разработанной в суперкомпьютерном центре NASA.

Идеи и сотрудники Gnodal вливаются в Cray

Cray приобрела интеллектуальную собственность компании Gnodal Limited, известной на рынке Ethernet-коммутаторов высшего класса. Вместе с патентами и копирайтами Cray купила и команду из 15 ключевых сотрудников.
Gnodal – британская компания, которую основал разработчик интерконнекта Quadrics Фред Хоумвуд. Разработчики останутся в Европе. В Cray подчеркивают, что ни акции, ни бизнес-подразделения компании куплены не были. Новые сотрудники Cray будут заниматься разработкой новых сетевых технологий.

Автомат памяти

Одна из сенсаций конференции SC'2013 – представленный компанией Micron ускоритель для работы с памятью, который называется Automata. Известно, что это перепрограммируемый чип (видимо, что-то вроде нестандартного FPGA), задача которого – не просто ускорить доступ к памяти, но и выполнять операции, характерные для приложений будущего, работающих с обработкой изображений, средствами безопасности, биоинформатикой, а также гибкая работа с аналитикой неструктурированных больших данных. Архитектура Automata такова, что одновременно может выполняться много разнообразных операций. Automata легко масштабируется, к одному модулю памяти можно присоединить до 8 ускорителей.
В Micron говорят, что к реализации этого чипа они шли семь лет. Это не просто попытка штурмовать «стену памяти», расширить «бутылочное горло» доступа, а полностью новая концепция вычислений, где памяти отводится не только пассивная роль хранителя информации, но и роль участника вычислительного процесса. Уже готов SDK для программирования ускорителей, в работе даже новый язык – ANML (Automata Network Markup Language). В разработках принимало участие несколько университетов США.



Big Data обзаведутся своим кремнием

Калифорнийский стартап SSRLabs (Scalable Systems Research Labs) обнародовал планы своих разработок сопроцессоров, которые должны существенно увеличить скорости обмена данными и уменьшить энергоемкость систем, обрабатывающих большие данные. Будет разработано целое семейство сопроцессоров для рынка Big Data. Эти устройства специально создаются, чтобы упростить и ускорить анализ многомерных данных. Пока о результатах говорить рано, разработки еще на ранних стадиях.

Нейротранзисторы на новом витке

В Гарвардской школе инженерии и прикладных наук (Harvard School of Engineering and Applied Sciences – SEAS) разработали принципиально новый нейрочип. В нем используются экзотические материалы. Транзисторы выполнены в тончайшей пленке из соединения, в которое входят самарий и никель. Это позволило реализовать переменное сопротивление в «синапсах», то есть связи между «нейронами» могут усиливаться и ослаблять-ся, имитируя нервную клетку человеческого мозга. При этом экзотическое устройство может быть эффективно интегрировано в системы с обычной кремниевой электроникой. Еще одно интересное свойство: эти искусственные нейроны сохраняют свое состояние при полном отключении электропитания.

О ползучих технических революциях

Текст Игорь Лёвшин

В отличие от революций политических, о технических революциях обычно вспоминать всегда приятно, а иногда и участвовать в них или наблюдать их в реальном времени. Одно из приятных свойств технических революций – их нельзя проморгать, проворонить. Даже обывателю трудно их не заметить. Еще вчера по прешпекту неслась, позвякивая бубенчиками, конка, глядь – а вагончики уже на паровой тяге.

Еще недавно многочисленные шкафы новейшего компьютера были начинены хрупкими лампами. И вот уже та же и даже много большая мощь поместилась в ящиках с транзисторами и диодами. Возможно, некоторые из нас доживут до времен, когда квантовые компьютеры расправятся с кремниевыми, как некогда транзисторные разделались с ламповыми. Есть серьезные аргументы в пользу того, что традиционным (с момента смерти ламповых) кремниевым компьютерам такая судьба не угрожает. Да только у времени свои аргументы.

Но как разглядеть тихую, ползучую революцию, тем более в таких областях техники, как HPC, не слишком наглядной для широкой публики, да и для специалистов более широкого круга, чем разбросанные по суперкомпьютерным центрам всего мира горстки ученых-параллельщиков? Хорошо, пусть не техническую революцию, а верхушечный переворот или, наоборот, бунт низов.

HPCG плюс минус Linpack

Имеются некоторые признаки того, что грядет ползучая революция в HPC, подковерный переворот или еще какие-то с трудом заметные, но важные движения в отрасли.

Когда Джек Донгарра анонсировал еще один новый бенчмарк – HPCG, – никаких революционных лозунгов не звучало. Даже наоборот, Джек всячески успокаивал сообщество, что HPCG вовсе не замена привычному Linpack. Он – дополнение, еще один столбец в таблице. Казалось бы, ничего особенного не произошло. Но не все специалисты и аналитики остались равнодушными зрителями с «поживем – увидим» на устах. Тем более что заказчик нового теста – могущественное Министерство энергетики США, обладающее самой мощной сетью вычислительных центров в стране. Им не просто захотелось новых тестов. Было заявлено, что старые их не устраивают, так как не отражают производительности на реальных задачах отрасли.

Кстати, почему «революция»?

Скорее уж тогда своего рода контрреволюция. Дело в том, что по новым тестам, как ожидает по крайней мере один из разработчиков теста – Джек Донгарра (его соавтор, Майкл Эру, както спрятался в тени этой большой фигуры), гибридные системы, обязанные своим впечатляющим цифрам и призовым местам наличию ускорителей, заметно опустятся в таблице TOP500. А переход к гибридным вычислениям, казавшийся необратимым, вполне тянул на умеренно тихую революцию в суперкомпьютерном мире.

Тесты Linpack, понемногу модернизируясь, существуют уже два десятка лет. Достаточно долго, чтобы к ним привыкли. Списки TOP500, которые обнародуют два раза в год: в США на конференции SC и в Европе на ISC, – превратились уже в ритуал, который завораживает и экспертов, и просто интересующихся HPC. Их ждут, они являются гвоздем программы этих конференций.

Но и «усталость» накопилась в сообществе. В прошлом году произошел первый тихий бунт против Linpack – в NCSA (Национальном центре суперкомпьютерных приложений) при Университете Иллинойса отказались прогонять эти тесты на своем новом суперкомпьютере Blue Waters. Отдавая должное богатой истории Linpack, руководство центра предупредило, что результаты нередко дезориентируют сообщество и участвовать в этом они не хотели бы. Вице-директор проекта Blue Waters Уильям Крамер прямо сказал: «Место в списке TOP500 не имеет ничего общего с полезностью системы для науки. Список подталкивает организации к неправильному выбору. Нужны новые тесты или наборы из тестов». В результате в NCSA предпочли измерять производительность на наборе из популярных «тяжелых» реальных приложений.

За NCSA никто пока не последовал. Однако разговоры о дезориентирующем влиянии Linpack слышны в кулуарах все чаще. Ведь таблицы TOP500, основанные на результатах этого бенчмарка, – не просто развлечение, которого два раза в год ждут болельщики-суперкомпьютерщики. Журналисты сравнивают суперкомпьютеры с болидами Formula 1. На самом деле Formula 1 компьютерного мира – это и есть TOP500. Как и в автопроме, победы в соревновании прямо и косвенно влияют на траекторию развития бизнеса, а раз так, то и оправдывают затраты на создание машин-победителей. Но для пущего параллелизма TOP500 и Formula 1 надо бы добавить в последнюю сверхсовременный болид Great Wall Hover с миллиардами юаней в поддержку – от щедрот Центрального комитета коммунистической картии Китайской Народной Республики.

Еще один признак «контры» в новом бенчмарке: революции вообще и революционные процессы в HPC имеют типично «демократическую» направленность. Все последние тенденции свидетельствовали, что ширпотребные процессоры вытесняют остатки сложных устройств, знакомых уже более по историческим статьям: процессоры линии SPARC в японском K Computer были исключением, подтверждающим правило. Теперь же именно эта система имеет шанс оказаться на первой строчке TOP500 – если ранжировать по результатам HPCG (на момент написания статьи лишь предсказанным).

Кто скрывается под псевдонимом

Но появление новых тестов, значимое или нет, – это всегда Событие. Между тем под поверхностью HPC происходят Процессы. Что такое суперкомпьютер? Процессоры, сеть, операционная система. «Аптека, улица, фонарь» – как говаривал Александр Блок. Каждый участник этой тройки не застыл, меняется со временем, а тройка та же и такой останется в обозримом будущем. Но только при одном условии: если слово «процессор» будет означать процессор, «сеть» – сеть, а «ОС» – ОС.

Оставим в стороне современный процессор, который уже в некотором смысле больше похож на то, что раньше считалось многопроцессорной системой, оставим сеть и обратимся к ОС.

В суперкомпьютере есть и будет ОС, в этом никакого сомнения. Да не одна, а многие тысячи ОС! Но не «жирно» ли – тысячи ОС на одну систему? Наверное, нет, ведь от ОС, загружающихся на узлы, остались самые необходимые функции. Современные сверхлегкие ядра ОС – крохотные кусочки кода в сравнении с ОС прошлого, да и настоящего тоже. То ли еще будет лет через пять.

Пит Бекман возглавляет группу, которой поручено разработать ОС для будущего – эпохи экзаскейла. То есть ОС будет править компьютерный бал и в 2020-м? Да как сказать. В группе вообще не говорят об ОС – этот термин просто утерял большую часть своего смысла. Говорят об OSR, то есть ОС + среда выполнения (run-time). Именно OSR определяет всю внутреннюю жизнь будущих систем. Примерно так же видит будущее системное ПО Томас Стерлинг. Но не будешь же писать в грантах и отчетах «OSR будущего» – не поймут-с.

Революционные облака

Концепция облаков, конечно, революционна. Особенно в прошлом – в те времена, когда о ее предыдущей инкарнации – ASP, провайдер сервисов приложений – говорили на каждом углу Кремниевой долины, то есть в конце 1990-х, перед грандиозным крахом «доткомов». Тогда никому, кажется, и в голову не приходило говорить о суперкомпьютерных ASP. Облака с тех пор стали реальностью даже в России, где всегда боялись отдавать свои данные таинственному облачному хозяину. Что до суперкомпьютерных облачных сервисов, то облака Amazon E2, так сказать, набирают вес.

Во-первых, появляется все больше средств работы с AWS – Amazon Web Services. Решивший воспользоваться услугами остается уже не один на один с арендованными узлами, а с огромным количеством инструментов для работы, базами данных, поисковыми приложениями. AWS обрастает интерфейсами, его мощности перекупают третьи фирмы, чтобы на их базе строить свои облака с другими возможностями, а их, в свою очередь, сдавать арендаторам.

Ползучая облачная революция связана, возможно, и с приходом в облака Больших Данных. На днях Amazon объявила о том, что NASA размещает для открытого доступа на их серверах спутниковые и метеорологические данные – температуру, осадки, площади лесных массивов. Но не просто для того, чтобы желающие могли полистать картинки. Вместе с данными научному сообществу предлагаются инструменты анализа данных, часть исследовательской платформы NASA Earth Exchange (NEX), разработанной в суперкомпьютерном центре NASA.

Но это для людей науки. А вот другое недавнее предложение Amazon – Kinesis. Так называется приложение, которое работает поверх облаков Amazon E2. Оно создано для анализа потока данных в реальном времени, таких как логи, финансовые транзакции или информация о кликах на Web-портале, статусы в социальных сетях. При этом данные могут поступать со скоростью до 100 ТБ в час. После того как данные обработаны, их можно уничтожить или перенаправить в другой поток.

Другими словами, облака, суперкомпьютеры и Big Data нашли друг друга. А такой союз может многое перевернуть в индустрии. Одно дело – ученые, другое – коммерческие компании, анализирующие нескончаемые потоки своих данных. Их инвестиции могут переформатировать отношения и внутри устоявшегося уютного HPC-сообщества. Конечно, эти предреволюционные зарисовки мало на что претендуют. Разве что на привлечение внимания к не самым громким событиям и не самым обсуждаемым настроениям. На самом деле и большие, «настоящие» технические революции всегда несколько размазаны и расплывчаты. Почему возможны споры о том, кто изобрел лампочку или радио? Разумеется, потому, что этот вопрос политизирован. Но основания для заявлений типа «Страна Х есть родина слонов» дает история развития техники. Старт революции может отсчитываться не от принципиального открытия, а от, может быть, незначительной с научной точки зрения «инновации», которая оказывается точкой кристаллизации или комом, с которого начинается лавина.

О суперкомпьютерных списках как определяющих сознание

Текст Антон Корж

Суперкомпьютерные списки в настоящее время привлекают все большее внимание широкой общественности, и это вполне объяснимо, ибо с их помощью даже неспециалисты могут сравнивать различные суперкомпьютеры как два действительных числа. Даже школьного образования достаточно, чтобы анализировать колонки чисел, подводить статистику и выполнять прочие наукоподобные манипуляции. Самый известный список в настоящее время – TOP500, его знают и чиновники, и журналисты, чувствующие себя причастными к суперкомпьютерной отрасли. Есть и другие, менее известные списки, такие как Green500 и Graph500. Если первый является лишь взглядом в другом ракурсе на все тот же TOP500, то Graph500 представляет собой совершенно иной подход.
В феврале 2013 года суперкомпьютерный центр Сан-Диего в Калифорнии объявил о планах создания нового списка Big Data TOP100, впрочем, пока неясна даже дата выхода первой редакции этого списка. Однако странным образом внимание общественности, формируемое представителями СМИ, в настоящее время приковано к объявленному в июне Джеком Донгаррой, одним из создателей TOP500, новому подходу к тестированию суперкомпьютеров на основе теста сопряженных градиентов HPCG.
С легкой руки журналистов уже говорят о новом списке TOP500 и проводимым Департаментом энергетики США свержении Linpack с трона царства бенчмарков. Однако профессионалы испытают дежа вю: что-то похожее уже было ранее, при создании (с участием того же Донгарры) пакета бенчмарков HPC Challenge. И никакой революции тогда не произошло. Но давайте рассмотрим все по порядку.

TOP500, Linpack. Linpack – самый известный бенчмарк в мире HPC. (К слову, в мире персоналок он практически не используется.) Но на самом деле Linpack является библиотекой численных методов линейной алгебры, разработанной в 1970-х годах и в настоящее время уже устаревшей. Тогда к библиотеке прилагался тест, решавший систему с 1000 неизвестных, и в 1970-е годы это было актуальной задачей для физических расчетов и отражало уровень самых быстрых компьютеров того времени. Впоследствии, как мы знаем, производительность вычислительных систем росла экспоненциально, а в последние десятилетия все суперкомпьютеры стали массивно-параллельными.
В ответ на эти тенденции тест был переписан под параллельные архитектуры и стал называться HPL (High Performance Linpack). Кроме того, выяснилось, что для достижения хорошей производительности на параллельных машинах нужно увеличивать размер задачи. А поскольку разные машины имели разрыв в производительности более трех порядков, то подобрать единый размер задачи было невозможно. Тогда и пришли к соломонову решению – каждый выбирает размер задачи под свою систему, что и позволяет достигать текущим системам высоких уровней эффективности. Вот только получилось, что скорость решения сравнивается системами разных задач. Другим отрицательным эффектом стал рост размера задач вплоть до систем размером в 12 миллионов неизвестных. В реальной жизни такие системы всегда являются разреженными и решаются совсем другими методами. Таким образом, Linpack и TOP500 привели к двум результатам – популяризации суперкомпьютерного мира среди неспециалистов и к вырождению смысла оценки на данном тесте. С точки зрения прикладного пользователя, результат на Linpack не гарантирует высокой производительности на реальной задаче. Но увеличенное внимание чиновников и журналистов к TOP500 заставило производителей суперкомпьютеров уделить повышенное внимание к оптимизации железа под данный тест. Конечно, квалифицированным пользователям суперкомпьютеров это проблем не создавало – в тендеры всех приличных суперкомпьютерных центров всегда вписывалась не столько производительность на Linpack, сколько производительность на наборе тестов, отражающих большую часть реальных задач данного центра. Тем не менее увлечение Linpack и TOP500 привело к некоторому крену суперкомпьютерной отрасли в сторону грубого и чисто маркетингового количества флопс. Например, увеличение числа флопс на такт до 8 в центральных процессорах и даже до 16 в ускорителе (впрочем, на многих реальных задачах оказалось, что более оправдано название «замедлитель») Xeon PHI было, на мой взгляд, вызвано гонкой за Linpack-флопсами и TOP500. На реальных задачах это увеличение крайне редко приводило к росту производительности.
Что касается Green500, этот список был удачной попыткой переключить внимание на энергоэффективность суперкомпьютеров, что в последние годы стало актуальным, так как ведущие машины подошли вплотную к экономически разумным пределам энергопотребления суперкомпьютера в десяток мегаватт. Но при этом оценка флопс на ватт опять же ориентирована на Linpack, который не интересен ни одному реальному пользователю суперкомпьютеров.

Бенчмарк HPCG. Попытка создания новой метрики, конечно, является актуальной.
По меньшей мере этот бенчмарк отражает производительность суперкомпьютеров при решении систем линейных уравнений с разреженными матрицами методом сопряженных градиентов. Такие задачи реально возникают у части пользователей, и с этой точки зрения данный бенчмарк более интересен, чем Linpack. Приведет ли введение HPCG к пересмотру сил в суперкомпьютерной отрасли, пока неясно. Судя по тому, что он поддерживается департаментом энергетики США, – есть шанс. Но если вспомнить, то ранее уже была подобная ситуация, когда из-за недовольства применения Linpack тот же департамент настоял на создании пакета тестов HPC Challenge, к которому также приложил руку Джек Донгарра. Но тот пакет, довольно-таки сбалансированный по составу тестов, не снискал большой популярности, и интерес к нему постепенно сошел на нет. Скорее всего, причиной была сложность интерпретации результатов – множество цифр производительности на разных тестах пакета, отсутствие единого рейтинга, способного привлечь внимание широкого круга неспециалистов. С учетом этой ошибки вторая попытка замены Linpack и была сделана с объявлением в июне теста HPCG. Замечу лишь, что остается вторая проблема – соотнесение с пиковой производительностью окажется гораздо менее осмысленным, чем в случае Linpack. А именно сравнение производительности на Linpack с пиковой производительностью сыграло важную роль при становлении нынешнего TOP500.
Не могу пройти мимо и не прокомментировать расхожее мнение, что на этом тесте окажутся в проигрыше системы с графическими ускорителями, мол, они вообще прироста производительности показать не смогут, так что их даже использовать при запусках не будут. Да, безусловно, сопроцессорам (GPU и Xeon PHI) придется трудно на этом тесте. Но я убежден, что сопроцессоры будут более эффективны на этом тесте, чем центральные процессоры, причем не столько за счет большей пиковой пропускной способности, сколько за счет гораздо большего уровня параллелизма. И лишь камнем на шее будет висеть необходимость коммуникаций, проходящая через узкое горло системной шины и центральный процессор. В связи с чем, по моему мнению, сопроцессоры заиграют лишь тогда, когда в них наконец-то будет интегрирован сетевой контроллер и большие объемы памяти.
Список Graph500 дает другой пример создания бенчмарка с целью ранжирования суперкомпьютеров и создания соревнования по достижению наибольшей скорости на этом бенчмарке. Graph500 довольно молод, датирует первую редакцию списка 2009 годом и в настоящее время содержит менее 200 систем. Основной задачей данного списка в 2009 году было введение принципиально альтернативной Linpack-метрики, не вместо Linpack, а как дополнение. Если Linpack оценивает производительность во флопсах, то многие задачи не имеют вычислений с плавающей точкой вообще. И именно эту территорию (куда попадает ныне модное Big Data) занимает Graph500. Обработка огромного графа, с одной стороны, является разумной задачей, непростой для заточенных на Linpack архитектур, да и моделей программирования. С другой стороны, задача поиска вширь довольная проста и красива с точки зрения ранжирования систем. Имеются единицы измерения «тепсы», или число пройденных ребер графа, дополняющие флопсы Linpack. В некотором смысле это реинкарнация «гупсов» из базового теста Random Access, входившего в пакет HPC Challenge. Это и позволило Graph500 в такой короткий срок занять третье место среди известных списков суперкомпьютеров. И, создав здоровую конкуренцию по скорости обработки графов, это уже привело к существенному прогрессу в области параллельных алгоритмов обработки больших графов. Но Graph500 как молодой представитель проявляет и наибольшую гибкость среди списков. Не успев повзрослеть и даже набрать 500 позиций, список уже собирается меняться за счет добавления второго ядра – поиска кратчайшего пути в графе. Однако вопрос, как учитывать при ранжировании результаты двух разных ядер, пока еще не получил единого ответа при обсуждении внутри руководящего комитета Graph500. Было решено вводить его опционально и параллельно первому ядру, а по мере накопления статистики придет и ясность с тем, как распределять места.
Кроме того, в Лейпциге был анонсирован список GreenGraph500, призванный оценить энергоэффективность при решении задач поиска в больших графах.
Резюмируя, хочется сказать, что важная задача создания суперкомпьютерного списка не так проста, как кажется. Нужно не только придумать хороший бенчмарк и метрику к нему, но и соблюсти компромисс между интересами как пользователей, так и производителей суперкомпьютеров. И лишь тогда будет понятно, в какую сторону экзафлопса (а может, петатепса) будет двигаться отрасль.

...зато ближе к реальным приложениям

Интервью с Джеком Донгаррой по поводу новых тестов HPCG (Джек Донгарра, Майкл Эру)

Игорь Лёвшин: HPCG расшифровывается как High Performance Conjugate Gradient. Что представляет собой Conjugate Gradient – метод сопряженных градиентов?

Джек Донгарра: Это известный метод, он широко используется. Когда имеют дело с итерационными вычислениями над сильно разреженными матрицами, используют предобусловливание, которое как бы подталкивает решение в некотором направлении.

И. Л.: Как возникла необходимость в новых тестах? Почему привычный бенчмарк Linpack недостаточно хорош?

Д. Д.: Появление Linpack – это скорее плод случайных обстоятельств, чем какой-то распланированной деятельности. Людям было просто любопытно, как можно оценить производительность их машин. Linpack поначалу жил своей собственной жизнью. И я вам скажу, что тогда, в 1970-х, когда этот бенчмарк был разработан, он давал довольно адекватное представление о производительности суперкомпьютера. Но с тех пор многое изменилось, и прежде всего приложения, которые на них считаются. Сейчас Linpack уже не слишком хорошо подходит для определения производительности на типичных задачах.

Но проблема не только в самих тестах, а в том, что люди воспринимают их очень серьезно, как важный критерий для принятия решений. Вплоть до того, что люди конструируют компьютер специально так, чтобы он показал хороший результат на Linpack и занял хорошее место в списках. Но ведь компьютеры создают не для тестов, а для решения реальных задач. Именно это было главной заботой Министерства энергетики США, когда они поставили задачу разработки новых тестов: стимулировать создание машин, быстрых на большинстве нужных им приложений, а не на Linpack. Мы собрались вместе и стали смотреть, что можно сделать.

Типичные современные задачи связаны с решением больших систем дифференциальных уравнений в частных производных. При решении используется дискретизация, итеративные методы, а что касается структуры данных, то для таких уравнений типичны сильно разреженные матрицы. Предобусловленный метод сопряженных градиентов, который мы выбрали, действительно хорошо отражает производительность на больших разряженных матрицах.

Некоторые особенности Linpack мы оставили – например, немалое время работы тестов. Компьютер должен получить реальную нагрузку, она должна стать настоящим испытанием для новой системы. Мы считаем, что тесты должны проходить как минимум 6 часов. Величина задачи коррелирует в новых тестах с количеством процессоров: чем их больше, тем больше система дифференциальных уравнений. По результатам новые тесты будут сильно отличаться от Linpack.

Linpack позволяет довольно близко приблизиться к пиковой производительности. Доходит до 90%, а машин, работающих на 70–80% или 65–70% от пиковой, очень много.

На новых тестах 10% может оказаться блестящим результатом. Зато ближе к реальным приложениям. Этот тест испытывает не только возможности вычислений с плавающей точкой, но и коммуникативные возможности, обеспечивающие эти вычисления. Результаты по Linpack и по HPCG часто будут сильно отличаться, и это различие в основном и будет появляться из-за разных коммуникационных возможностей, которые Linpack не особенно учитывал.

Мы обнародуем новый бенчмарк на конференции SC2013 в Денвере, там у участников будет возможность все увидеть и получить код для тестирования. Linpack развивался десятилетиями, и новым тестам придется тоже пройти немалый путь, прежде чем стать признанными и популярными, но я надеюсь, что новая колонка – HPCG – в таблице TOP500 будет постепенно заполняться, так что пользователи смогут упорядочивать данные сразу по двум разным спискам, сравнивать и делать выводы. Это удобно, но это не главная наша цель. Мы надеемся, что производители компьютеров учтут результаты HPCG при проектировании новых систем.

И. Л.: Тест разрабатывался по инициативе Министерства энергетики. Значит ли это, что он близок задачам, типичным для этого ведомства?

Д. Д.: Министерство энергетики – один из главных потребителей суперкомпьютерных ресурсов. И их, конечно, очень волнует, будут ли будущие системы экзапроизводительности чемпионами Linpack или же они будут эффективно решать задачи министерства. Для этих задач метод предобусловленных градиентов довольно типичен. Но не только для них.

И. Л.: А для кого еще?

Д. Д.: Климат и предсказание погоды. Сгорание. Ядерные реакции. Материаловедение. Там повсюду можно увидеть системы дифференциальных уравнений в частных производных, там используется дискретизация, там большие системы уравнений с разреженными матрицами.

И. Л.: А какие задачи и методы были типичны для приложений, чью эффективность неплохо отображал Linpack?

Д. Д.: Это похожие задачи, в тех же примерно областях, что я только что перечислил, но там речь идет о плотных матрицах, заполненных ненулевыми элементами. Сейчас разреженные матрицы используются намного чаще, поэтому Linpack искажал картину. Но и плотные матрицы по-прежнему используются – мы ведь говорим не об отмене Linpack, а о дополнении к нему.

И. Л.: Кроме TOP500, основанного на Linpack, имеется немало других списков. Например, Graph500, который совершенно не похож на Linpack.

Д. Д.: Да, списков много, есть и Green500, но это просто TOP500, пересчитанный на потребляемую энергию. Graph500 действительно совершенно другой. Он оценивает не производительность с плавающей точкой, а эффективность целочисленных операций, умение обращаться со структурами данных, умение следовать цепочкам данных, идти по ребрам графа, эффективность поиска по графу. Но главное, что речь идет только о целых числах, а это принципиальная разница. Если в приложении много целочисленных операций, если там активно используются указатели, то данные Graph500 неплохо отразят производительность на таких задачах. А вообще, лучше всего прогонять самые разнообразные тесты, если есть такая возможность. В любом случае судить о системе на основании только одного числа – это совсем плохо.

И. Л.: Я вспомнил о нем потому, что там коммуникационная часть чрезвычайно важна. Важна эффективность обмена короткими сообщениями.

Д. Д.: Да, там необходим доступ к данным, расположенным случайным образом, а для этого нужны быстрые интерконнекты.

И. Л.: Но можно ли соответственно ожидать, что, несмотря на принципиальную разницу между вычислениями с плавающей точкой и целочисленными, система, показавшая хорошие результаты и на Linpack, и на Graph500, скорее всего, покажет недурную эффективность на HPCG?

Д. Д.: Нет, вовсе не обязательно. Корреляции Linpack, HPCG и Graph500 неочевидны. Даже если они есть, не думаю, что они сильны. Выводы и предсказания, как поведет себя система на одном тесте на основании результатов других, сомнительны. Конечно, если мы берем, скажем, Linpack и тестовое перемножение матриц, корреляции будут сильными. Именно поэтому, если у вас есть результаты Linpack, то прогонять перемножение матриц не имеет особого смысла. Но Linpack, HPCG и Graph500 – другое дело.

И. Л.: Если мы посмотрим на список Graph500, то увидим, что он совершенно не похож на TOP500. А насколько сильно HPCG перетряхнет список TOP500?

Д. Д.: Уверен, что первые строчки не совпадут. Если бы меня попросили погадать, какая машина будет на первой строчке HPCG, я бы поставил на японский K Computer потому, что должна победить машина с хорошим балансом скорости вычислений с плавающей точкой и быстрых коммуникаций. Это вообще прекрасная машина, сконструированная специально для научных вычислений – и процессоры выбирались из этих соображений, и сеть. А вот китайский суперкомпьютер, нынешний лидер, построенный на ширпотребных процессорах Intel и их же ускорителях Xeon Phi, сильно опустится в списке. Потому что графические ускорители вообще не слишком хорошо вписываются в HPCG. Я думаю, что для того, чтобы выжать хорошую эффективность HPCG из ускорителей, придется приложить очень большие усилия. Возможно, машины с ускорителями окажутся даже менее быстрыми, чем без них. Думаю, что неплохие результаты

покажут IBM BlueGene. Они, как и K Computer, разрабатывались для научных вычислений, у них тоже продуманные и быстрые коммуникации. Очень хорошо сбалансированные машины. Так что, возможно, список будет выглядеть так: вверху K Computer, дальше несколько BluGene. Системы с ускорителями заметно опустятся. Но… увидим.

И. Л.: Из десятки TOP500 хоть кто-то останется в новой?

Д. Д.: Не берусь предсказать. Напомню, что тестам Linpack уже 20 лет, у них история и авторитет. Новые тесты не сразу приобретут популярность, их не сразу возьмут для тестирования. Первое время люди наверняка будут смотреть на оба числа. Кстати, Graph500 – это только название. В нем пара сотен систем, а не 500. Пока наберется 500, пройдет время.

И. Л.: Но есть разница. За HPCG стоит Министерство энергетики, правительство, может быть, и ваш личный авторитет. То есть существуют все предпосылки для того, чтобы тесты приобрели популярность довольно быстро.

Д. Д.: Может быть. Но не только это. Бенчмарков много, но популярными становятся те, что понятны сами по себе, где понятно, как они соотносятся с производительностью определенных приложений, как взаимодействуют с архитектурой суперкомпьютера и, наконец, насколько легко их запустить и настроить. Мы старались сделать их понятными и легкими в использовании.

И. Л.: Сейчас все говорят об эпохе экзаскейла. Но если, допустим, через пять лет ваши новые тесты станут популярными, если их предпочтут другим, то слово «экзаскейл» приобретет несколько другое значение, это будет другой экзаскейл.

Д. Д.: Да, конечно другой. Но и сейчас порой случается путаница. Когда говорят, что экзаскейл будет достигнут в таком-то году, имеют в виду пиковую производительность. Экзафлопсные машины, возможно, покажут даже менее 100 Пфлопс на HPCG. Между тем китайская система Tianhe-2 уже сейчас показывает более 30 Пфлопс на Linpack, а пиковая у нее вообще около 55 Пфлопс.

И. Л.: Для многих стран место в списке TOP500 имеет политическое значение. И для США, наверное, а уж для Японии наверняка, а что до Китая, так тут уж никаких сомнений. Не будет ли разговоров, что вот де американцы нарочно придумали новые тесты, чтобы убрать с первой строчки своего самого грозного конкурента – Китай?

Д. Д.: *(Смеется)* Не знаю, не знаю. Конечно, быть на первой строчке престижно. Но ведь суперкомпьютер – это просто инструмент. Надо еще иметь людей, которые этим инструментом умеют пользоваться. И хорошо было бы, если бы эти люди не просто пользовались, а делали открытия, реализовывали свои идеи, лучше понимали законы физики. Это мощные и дорогие инструменты научного исследования, но ведь есть еще ускорители в ЦЕРНе, есть телескоп Hubble – тоже сложные и дорогие. Но их используют для проверки важных идей и поиска новых явлений. То же должно происходить и с суперкомпьютерами. Важно лишь как их используют, только это может оправдывать их существование.

И. Л.: Разрабатывая тесты, Вы наверняка общались с представителями основных производителей компьютеров. Какова их реакция? Я к тому, что новые тесты затронут интересы многих компаний. Наверняка NVIDIA, возможно, Intel и AMD.

Д. Д.: О да, у Intel очень серьезные намерения по части ускорителей. Но в любом случае ускорители останутся полезными для определенного рода задач. А вот окажутся ли они полезными при решении большого набора разно-образных задач – это вопрос. Но это же интересно – получится важное исследование возможностей и ограничений ускорителей.

И. Л.: А системы на базе процессоров ARM будут себя уютно чувствовать на новых тестах?

Д. Д.: Вообще, я бы не стал выделять ARM в отдельную категорию. Скорее так: сейчас в HPC доминируют три архитектуры. Первая группа – системы на традиционных ширпотребных процессорах. Вторая – то же плюс ускорители. Третья – системы с легкими ядрами. В BlueGene тоже легкие ядра, они выполняют простые операции. Как и ARM. Конечно, к ARM сейчас большой интерес, возможно, это одна из точек роста. Люди следят за интереснейшим проектом с ARM в Барселоне. Потенциал есть у всех трех, а которая из них одержит верх – сказать сейчас трудно.

И. Л.: Среди этих трех закономерно нет архитектуры систем на сложных, неширпотребных процессорах. Но если K Computer вновь займет первую строчку, не подтолкнет ли это к развитию и этого направления?

Д. Д.: Да, вполне может быть. Системы на ширпотребных про-

Максим Кривов

Переход от бенчмарка HPL (High Performance Linpack) – в суперкомпьютерном сообществе уже стало традицией называть его просто Linpack – заключается в сохранении предметной области, но смене используемого метода. Так, если в HPL инвертируется просто огромная матрица, то в его потенциальном приемнике HPCG решается также огромная, но уже СЛАУ (система линейных алгебраических уравнений). Это является проблемой более частной, но большинство расчетных задач именно к ней и сводится. Другое существенное отличие – переход от плотных матриц к разреженным, что, опять-таки, более точно соответствует тем расчетам, для проведения которых суперкомпьютеры и строятся. Осталось лишь «запихать» в этот бенчмарк всевозможные схемы работы с данными – и вот она, почти идеальная метрика!

Последнее достигается за счет выбора алгоритма решения той самой СЛАУ. В HPCG было предложено использовать популярный метод сопряженных градиентов, суть которого проста: давайте-ка исходную систему запишем как какой-нибудь функционал, минимум которого будет соответствовать ее точному решению. И таким образом будем просто искать этот самый минимум, используя универсальный метод оптимизации. Согласно которому сначала строится приближенное решение (например, нулевой вектор), а дальше делается серия шагов по направлению, определяемому по градиенту функционала на каждом предыдущем шаге. И таким образом будем потихоньку двигаться к искомому решению СЛАУ в лице минимума этого самого функционала.

Проблемой данного алгоритма является именно «потихоньку». Если матрица в СЛАУ имеет слишком большое число обусловленности, то процесс будет очень долгим, а может, даже и расходящимся. Чтобы этого избежать, исходная СЛАУ немного видоизменяется путём умножения обеих частей на матрицу-предобусловливатель, которую легко инвертировать. Таким образом, с одной стороны, решаться будет «хорошая» СЛАУ, а с другой, по полученному решению легко восстановить искомое. Построение же матрицы-предобусловливателя – это отдельная и немного грустная история, но в случае HPCG о ней можно забыть. Там используется простой предобусловливатель Гаусса-Зейделя, который является нижней половинкой матрицы исходной СЛАУ.

Итак, что же все это дает? А то, что теперь требуется выполнять операции, в которых совершенно разные схемы работы с данными. И массивы складывать, и их свертку проводить, и треугольные СЛАУ решать, и даже вектор на матрицу умножать. Поэтому появляется множество «узких мест», которые позволяют комплексно оценить весь суперкомпьютер, а именно его вычислительную мощь, скорость межсетевых соединений и объем памяти. А что еще более важно, именно эти «узкие места» и наблюдаются в прикладных задачах, что позволяет избежать озвученной выше проблемы HPL — получаемые на тесте HPCG баллы могут использоваться не только для определения места в TOP500, но и в качестве характеристики скорости решения реальных задач.

цессорах относительно дешевы, с ними трудно конкурировать, но кто знает?

И. Л.: Кстати, о Японии. Ведь это страна, где до сих пор в компьютерной индустрии живы и развиваются RISC-процессоры и системы на их базе, в то время как во всем мире они фактически ушли в прошлое. Возможно ли возрождение сложных процессорных архитектур?

Д. Д.: В США довольно много занимаются разработкой новых процессорных архитектур, но, скорее, на академическом уровне. Это исследовательские проекты. Смотрят, какие преимущества можно бы извлечь из новых архитектур, какие проблемы появляются.

До воплощения в кремнии такие исследования обычно не доходят: дорого. Но я надеюсь, что новые идеи в конце концов воплотятся в процессоры, а те станут базой для суперкомпьютеров нового поколения.

Но пробиться в мейнстрим сложным, специализированным процессорам, конечно, труднее.

И. Л.: Мы не касались пока важнейшей темы – программного обеспечения. Вы говорили, что производительность HPCG может оказаться меньше 10% от пиковой. Но, может быть, если переписать код приложений и системное ПО, то дойдет и до 20%?

Д. Д.: Да, правильно. ПО – это, конечно, важнейший вопрос, и оно может многое изменить. Но кроме ПО есть и еще один фактор, который надо учитывать: рабочее время, необходимое на разработку нового ПО и его разворачивание на системах. Это самый дорогой из ресурсов. Насколько программа должна работать быстрее, чтобы оправдать время на ее переписывание и настройку? Боюсь, что часто лучшей машиной оказывается та, что проще, а не немного быстрей. Суперкомпьютеры время от времени сравнивают с болидами. Но выигрывают не болиды, а пилоты, которые ими управляют.

От людей, работающих на суперкомпьютере, от их квалификации зависит многое. И очень много зависит от того, насколько им легко и комфортно работать с прикладными и системными программами.

И. Л.: Смогут ли изменить ситуацию с ПО такие проекты, как разработка принципиально новой операционной системы для машин эпохи экзаскейла? Я говорю о проекте, который ведет Пит Бекман.

Д. Д.: Это очень интересный проект, необходимый для развития индустрии.

Но только не надо забывать, что проект этот – исследовательский, что речь идет о суперкомпьютерах будущего. Там не ставится задача получить софт, дающий преимущества в уже существующих суперкомпьютерах.

И. Л.: Как мне кажется, в мире существует бесчисленное множество вычислительных центров, где подход примерно одинаков: заполучить пару машин с высоким местом в списке TOP500, то есть по результатам Linpack, и к ним еще парочку каких-нибудь специфических. Но есть буквально считанные центры, которые идут другим путем, явно Linpack-ориентированным. Про центр в Барселоне мы уже говорили. Есть и очень своеобразный центр в Сан-Диего, не так ли?

Д. Д.: Да, есть суперкомпьютерный центр в Сан-Диего, который, кстати, недавно выиграл грант NSF – Национального Научного Фонда. В этом центре особенно много внимания достается данным – их анализу, доступу к ним. HPC-центры сами по себе генерируют много данных, с которыми надо уметь эффективно обращаться. Так что HPC и Big Data в любом случае идут рука об руку.

В Сан-Диего хотят выйти на новый уровень понимания того, как работать с большими данными. Это очень интересно и полезно для индустрии.

И. Л.: Ждет ли в будущем большие данные и HPC конвергенция?

Д. Д.: Она всегда в какой-то форме была. В научном сообществе всегда понимали, что большие данные и высокопроизводительные вычисления – это близкие области.

Но сейчас люди, которые никогда не были внутри HPC-сообщества, поняли, что для бизнеса анализ огромных массивов данных может принести большую пользу, и окунулись в эти области. Это новая группа в сообществе.

И. Л.: Допустим, некоторая корпорация решила анализировать свои данные. Им нужен суперкомпьютер. На какие цифры им обращать внимание прежде всего – на Linpack или на HPCG? Или на то и другое? А может, следует прогонять какие-то другие тесты?

Д. Д.: А это зависит от того, как они собираются анализировать свои данные. Если они собираются пользоваться методом главных компонент, уменьшать размерность данных, выделять самое интересное, то у этих методов много общего с теми, что лежат в основе HPCG, так что новые тесты как раз хорошо отразят возможности их компьютера.

Новые тесты не сразу приобретут популярность, их не сразу возьмут для тестирования. Первое время люди наверняка будут смотреть на оба числа. Кстати, Graph500 – это только название. В нем пара сотен систем, а не 500. Пока наберется 500, пройдет время.

Круглый стол в «Моряке»

Участники круглого стола:

- Воеводин Владимир Валентинович, заместитель директора НИВЦ МГУ
- Аксёнов Андрей Александрович, технический директор компании «ТЕСИС»
- Болдырев Юрий Яковлевич, профессор СПГПУ
- Бухановский Александр Валерьевич, директор НИИ наукоемких компьютерных технологий СПбГУ ИТМО
- Гергель Виктор Павлович, декан факультета ВМК ННГУ
- Корнеев Виктор Владимирович, заместитель директора НИИ «Квант»
- Якобовский Михаил Владимирович, заместитель директора Института прикладной математики имени Келдыша

Финальным аккородом конференции «Научный сервис в сети Интернет 2013», по традиции проводимой в доме отдыха «Моряк» неподалеку от пос. Дюрсо, была дискуссия о будущем HPC. Мы публикуем фрагменты этой беседы.

В. В. Воеводин: За последнее десятилетие все, что касается архитектуры, ушло далеко вперед по сравнению с тем, что мы имели еще в конце 1990-х годов. Если мы здесь будем говорить о «суперкомпьютере-2020», то не стоит ли сейчас остановить развитие вычислительной техники и колоссальные высвободившиеся ресурсы направить на развитие методов, алгоритмов, программного обеспечения? Мы уже сейчас не можем использовать все преимущества, которые имеют компьютеры из первой сотни TOP500.

В. В. Корнеев: Останавливаться ни в коем случае нельзя. В ПО (особенно в системном) есть множество мест, которые были сделаны абы как, без учета производительности. Это нерешенные проблемы и с динамической памятью, и с контролем границ и т. д. Поэтому при росте производительности компьютеров необходимо переходить на более надежное и эффективное программное обеспечение. И конечно, мы все ожидаем появления перспективно иных технологий, таких как квантовые компьютеры. Квантовый компьютер позволит решать традиционные задачи существенно быстрее, а переход к более эффективным, малоразмерным элементам, выполняющим новые функции, неизбежен.

М. В. Якобовский: Достаточно вспомнить появление транспьютеров. Первые транспьютеры использовать было не очень просто. Но то, что оказалось написано для транспьютеров, получило свое развитие, благополучно используется сейчас и будет использоваться завтра. А вообще, есть надежда, что через некоторое время нам предложат аппаратуру, работать на которой будет проще. Процесс развития суперкомпьютерной техники идет с двух сторон. С одной стороны, предлагается аппаратура, которую может де-

лать промышленность (многоядерные процессоры для многих стали неожиданностью), с другой стороны, производители аппаратуры реагируют на запросы разработчиков программного обеспечения – например, запрос о приближении вычислений к оперативной памяти явно прозвучал со стороны научного сообщества и это вылилось в использование графических процессоров для вычислений.

А. В. Бухановский: Жизненный цикл технологии – 3–6 лет, а жизненный цикл вычислительной модели – 10–20 лет. Модели и технологии связаны между собой. Еще в 2005 году американскому президенту был представлен отчет группы экспертов, в которой была отмечена эта взаимосвязь. Что будет, если мы остановимся? Технологии развиваются быстрее, нежели развиваются сами модели. Если, скажем, технологии не будут развиваться в течение пяти лет, нам потом понадобится десять лет на то, чтобы наверстать отставание.

В. В. Воеводин: Кто правит бал в индустрии – IT-компании, создающие технологии, или научное сообщество декларирует свои задачи и производитель подстраивается?

В. В. Корнеев: Вычислительная техника – прекрасная иллюстрация истины, что искать нужно во всех местах, но найти можно только под фонарем. Инженеры ищут возможность выполнять некоторые преобразования данных, а задача программистов – изложить проблему в терминах, наиболее эффективных для найденных преобразований. Например, было понятно, что на выполнение и дешифровку каждой команды тратится очень много времени. Тогда придумали векторные операции, сделали CRAY1, имевший 160 млн операций в секунду. И общественность долго ломала голову, как его использовать. Но современный этап характеризуется тем, что теперь векторизованных решений задач существует великое множество, для всех проблемных областей. И это общая закономерность. Вычислительная машина – совокупность приемов. Тот, кто умеет эти приемы эффективно использовать, тот и создает эффективные программы.

Инженеры создают то, что возможно на данном этапе развития технологий, а программисты в этой области возможного должны сформулировать проблему, для которой требуется решение.

В. В. Воеводин: Таким образом, «мы все обречены»? Что попадется под фонарем, с тем нам и жить?

В. В. Корнеев: По сути да. Приведу в пример все тот же квантовый компьютер на кубитах. Пока существует весьма ограниченное количество задач, которые можно решать на подобной технике, типа разложения на множители и т. п. А есть другой тип квантового компьютера, который решает вообще одну определенную задачу – имитации отжига. Но квантовые компьютеры уже стали интересны компаниям – информационным гигантам типа Google, которые находят в них некие мультимедийные возможности.

А. А. Аксёнов: Мне кажется, что ни научное сообщество, ни IT-индустрия в чистом виде тут ничего друг другу не диктуют. Вычислительная техника и ПО создаются для решения какого-то определенного класса проблем. Но в конечном счете все определяет рынок, в самом широком смысле. Компания NVIDIA вначале стала делать свои графические карты (ускорители) для большей реалистичности компьютерных игр. Появилась технология. А потом додумались использовать графические процессоры для вычислений, что, соответственно, вновь привело к расширению рынков сбыта продукции и дало новые возможности. С моей точки зрения, любая технология подготавливается для решения какой-то определенной проблемы, а потом транслируется на целый класс задач.

А. В. Бухановский: Достаточно вспомнить определение суперкомпьютера для экономических вузов, которое гласит, что суперкомпьютер – это такая система, которая всего лишь на порядок не удовлетворяет потребности современной науки. Исходя из этого определения главенствует наука. Именно она декларирует свои нынешние и перспективные задачи, а инженеры делают и предлагают к использованию технику. Но тут начинают работать обратные процессы: «А как же предложенное использовать?» И находятся новые возможности, появляются новые задачи, порожденные ими. В этом симбиозе и есть движение.

М. В. Якобовский: Основным движителем развития становятся глобальные вопросы, «большие вызовы», которые стоят перед обществом, страной, перед человечеством в целом. А потом только выстраивается реальная цепочка «глаза в глаза» – от тех, кто проектирует, заказывает, до тех, кто использует. И в эту цепочку потом «вкачиваются» реальные деньги, хотя начало этой цепочки мне видится где-то в 1980-х годах.

В. В. Воеводин: Мы сейчас создаем настолько сложные устройства, что уже не понимаем, как они работают, и начинаем изучать их, как если бы это был «черный ящик». Сохранится такое положение вещей в будущем, или есть шанс, что архитектура суперкомпьютеров начнет так или иначе упрощаться?

Ю. Я. Болдырев: Компьютер как состоял из известной совокупности основных частей: «АУ», «УУ» (то есть арифметические устройства и устройства управления) и «памяти», так и состоит. Просто все эти функции усложнились. Появляются новые операционные системы, новые языки программирования, но основные функции системы остаются неизменными.

Ничего нового в функциях не появляется и не появится никогда. А вот эти компоненты компьютера, вне всякого сомнения, будут продолжать непрерывно совершенствоваться. Можно говорить про суперскалярность, векторизацию, иерархию памяти, функциональное устройство и так далее, но на уровне «АУ», «УУ» и «вывода в память» все остается неизменным. Архитектура компьютеров по важнейшим позициям в ближайшее время не изменится. Поэтому мне кажется, что вопрос надо свести к более простому – насколько УДОБНЕЕ станет работать на суперкомпьютере. Не «проще», а «удобнее». Это не одно и то же. И тут я уверен, что программировать станет проще.

В. В. Корнеев: Обнадежить, что жизнь будет проще, наверное, можно. Только она будет проще у 80% тех, кто пишет простые программы, а у 20% тех, кто пишет сложные, она только усложнится. Это я транслирую известное наблюдение, что 20% сотрудников делают 80% работы, ну и дальнейшие экстраполяции этой закономерности. Если бы, чисто гипотетически, мы сейчас построили компьютер с простой архитектурой, которая была у первых компьютеров, но с быстродействием на уровне современных образцов, то мы имели бы компьютер с быстродействием миллион операций в секунду. Потому что быстродействие схем памяти сейчас приблизительно на этом уровне. А мы имеем как минимум в тысячу раз больше. За счет чего? За счет того, что мы научились организовывать многоуровневую память, подкачиваем данные из «большой» памяти в «маленькую» (кэш или буферную), с которой работать умеем быстро. Надо этим пользоваться? Конечно. Механизмы эти работают, но не одинаково для каждой конкретной задачи. Поэтому инженеры заложили новую возможность: возможность делать предвыборку из памяти, разрешив программисту самостоятельно управлять пересылкой данных между уровнями памяти.

Таким образом, если используется, скажем, библиотека Intel MKL, получается эффективная программа, а если написано просто и наивно на C, используя, скажем, умножение матрицы на вектор, получается производительность в 300–400 раз меньше.

И опять же о графических процессорах. Основная суть в том, что мы приходим к синхронному выполнению операций. Если для каждой операции нам надо было бы выбирать команду из памяти, потом выполнять ее, то получили бы замедление в 15–20 раз. Потому, если хочешь получить эффективную программу, будь добр, используй архитектуру компьютера. И так будет всегда, особенно для задач, которые лежат на пределе возможностей.

А. В. Бухановский: Дискуссия о том, что мы начинаем изучать собственные компьютеры, не должна удивлять. Сложные системы требуют осознания. Можно спуститься на уровень ниже. Выходит очередная версия пакета прикладных программ, и сразу образуется сообщество, посвященное этой версии; внутри сообщества собираются группы, инициируются интернет-конференции по изучению особенностей и возможностей пакета. Через какое-то время интерес спадает, поскольку программа отлаживается, новых «артефактов» выявляется все меньше и меньше. Этот стандартный, волнообразный процесс следует за очередным усложнением, очередной версией.

В. В. Воеводин: Можете ли вы назвать области, где у российских команд (коллективов) есть достаточно опыта, чтобы к 2020 году добиться результатов мирового уровня? Интересуют конкретные примеры ПО и того, что на этой основе может быть сделано.

Ю. Я. Болдырев: У нас сейчас есть две лаборатории, работающие, безусловно, на мировом уровне. Пишется полностью собственное оригинальное ПО, и в рамках этой разработки с его помощью тестируются самые современные модели турбулентности. И это позволяет занимать сейчас ведущие позиции в мировой прикладной аэродинамике. А это архиважная тема!

Второе направление – это лаборатория вычислительной механики А.И. Боровкова, к которой в очереди стоят гранды мирового автомобилестроения, вплоть до BMW и других. Здесь крайне эффективно используют существующее на рынке коммерческое и открытое ПО. Достаточно сказать, что платформа для автомобиля класса BMW X3 здесь делается за две недели. Этого не могут ни китайцы, ни индийцы – образования не хватает, подобного такому, что дается в ведущих вузах нашей страны.

Ю. Я. Болдырев: А. И. Боровков видит применение вычислительной мощи в основном как решение массового потока многовариантных технических задач. Я же отношусь к этому немного иначе: считаю, что суперкомпьютерные технологии – это в первую очередь инструмент решения задач реального мира.

А. В. Бухановский: Мне довелось оценивать немалое количество работ, которые делаются в России, и я пришел к выводу, что в среднем каждая десятая работа без натяжек превышает мировой уровень, а 20–30% работ приблизительно соответствуют мировому уровню. На наши конкурсы проектов приходит 20–25 заявок. Отсюда можно получить приблизительную картину.

В. В. Воеводин: Хорошая оценка, с учетом того, что через Минобрнауки проходит порядка 1000 контрактов. Так что выходит примерно 250 команд, которые реально могут что-то сделать.

В. В. Корнеев: В России появляются команды, работающие в областях, браться за которые раньше не было возможности. Например, распознавание речи, естественного языка, текста, а также автоматические переводы. Они

работают по 5–7 лет и уже добились заметных результатов. Другое дело, что, сравнивая с зарубежными достижениями, надо помнить, что, скажем, английский и русский языки – это разные вещи. С русским работать сложнее, но тем не менее результаты есть.

М. В. Якобовский: Я тоже хотел привести пример области, в которой необходимы суперкомпьютеры и параллельная обработка, – это акустика. Есть такая глобальная задача, поставленная несколько лет назад: шум от самолета необходимо свести до уровня комариного писка. Для этого, конечно, надо смоделировать то, что происходит в разных узлах самолетов. У нас эта задача сейчас решается на общемировом уровне.

Есть и другие ситуации, когда нужны даже не суперкомпьютеры, а очень мощные рабочие станции, способные здесь и сейчас очень быстро решить некую специфическую задачу. В качестве примера могу привести некоторые работы нашего бюро, возглавляемого профессором В. А. Галактионовым. Так, они выполняют расчеты освещения в помещениях со сложной геометрией – например, в таких как салон самолета. Вот и другая задача, которую они решают: есть струйный принтер, известна конфигурация печатающего узла – надо оценить, какой цвет получился в результате смешения цветов. Пока для достижения идеальной цветопередачи остались самые сложные 5% пути, которые очень трудно пройти: на сегодня человек, своими руками смешивающий краски в нужной пропорции, все еще дешевле, выгоднее, но это не значит, что это и завтра будет выгоднее компьютерного подбора цвета.

В. В. Воеводин: А сейчас конкретный вопрос, очень простой: в каком году, по-вашему, технология MPI перестанет быть основой параллельного программирования, и какими чертами, на ваш взгляд, будет обладать новая технология, пришедшая на смену?

В. В. Корнеев: Я придерживаюсь той точки зрения, что чем скорее это произойдет, тем лучше. Эту технологию надо убивать на корню и возвращаться к тем технологиям параллельного программирования, что развивались в начале 1970-х. Тогда логика была такая: указывается, что можно делать параллельно, и далее предлагается следовать более-менее простой модели: к памяти по чтению можно обращаться произвольно; чтение предшествует записи; запись после чтения тоже происходит в произвольном порядке. И вот, приняв эту простую модель, можно дальше писать параллельные программы, которые будут массово создавать параллелизм.

Дальше можно использовать языки высокого уровня. Такой стиль программирования позволит создавать синхронные треды – пачки тредов, выполняемых по одной команде.

Далее появляется задача превратить обычную программу в программу для графического процессора. Этот стиль позволит эффективно загружать ресурсы, создав для программиста унифицированную аппаратно независимую модель параллельной задачи, заставив его мыслить «высокими» категориями вроде множеств и математических моделей и уйти от конкретики физических реализаций архитектур.

Есть в Америке, в Международном Университете Флориды, профессор А. Е. Мышкин, который проводит на своих студентах и аспирантах эксперименты: берет команду, программирующую традиционно на C++ или Pascal, и ребята начинают осваивать подобную модель программирования. Получается, что те, кто эту технологию освоил, и быстрее учатся параллельному программированию, и работают эффективнее и надежнее. Т. е. он на своих «подопытных» студентах доказал, что это не экзотика, а нормальная технология производства параллельных вычислений.

А. В. Бухановский: Если говорить о сроках жизни технологии MPI, то я бы отвел ей еще лет 30, и не потому, что мы очень хорошо к ней относимся, а просто из соображений жизненного цикла вычислительных моделей и соответствующего ПО.

Много что написано под MPI, отлажено, в том числе и сообществом, так что в этом уже сложно будет кому-то разобраться. И более того, сообщества эти зачастую состоят не из программистов, а из пользователей – физиков, химиков, и код написан с учетом их понимания и требований, и, как говорится, «если работает, лучше не трогать».

Переделка потребует слишком серьезных ресурсов. Вот когда изживут себя вычислительные системы, на которых работает MPI, тогда, видимо, и придется переписывать.

В. В. Воеводин: Может быть, MPI как Фортан – вечен?

А. В. Бухановский: Ну, если мы видим программы на Алголе, которые переписывают только сейчас, то что говорить об MPI!

Ю. Я. Болдырев: Во что все упирается? Стало понятно, что огромное количество задач сводится к решению систем дифференциальных уравнений в частных производных. В 1940-х годах появились численные методы, принципиально не отличающиеся от современных. Вот уже 70 лет прошло, а новых-то идей нет. Все замыкается, я думаю, на математику. Появятся, вне всякого сомнения, новые подходы, новые веяния. Должно что-то произойти, должен случиться какой-то прорыв. И тогда парадигма, о которой говорили коллеги, может измениться.

Может, это и наивный взгляд, но мы знаем, в математике и механике случились прорывы в XX веке. И нам стоит надеяться, что появятся совершенно новые идеи, связанные и с вычислительными методами.

В. В. Корнеев: Я помню времена, когда параллельных вычислений вообще не было. А тех, кто о них говорил, называли чудаками. А потом усилиями сообщества они появились, и сейчас мы о них говорим как о данности.

Но мы уже сейчас имеем задачи, которые достаточно трудно запрограммировать: нерегулярный доступ к памяти, нерегулярные сетки. Для таких задач нужны машины, где такие понятия, как MPI, неприменимы в принципе.

В. В. Воеводин: Предположим, у вас есть возможность распоряжаться суммой в 250 миллионов рублей. На какую из областей вы бы направили эти средства, какую считаете приоритетной? По шуму в зале понимаю, что вопрос задел всех за живое.

В. П. Гергель: На живые системы, исследования человека, разработку лекарств – всё это очень важно. *(Аплодисменты).*

В. В. Корнеев: Мне приходится довольно часто выступать экспертом РФФИ. Через нас проходит много проектов. Я бы направил эти деньги на проекты архиважные, но которыми занимается почему-то только Хабаровск. Это «цифровая медицина» – когда делается электронная модель человека и каждый конкретный человек может быть с помощью технических средств просканирован, и можно дальше следить за функционированием каждого его органа в отдельности. Это переводит медицину на качественно иной уровень. Движение от шаманства с постукиванием к научной деятельности на твердой основе. 250 миллионов – конечно, маловато, но нужно с чего-то начинать. В Америке это уже делается – в Техасском университете. Для такого проекта нужен суперкомпьютер, средства визуализации и подготовленные специалисты-медики.

М. В. Якобовский: Сумма маленькая какая-то. Если бы денег было много, я бы направил их на систему школьного образования. Чтобы туда шли квалифицированные кадры из вузов и спокойно там работали за хорошие деньги.

Если же «ужиматься», то направил бы эти средства на вопросы альтернативной компьютерной архитектуры. Допустим, управление базами данных – там идеи-то неплохие, а результаты пока не очень. Непонятно, как это правильно принять. Хотя одна из альтернатив может позволить проще работать с машинами, проще писать программы и получать высокую эффективность.

А. В. Бухановский: Я бы воспользовался этими средствами, чтобы поднять года за два лабораторию, которая бы занималась предсказательным моделированием в науках об обществе, но из первых принципов. Как в нанотехнологии: как научились молекулы материала делать – так и стало понятно, что происходит. Бери структуру – и клепай себе сколько тебе нужно. И с людьми то же самое, и моделировать это можно, только пока вопрос до конца не решен, потому что первый принцип – это какие же ресурсы нужны! А проблема в том, что каждый человек – это личность. И моделирование таких систем – проблема архикомпьютерная и архиматематическая, и для систем визуализации...

В. В. Воеводин: Не опасаетесь, что получится, как в нанотехнологиях: поняли, как устроено, – и потом ваяй что хочешь?

А. В. Бухановский: Ну, наверное, пройдет еще немало времени, прежде чем поймут, как устроено!

А. А. Аксёнов: У меня давно есть идея: создание платформы расчетов, которую другие исследователи могли бы включать в свои математические модели. Чтобы все это было легко программировать, чтобы все безболезненно переносилось на суперкомпьютеры. Чтобы исследователи могли создавать и внедрять свои математические модели на перспективу, не путаясь в тонкостях параллельного программирования. А цель – создать такой рынок, чтобы люди создавали свою интеллектуальную собственность и потом продавали заказчикам, производствам.

В. В. Воеводин: Последние вопросы: как вы думаете, какие темы будут актуальны на суперкомпьютерной конференции в 2020 году, и что предложите на конференцию 2014-го?

Ю. Я. Болдырев: Коллеги, мне кажется, мы уже многие годы на конференциях концентрируем внимание на параллельных вычислениях. Но настало время подумать о том, чтобы суперкомпьютеры как инструментарий использовались везде – от медицины до науки о космосе. Важнейший лозунг 2020 года – широчайшее использование суперкомпьютеров во всех сферах! А в следующем году хотелось бы, чтобы крен в сторону, которую я озвучил, уже стал достаточно большим.

В. П. Гергель: Один из главных вопросов – подготовка кадров. Кадры решают все – что в следующем, что в 2020-м году.

М. В. Якобовский: Поддержу Виктора Павловича насчет кадров. Что касается 2020 года, модели будут сильно развиты, и технические устройства, технологии можно будет легко смоделировать. Встанет вопрос: как бы нам так обучить суперкомпьютер, который к тому времени и так многое будет уметь, стать более человечным?

Камиль Ахметович Валиев: неблизкий путь к квантовым вычислениям

Все мы дети Галактики, как поется в одной подзабытой песне, но немногим из нас выпадает стать Великими Наследниками Вселенной.

Текст Родион Водейко

На дворе – январь 1931 года, а сам двор – посреди деревни Верхний Шандер Татарской АССР. Не на всякой карте найдешь, даже в эпоху Google Maps. А на заре эпохи индустриализации СССР единственный способ сориентировать обитателя, скажем, Москвы, на это селение – сказать, что это примерно в 150 км к востоку от Казани. И добавить через микропаузу: «Тьмутаракань»... И неважно, что античная Гермонасса, она же Тьмутаракань, располагалась на месте нынешней Тамани в Краснодарском крае, – «в русской разговорной речи слово Тьмутаракань ассоциируется с чем-то недосягаемо далеким и неизвестным, сродни «за семью морями», «неизвестно где» – обычно с пренебрежительным оттенком – как синоним слова “глушь”» (Wikipedia).
Вот в этой глуши и появился на свет еще один ребенок в семье Ахмета Валиева, заслуженного красного командира, по совместительству сына раскулаченного в годы коллективизации зажиточного крестьянина Мухамедзяна Валиева. Назвали малыша именем Камиль, т. е. «совершенный во всех отношениях, лучший». Что ж, прозорливости Ахмета Мухамедзяновича можно позавидовать – именно Камилю Валиеву было дано не только прославить свой род, но и стать одним из лучших людей как в своей стране, так и в своей специальности на мировом уровне.
Но время не сразу начинает лететь стремительной стрелой, и поначалу Дети Вселенной пребывают в наиболее естественном своем состоянии – в детстве. У Камиля Валиева, как и у всех почти детей той поры, оно не было безоблачным: из восьми детей Ахмета и Махруй Валиевых трое умерло, а когда Камилю было девять, его отцу в поисках лучшей доли пришлось податься в шахтеры, и семья перебирается в Саратовскую область.
Первые три класса школы Камиль к тому моменту уже закончил, причем за два года – сказалась домашняя подготовка и общая нацеленность родителей на образованность детей. Отец, немало повидавший в жизни, прекрасно понимал значение знаний, а мама сама до вступления в брак была учительницей.
Но в новой школе мальчишку ждут серьезные испытания – ведь он впервые оказывается в русском поселке, в русской школе, и к тому же не знает русского языка.
Потому Камиль снова идет в третий класс, а параллельно учит «великий и могучий», но тот велик и могуч настолько, что дело идет туго, и будущему светилу советской фундаментальной науки приходится идти в третий класс в третий раз. И в четвертый тоже.
Если вас или кого-то в вашем классе оставляли на второй год – вы можете попытаться себе представить, через что довелось пройти мальчугану, просидевшему в третьем классе четыре года! Кого-то другого это могло сломать, но для Камиля любые испытания были лишь доказательством старой мудрости: «То, что не убивает нас, делает нас сильнее». Впоследствии он будет с улыбкой говорить, что именно этот бесконечный 3-й класс заложил основу всех его дальнейших успехов.
В 1949 году юноша заканчивает школу, и перед ним встает извечный вопрос «ста путей, ста дорог». Все работы хороши, выбирай на вкус, – и он выбирает точные науки. И тут же поступает в Казанский госуниверситет. Так уж получилось, что многие известные умы, ярчайшие личности былых эпох отдали должное стенам этого храма науки, но если многие из них прославились в первую очередь на политической стезе, то Камиль Валиев, напротив, стремился уйти подальше от политики – и в этом его поддерживал, а может, и направлял отец, прекрасно понимавший все риски близости к тем, кто вершил судьбу страны.
На счастье нашего героя, КГУ всегда давал стране не только революционеров по складу личности, но и настоящих первопроходцев в науке. Благо крепкая научная школа университета была одной из ведущих в мире. Преподавателями Камиля Валиева в области физики были ученики и коллеги величайших умов того времени – будущего Нобелевского лауреата И. Е. Тамма и первооткрывателя явления магнитного резонанса Е. К. Завойского.
Сам Камиль заканчивает вуз с отличием и поступает в аспирантуру. Здесь появляются первые из его 600 с лишним научных трудов, здесь он параллельно с преподавательской деятельностью работает над своей кандидатской по теме «Магнитный резонанс на ядрах парамагнитных атомов» и в 1958 году защищает ее, получив свою первую ученую степень. Вместе со своим научным руководителем А. С. Альтшулером они проводят теоретические работы по исследованию электронной спин-решеточной релаксации для комплексообразующих ионов металлов в жидких растворах. Предложенный учеными механизм релаксации называют их именами – «механизм Альтшулера–Валиева».
Впоследствии именно магнитный резонанс на спинах станет теоретической основой наиболее современных фундаментальных и практических научных разработок Валиева в области квантовых вычислений. Чувствуете, как ускоряется время?
Нет, оно еще не столь стремительно, как поток нейтронов в Большом Адронном Коллайдере, но уже вовсю несет вперед молодого талантливого ученого: в 1959-м Валиев – доцент и завкафедрой физики Казанского педагогического института, здесь же при его активном участии создается аспирантура вуза, формируется новая научная школа, специализирующаяся на изучении проблем жидкого вещества методами магнитного резонанса и оптической спектроскопии.
Камиль Ахметович становится признанным специалистом в этой проблематике, регулярно выступает с научными докладами и сообщениями по данной тематике в Казанском и Московском университетах, ФИАНе, Институте химической физики и других научных центрах.
В 1964-м он становится доктором физико-математических наук, защитив диссертацию на соискание этой ученой степени по теме «Теоретические вопросы исследования жидкого вещества спектроскопическими методами».
И в этот самый момент, на пике его карьеры в области фундаментальной физики, жизнь готовит

Камилю Валиеву было дано не только прославить свой род, но и стать одним из лучших людей как в своей стране, так и в своей специальности на мировом уровне

ему неожиданный и судьбоносный подарок, резко меняющий почти все! По рекомендации знакомого сотрудника ФИАНа Камиль Валиев принимает предложение поработать в совершенно новой для страны отрасли – микроэлектронной промышленности.
Микроэлектроника становится настоящим вызовом для страны, опередившей своего главного соперника в космической дуэли. И для решения столь масштабных и беспрецедентных задач нужны рыцари без страха и упрека, люди незаурядного ума, несомненного организационного таланта и выдающейся решительности одновременно. Камиль Валиев подходит на эту роль как нельзя лучше. Ему предлагают место начальника

Возможно, все будущее человечества напрямую зависит от решения задач квантовых вычислений. Можно сказать, что все грядущие успехи информатики и микроэлектроники опираются на спины Первокирпичиков мироздания – элементарных частиц и великих умов далекого и недавнего прошлого.

физического сектора п/я 2015 (будущий Научно-исследовательский институт микроприборов Министерства электронной промышленности СССР), оказывают всю возможную помощь в решении бытовых вопросов, и семья амбициозного специалиста (в которой уже двое детей) переезжает в Зеленоград.
Не проходит и года, а энергия и страсть Валиева на новом месте уже обеспечивают ему личное знакомство с министром электронной промышленности А.И. Шокиным – и новый шаг по лестнице, ведущей на самый верх научно-технологической пирамиды СССР: в январе 1965 года 34-летний Камиль Валиев становится директором новейшего Научно-исследовательского института молекулярной электроники (НИИМЭ) Министерства электронной промышленности СССР. Здесь мы еще раз перекинем мостики в более близкие к нам годы – именно в составе НИИМЭ в 1967 году был создано предприятие «Микрон», ставшее настоящей советской Силиконовой долиной – более 40 лет там производились и производятся основные объемы советских, а позже и российских микросхем, используемых в самых различных приборах – от калькуляторов и бытовой техники до мощнейших ракетных комплексов «Град». Современному поколению бывшие НИИМЭ и «Микрон» известны скорее под именем Sitronics – после акционирования в 1997 году они вошли в состав концерна «Научный центр», позже получившего модное «глобализованное» название.

Но вернемся в 1960-е. Приняв предложение стать директором НИИМЭ, Камиль Ахметович Валиев круто меняет профиль своей научной работы – теперь его силы брошены на сугубо практическое направление, но именно то, где необходим мощный аналитический ум фундаментального ученого и его опыт решения сложнейших организационных задач. Родина не ошиблась в своем выборе: две недели требуется руководителю нового направления, чтобы провести оценку ситуации и составить долгосрочный стратегический план развития отрасли.
Валиев берет на себя задачу сделать НИИМЭ основным разработчиком и производителем интегральных схем в стране и за десять с небольшим лет с нуля выстраивает эффективное производство, без которого немыслима дальнейшая жизнь страны – как мирное существование, так и обороноспособность ее вооруженных сил. Именно для решения этой задачи создается «Микрон» – главная производственная площадка новой отрасли, и в 1974 году Камиля Валиева в числе шести представителей министерства награждают Ленинской премией. Теоретическая физика в руках Мастера конвейерным способом материализуется в новых технологиях и современнейшей продукции. Фактически в эти 10–12 лет страна совершила огромный скачок и практически сравнялась с зарубежными конкурентами в части высоких технологий. Нынешняя микроэлектронная промышленность России, Беларуси, Латвии и других стран, оставшихся после распада Союза, обязана своим существованием тому прорыву, что совершали люди масштаба Камиля Валиева.
Сам он писал о своей работе так: «Название института «Молекулярная электроника», далекое от его реального предназначения, было выбрано не из соображений секретности. Его «претенциозность» объясняется царившими в то время энтузиазмом и эйфорией в умах молодых людей, приступивших к созданию микроэлектроники: казалось, что путь от кремниевых интегральных схем до молекулярной электроники совсем близок: сегодня занимаемся интегральными схемами, завтра – молекулярными. Прошло 30 лет с тех пор, а время молекулярной электроники можно смело отодвигать еще на 30... Я счастлив, что мне удалось участвовать в такой программной работе, как создание микроэлектроники в СССР».
Да, в интеллектуальной и технологической элите СССР 1960-1970-х умами владел бесконечный романтизм. Да и как было не поддаться его обаянию стране, творившей настоящие чудеса по пути в макро- и микромир, на фоне того фанатичного и искреннего увлечения покорением тайн природы, которым пропитан культовый «Понедельник...» Стругацких. Это было время, когда Выбегаллы и Камнеедовы еще не стали важнее грубых Корнеевых и настоящих магистров волшебного научного цеха, когда казалось, что реальность обгоняет самые смелые творения писателей-фантастов.
И пусть они, эти магнитно-резонансные мечтатели, ошибались в своих расчетах, пусть их мечты не очень-то совпали с реальностью, – но разве можно кинуть в них за это камень? Разве не единственное возможное применение камня – поставить им памятник?
Время, разогнавшееся до второй космической и «десять в минус девятой» наноскорости, еще не стирает города и цивилизации. Но оно безжалостно к тем, кто гонит его вперед.
В 1977 году Камиль Валиев берет тайм-аут – пишет заявление с просьбой об освобождении от своего поста. Самая сложная часть задачи решена, маховик отрасли раскручен, работа кипит; однако живому человеку, какой бы энергией ни наградила его природа, нужен отдых. Нет, не праздность: как и любой великий (да и просто очень умный) человек, Камиль Ахметович находит отдых в смене деятельности. Он фактически возвращается после долгого перерыва в фундаментальную науку. Да, он не оставлял ее и во время гонки микрочиповых вооружений, но полноценно совмещать теоретическую работу с административно-практической было невозможно.
Помогают знакомые: академик Сагдеев создает в Институте космических исследований АН СССР сектор, не имеющий названия, а в штате насчитывающий одну единицу – К.А. Валиева. Целый год доктор наук наверстывает упущенное, изучает космическую пыль, заново погружается в мир формул, теорий и научного анализа. Через год он становится заведующим сектором микроэлектроники ФИАНа, в 1980-е его лаборатория получает статус отдела микроэлектроники в Институте общей физики, где Валиев получает пост замдиректора по научной работе, позже – директора-организатора. А в 1986 году профессор Валиев возглавляет Институт микроэлектроники АН СССР. Через два года его бывший отдел в ИОФАНе преобразуется в Физико-технологический институт, руководителем которого назначают, конечно, Камиля Ахметовича.
А еще три года спустя Валиев получает ученое звание академика.
Именно в ФТИ и начинается последняя масштабная работа «самого совершенного» из рода Валиевых – разработка теории и практики квантовых вычислений, развивающих идеи Ричарда Фейнмана, опубликованные в том же году, когда молодой Валиев стал кандидатом наук.
Именно эта область знаний закольцевала научную карьеру академика Валиева, ведь моделирование и проектирование квантового компьютера напрямую связано с темой исследований магнитного резонанса.
Возможно, все будущее человечества напрямую зависит от решения задач квантовых вычислений. Можно сказать, что все грядущие успехи информатики и микроэлектроники опираются на спины Первокирпичиков мироздания – элементарных частиц и великих умов далекого и недавнего прошлого.
Квантовые вычисления – удивительная область, где передовая технология неотделима от фундаментальных исследований. Создание даже логического прототипа квантового компьютера требует решения сложнейших фундаментальных физико-математических задач, а физические экземпляры подобных устройств пока весьма ограничены в числе базовых вычислительных единиц, кубитов.
Для решения же реальных задач из области квантового, атомно-молекулярного моделирования – а именно в этой сфере и имеют смысл квантовые компьютеры, так как мощность даже топовых суперкомпьютеров классического типа не позволяет получать точные результаты за сколь-нибудь приемлемое время, – необходимы системы из 1000 и больше кубитов, а достоверное создание таковых при нынешнем уровне развития технологии пока крайне проблематично, если не сказать «невозможно».
Камиль Валиев так описывал особенности этой работы: «Если у вас есть один кубит и вы им управляете, то дальше уже можете строить большой компьютер. Наша деятельность технологическая и теоретическая. Если сформулировать мою личную цель, она такая: я стараюсь создать российскую школу, работающую в этом направлении — коллектив людей, который понимал бы, что делают в мире».
Вот такие задачи решает сегодня научная школа, созданная под руководством Камиля Валиева – человека, вся научная жизнь которого была посвящена явлениям микромира и тому, как поставить их на службу миру, реальность которого мы способны воспринимать. Куда там графу Калиостро с его «материализацией чувственных идей»! Сегодняшние Чародеи творят мир будущего из того, что не только пощупать и увидеть нельзя, но и осознать сложно.
...Он ушел из жизни совсем недавно, огненно-удушливым летом 2010-го, оставив после себя – и это главное – невидимую, непонятную для большинства, но совершенно материальную основу нашей надежды на то, что Россия в XXI веке не останется на задворках передовой научной мысли и высоких технологий.
Он ушел в свой новый мир, откуда связь между атомом и космосом и их единство, наверное, выглядят вовсе не абстрактным сумасшествием Высшего разума, а его великим и прекрасным точнейшим замыслом.

Сложности перевода квантовых вычислений в классические

Текст О. В. Корж, А. Ю. Чернявский
Иллюстрация Владимир Камаев

Вычислительные технологии в XX веке стали одним из важнейших источников прогресса, их применяют во множестве областей. И если для некоторых задач текущих мощностей более чем достаточно, то для таких, например, как управление термоядерными реакциями или искусственный интеллект, требуются максимально доступные ресурсы или даже новые более производительные компьютеры. А есть задачи, которые на обычных компьютерах просто нельзя решить в силу фундаментальных причин, и одной из них является масштабное моделирование квантовой физики.

Причина проста: при добавлении каждой новой квантовой частицы в моделируемую систему число необходимых переменных увеличится не «на», а «в» количество степеней свободы частицы. То есть происходит экспоненциальный рост размерности, бороться с которым в общем случае невозможно. Но человеческий разум обладает важной способностью оборачивать некоторые проблемы и ограничения во благо: так произошло и в этом случае, и в начале 1980-х годов американский физик, нобелевский лауреат Ричард Фейнман и российский математик Юрий Манин независимо друг от друга высказали идеи о том, что природную вычислительную сложность квантовых систем можно использовать для построения нового типа компьютера – квантового.

Краткое введение в квантовые вычисления было представлено в 10-м номере журнала в статье С. Сысоева «Квантовые вычисления: От бита к кубиту», поэтому мы не будем повторяться, а приведем некоторые дополнительные сведения о квантовых вычислениях и представим некоторые факты, важные для понимания вопроса моделирования квантовых алгоритмов на классических компьютерах. Первое, на что хотелось бы обратить внимание читателей, – это задачи, которые квантовый компьютер сможет хорошо решать, когда будет построен. Многие, скорее всего, слышали об алгоритмах Гровера и Шора. Первый позволяет решить задачу перебора (решить уравнение) за корень из классического времени. Алгоритм Шора позволяет раскладывать числа на простые множители, что, например, может позволить «взломать» систему безопасности банковских операций, основанную на алгоритме RSA, – для этого понадобится квантовый компьютер из примерно 250 квантовых битов. Насколько быстр будет такой квантовый компьютер по сравнению с современными суперкомпьютерами? Допустим, мы будем раскладывать число с 250 десятичными знаками: 20-петафлопсный суперкомпьютер Cray Titan справится с задачей за год, а квантовый компьютер с частотой 1Mhz – за 4 секунды! Вполне серьезное ускорение. Но что будет с 1000 знаков? Квантовому компьютеру придется «попотеть» примерно полторы минуты, а вот современным суперкомпьютерам уже не хватит времени всей жизни вселенной.

Обычно популярные сведения о квантовых алгоритмах на этом и заканчиваются, но квантовый компьютер сможет решать и другие важные задачи. И главная из них та, от которой и возникла идея квантовых вычислений, – моделирование квантовой физики. Успех в ее решении фундаментально изменит нашу жизнь – фармакология, нанотехнологии, биохимия и многие другие области поднимутся на принципиально новый уровень. Всего же известных эффективных алгоритмов для квантового компьютера порядка 40, их актуальный список можно посмотреть на сайте Национального института стандартов США (**http://math.nist.gov/quantum/zoo/**).

А что же, собственно, представляет собой квантовое программирование? Алгоритмы задаются так называемыми квантовыми схемами, пример которой изображен на рисунке ниже. Каждая линия соответствует одному квантовому

биту (кубиту), а различным элементам на них – квантовые операции. Обычно операции действуют на один, два или максимально три кубита. Где же скрыта вычислительная мощность? На самом деле состояние n кубитов задается 2n комплексными числами (амплитудами), каждое из которых соответствует одному из состояний классических n битов, а любая операция, даже однокубитная, меняет сразу все эти амплитуды. Таким образом, мы имеем некоторый набор операций, каждая из которых содержит внутри себя порядка 2n элементарных операций (очень сильная параллельность, подаренная нам квантовой природой мира). Кроме того, мы не можем считывать амплитуды, мы лишь можем получить одно из

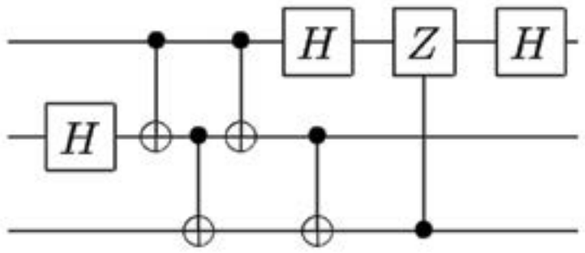


Рис. 1. Пример квантовой схемы

классических n-битных состояний с вероятностью квадрата модуля соответствующей амплитуды. Отсюда становится понятным, почему программирование для квантового компьютера катастрофически сложно и существует так мало квантовых алгоритмов. Можно провести определенную параллель с классическими высокопроизводительными устройствами: программирование для одного ядра процессора – самое простое, перейдя же к нескольким ядрам, мы получим небольшое ускорение, но и небольшие сложности программирования. Для кластеров и графических адаптеров степень параллелизма растет вместе с трудностями эффективного написания программ, а квантовый компьютер стоит практически на краю параллельности: скорость просто огромна, но каждая эффективная программа – произведение математического и алгоритмического искусства. Сложность квантового программирования – одна из причин необходимости моделирования квантовых вычислений.
Вторая и, пожалуй, еще более важная причина – трудности создания квантового компьютера. На данный момент имеется множество подходов к его реализации: ионы в ловушках, фотоны, сверхпроводники, NV-центры в алмазах и т. д., однако ни одна технология пока не преодолела трудности борьбы с декорентностью – квантовым шумом. И здесь моделирование многокубитных систем с использованием современных суперкомпьютеров становится критически важным. Начинать, естественно, необходимо с простого, поэтому первый необходимый шаг – это моделирование идеальных квантовых алгоритмов. Сколько кубитов можно моделировать, используя доступные сегодня вычислительные ресурсы? На хорошем персональном компьютере можно выполнять моделирование до 30 кубитов. Если максимально задействовать ресурсы самого мощного в России суперкомпьютера «Ломоносов», то получится выполнить моделирование работы максимум 40 кубитов. Если очень постараться и задействовать все суперкомпьютеры из актуального списка TOP50 одновременно, то получится выполнить моделирование не более 43 кубитов. Даже суперкомпьютеру экзафлопсного уровня, появление которого ожидается к 2018 году, под силу окажется моделирование лишь не более 100 кубитов.
На что же тратится так много вычислительных ресурсов? Слабых мест три. Во-первых, на каждом шаге нужно много памяти: плюс один кубит требует двухкратного увеличения памяти для хранения вектора состояний. На каждом шаге необходимо проводить изменение всего вектора состояний, поэтому возможно работать только с данными, которые хранятся в оперативной памяти, потому что чтение данных с диска существенно замедлит работу программы. Для моделирования 35 идеальных кубитов уже требуется более 1 терабайта оперативной памяти.
Во-вторых, очень много обменов данными между процессами. При этом коммуникационный шаблон нерегулярный. Большинство современных систем моделирования квантовых вычислений, написанных с использованием библиотеки MPI, имеют плохую масштабируемость, что с учетом объема оперативной памяти современных вычислительных устройств не позволяет моделировать более 35–36 кубитов.
В-третьих, время моделирования растет экспоненциально. Например, если для моделирования идеального алгоритма Гровера для 32 кубитов время эксперимента составляет не более минуты, то для моделирования этого же алгоритма для 38 кубитов на суперкомпьютере «Ломоносов» нужно порядка 10 часов.
С точки зрения параллельных вычислений базовые алгоритмы квантовых компьютеров относятся к классу DIC (Data Intensive Computing). Главным свойством задач этого класса является существенное преобладание чтения-записи данных по сравнению с количеством вычислений. При проведении каждой одно-, двух- и т. д. кубитной операции происходит изменение всего вектора состояния. При этом доступ к данным осуществляется в произвольном порядке: порядок зависит от последовательности номеров кубитов, для которых выполняется преобразование. Квантовые вычисления могут быть эффективно смоделированы на машинах с общей памятью, однако очевидно, что размер современных машин с общей памятью не позволит выполнить моделирование существенного количества кубитов. Интерес представляет моделирование квантовых вычислений на суперкомпьютере с распределенным хранением данных. В этом случае в настоящее время речь идет о максимальном моделировании 40-45 кубитов.
Использование стандартного протокола MPI для такого рода задач представляется нецелесообразным, поскольку этот протокол является неэффективным для коммуникационно сложных задач.
Эффективным представляется моделирование с помощью метода активных сообщений и, в частности, библиотеки DISLIB. Преимуществом такого подхода является возможность оптимизации пересылок данных между процессорами за счет использования низкоуровневых протоколов передачи данных в коммуникационных сетях. Такой подход позволяет написать программу с хорошей масштабируемостью на сотнях тысяч вычислительных ядер.
Еще одна трудность, с которой придется столкнуться при моделировании квантового компьютера, – это недостаточная точность вычислений. Например, вычислений с двойной точностью (тип double в языке Си) хватит для моделирования 39–40 кубитов. Далее необходимо использовать вычисления с фиксированной точкой, т. к. диапазон значений, используемых при моделировании, достаточно ограничен.
Проблему ввода-вывода в данной задаче тоже никто не отменял. Если понадобится сохранить результаты эксперимента на диск, то для 40 кубитов вектор состояний будет записываться порядка двух-трех часов.
Моделированием квантового компьютера на классической архитектуре занимается довольно большое количество исследовательских групп по всему миру. Список наиболее известных систем, позволяющих выполнять моделирование работы квантовых компьютеров, можно найти по ссылке **http://www.quantiki.org/wiki/List_of_QC_simulators.**
Таким образом, можно заключить, что перевод квантовых вычислений в классические представляет собой наукоемкую задачу, решение которой требует существенных усилий для достижения результата, а также больших вычислительных мощностей, в частности – использования самых мощных суперкомпьютеров.

Огромный интерес вызывает компания D-Wave, основанная в 1999 году. Начав с 16 кубитов, на данный момент компания заявляет о создании 512-кубитного квантового компьютера. Но этот компьютер работает не на принципах квантовых гейтов, а на так называемом квантовом отжиге и в этом смысле не является универсальным квантовым компьютером. Одно из интересных направлений работы компании – сотрудничество с Google: созданные D-Wave компьютеры используются для обучения масштабных нейронных сетей. Однако большая часть квантово-информационной научной общественности скептически относится к разработкам компании, например, подробное обсуждение можно прочесть в блоге Скотта Аароносна **(http://www.scottaaronson.com/blog/).**

Квантовый компьютер сможет решать разные важные задачи. И главная из них та, от которой и возникла идея квантовых вычислений, – моделирование квантовой физики. Успех ее решения фундаментально изменит нашу жизнь – фармакология, нанотехнологии, биохимия и многие другие области поднимутся на принципиально новый уровень

ПЛИС встраивается в облака

Текст В. Горбунов, А. Соколов, **ФГУП «НИИ «Квант»**

Сочетание программируемых логических схем (ПЛИС или FPGA), HPC и облачных технологий может показаться неожиданным. Облачные технологии – это виртуализация ресурсов ЦОД, HPC – производительность, ПЛИС – схемотехника.

Привычное место для ПЛИС – встроенные устройства и коммуникационное оборудование. Применение именно перепрограммируемой логики в этих устройствах определяется тем, что необходимо выполнять некоторые функции, которые не настолько востребованы, чтобы оправдать тиражи СБИС, или необходимо эти функции иногда изменять.
Нередко ПЛИС применяют для ускорения вычислений. Целесообразность их применения обычно возникает в случаях, когда имеются часто повторяющиеся однотипные операции над данными, если применяется алгоритм, который можно свести к простой регулярной функции, если алгоритм можно приспособить к конвейерной обработке данных. В этих условиях применение ПЛИС может дать большой выигрыш в производительности и, что существенно в современной реальности, экономии ресурсов, площади и энергии.
Энергоэффективность ПЛИС в действительности значительна. Применение ПЛИС в ряде случаев позволяет выполнять вычислительные операции, потратив в 5–10 раз (а иногда и более) меньше электроэнергии. Причем за счет свойства реконфигурации этих устройств можно добиться высокой энергоэффективности на достаточно широком наборе задач. В ряде случаев можно получить преимущество до 100 раз. При построении больших центров обработки данных для HPC (ЦОД) эту возможность ПЛИС нельзя недооценивать.
ЦОД сегодня ассоциируются с применением облачных технологий. С HPC сложнее, облачные технологии здесь пока еще недостаточно эффективны. Но прогресс в этом направлении кажется очевидным. Рассматривая все вышесказанное, можно прийти к выводу, что сочетание ПЛИС, HPC и облачных технологий не так уж и неожиданно, а, скорее всего, является новым перспективным направлением исследований.
Важный фактор актуализации этих исследований – общее снижение стоимости ПЛИС на фоне экспоненциального увеличения их возможностей. Так, например, сейчас компания Xilinx производит достаточно мощную микросхему ПЛИС Kintex-7, причем сравнение стоимости и возможности на ряде функций выше других решений, таких как CPU и GPGPU. Поэтому нередко предлагаются идеи создания вычислительных блейд-серверов, в которых вместо GPGPU-ускорителей (и даже, может быть, вместо CPU) будут установлены платы с ПЛИС.
И последняя причина, о которой стоит упомянуть, заключается в том, что в ФГУП «НИИ «Квант» создана универсальная платформа, объединяющая до 64 ПЛИС. На одной плате мы подключаем 8 ПЛИС к универсальному процессору, но легко можем перейти к соотношению 2/8 или 3/8 и т. д. Но железа для успешного облачного сервиса недостаточно. В этом контексте нас интересуют облачные технологии не как способ продажи собственных вычислительных ресурсов, а как способ рационального управления этими вычислительными ресурсами.

О режимах облачных сервисов на ПЛИС

Первый режим – когда индивидуальный пользователь ЦОД хочет получить ресурс в виде перепрограммируемой логики, нескольких или части ПЛИС. В этом случае пользователь должен уметь разработать соответствующую схему на одном из языков, которые «понимает» ПЛИС. Надо сказать, что это можно выполнять на локальных ресурсах. Далее начинается загрузка, тестирование и отладка схемы в ПЛИС – процессы с плохо предсказуемыми временными рамками, поэтому для пользователя сервиса ПЛИС как ресурса важна виртуализация – чтобы рациональным образом использовать ресурсы ЦОД.
Второй режим – использование ПЛИС как сервиса, обеспечивающего ускорение обработки данных в ЦОД. Ускорению подлежат типичные, часто повторяющиеся операции. Подразумевается, что в этом случае основные алгоритмы уже реализованы в схемотехнике ПЛИС, а сами функции заведомо востребованы и речь идет о предоставлении ресурсов ПЛИС по требованию. В этом случае уже есть настроенная виртуальная машина, которая работает с определенным количеством ПЛИС, и она должна по запросу пользователя или по какому-то внешнему событию загрузиться на свободный узел в ЦОД. Но если загрузка неравномерна, то должна существовать возможность масштабирования задачи – динамическое подключение к задаче новых ПЛИС, новых узлов. Это называется «эластичностью облака».
Третий режим – исследования новых архитектур будущих экзамасштабных суперкомпьютеров. Управление существенно не отличается от второго режима, за исключением того, что пользователю предоставляется в виде сервиса заранее наработанная модель суперкомпьютера будущего. Здесь важны свойства эластичности, виртуализации и предоставления по требованию ресурсов моделирующей гибридной вычислительной системы (МГВС – keldysh.ru/exaflops.pdf).
С учетом этих трех режимов работы, с использованием облачных технологий ФГУП «НИИ «Квант» совместно с СПбГПУ ведет разработки по виртуализации ресурса ПЛИС и интеграции их в облачные структуры.

Реконфигурируемые ускорители и облачные платформы

Для построения блоков и плат ускорителей с ПЛИС в разработках ФГУП «НИИ «Квант» применяются микросхемы компании Xilinx – в настоящее время это ПЛИС семейств Virtex-6 и Xilinx-7. Ускоритель представляет собой плату или блок с несколькими ПЛИС (ускорителями вычислений), которые по высокоскоростному интерфейсу взаимодействуют с универсальным микропроцессором. К каждой рабочей ПЛИС может быть подключено статическое и динамическое ОЗУ. Плата-ускоритель содержит необходимую инфраструктуру для обеспечения электропитания, отладочного доступа к ПЛИС, синхронизации, мониторинга технического состояния ПЛИС.
Одновременно с развитием собственной аппаратной платформы разрабатываются средства программирования ПЛИС, программные средства организации вычислений и взаимодействия универсального микропроцессора с рабочими ПЛИС, программные средства мониторинга технического состояния ПЛИС, что важно при включении аппаратуры в инфраструктуру ЦОД. Наряду с возможностью проектирования алгоритма стандартными для ПЛИС средствами схемотехнического дизайна или описания на языках VHDL и Verilog в изделиях обеспечивается возможность использования средств проектирования на алгоритмических языках высокого уровня: Mitrion-C, Catapult-C.
В настоящее время ведется разработка программных средств поддержки реконфигурируемых ускорителей с ПЛИС для облачной среды OpenStack.
Существует два пути реализации облачных сервисов с поддержкой ускорителей на ПЛИС:

- *интеграция ПЛИС в стек сервисов OpenStack, добавление дополнительных команд в сервисный интерфейс и команды утилит;*
- *создание отдельного программного сервиса, реализующего требуемую функциональность.*

Первый подход позволит интегрировать сервисы поддержки ПЛИС в облачной платформе OpenStack. Второй, более универсальный подход позволит реализовать требуемую функциональность, обеспечивающую интеграцию сервиса с разными облачными платформами, например, OpenStack, OpenNebula, CloudStack, Eucalyptus.

ПЛИС и гипервизоры XEN и KVM

В исследованиях (выполнено Алексеем Лукашиным, СПбГПУ, http://xenlet.stu.neva.ru/abrau/Papers/pdf/130.pdf, основные результаты показаны в таблицах 1 и 2 и на рис. 1) отрабатывались различные варианты использования ПЛИС в виртуальных машинах – например,

Таблица 1. Конфигурация исследовательского стенда

Операционная система	Ubuntu 12.04.2 LTS (precise) x86_64
Процессор	Intel Xeon Sandy Bridge, 2.6 ГГц
Версия гипервизора	xen-hypervisor-4.1
Версия ядра	3.2.0-39-generic

возможность подключения всех ПЛИС сервера к одной виртуальной машине, либо подключение нескольких групп ПЛИС к различным виртуальным машинам и их одновременное использование. Проводились экспериментальные исследования на сервере, к которому подключалось 16 ПЛИС. Наиболее интересным является исследование возможности использования ПЛИС в виртуальных машинах под управлением широко распространенного гипервизора Xen. Под управлением Xen виртуальные машины могут запускаться в двух режимах – паравиртуализации и полной виртуализации. Режим паравиртуализации предполагает наличие в виртуальной машине дополнительных драйверов, связанных с ядром XEN. Это позволяет обеспечить лучшую, по сравнению с полной виртуализацией, производительность при выполнении операций ввода-вывода. При использовании Xen есть возможность подключения и использования всех ПЛИС реконфигурируемых ускорителей в одну виртуальную машину, что позволяет строить достаточно мощные реконфигурируемые кластеры, управляемые виртуальной средой.
Тестирование работы ПЛИС в виртуальной машине проводилось при обмене массивами данных размером 204 800 слов. Тестирование производилось в системе без виртуализации и в виртуальных машинах под управлением гипервизоров Xen и KVM на вычислительном стенде, характеристики которого сведены в таблицу 1.
Исследования проводились для гипервизоров XEN и KVM, при этом общая схема передачи ускорителей в ВМ приведена на рис. 2. На стенде были установлены две группы ПЛИС. Были проведены эксперименты по передаче двух групп в одну виртуальную машину и одновременно по одной группе в две виртуальные машины. Полученные результаты сведены в таблицу 2.
Результаты демонстрируют возможность использования реконфигурируемых ускорителей с ПЛИС в виртуальном окружении. Остается проблема, заключающаяся в некоторой потере средней пропускной способности и довольно большом разбросе значений. Иногда пропускная способность близка к максимально возможной. В некоторых режимах пропускная скорость заметно ниже. Возможно, проблема в том, что тестирование проводилось в режиме полной виртуализации. Уже запланирован шаг исследований – тестирование в режиме паравиртуализации. Уверенность в улучшении скорости обмена при виртуализации ПЛИС подтверждается результатами других научных коллективов, которые решают схожие проблемы. Например, группа университета г. Оттавы (http://www.sciencedirect.com/science/article/pii/S1383762112000197) уже добилась определенных результатов в паравиртуализации ПЛИС, предоставляемых виртуальной машине по интерфейсу PCI-E.

Таблица 2. Результаты проведенных тестов

Гигабайт в секунду, 10^9 байт/с

	Без виртуализации		Виртуальная машина XEN		Виртуальная машина KVM
Количество ПЛИС (шт.)	16	8	16	8	3
Среднее значение	4,1	2,1	2,6	1,7	0,45
Максимальное значение	4,5	2,2	4,3	2,3	0,47
Минимальное значение	3,9	2,0	1,1	0,67	0,38

Перспективы

Исследования возможностей предоставления и использования ресурсов ПЛИС в облачных сервисах идут во многих научных лабораториях, о чем можно прочитать в ведущих изданиях мира (например, в Journal of Systems Architecture). Подобные разработки ведутся в университете SASTRA в Индии, в Испании, а также в уже упомянутой ранее Канаде, где исследователям удалось добиться хороших показателей с помощью паравиртуализации на базе гипервизора Xen. Имеются результаты практического использования ПЛИС в облаках – декодирование популярного кодека H.264, реализованного на базе ПЛИС в облачном окружении (https://biblio.ugent.be/input/download?func=downloadFile&recordOId=2913079&fileOId=2913080).
Это означает, что исследуемая нами тема актуальна, и преимущества использования ПЛИС в ЦОД и HPC замечены не только нами. Сочетание ПЛИС, облачных технологий и HPC в перспективе позволит решать множество проблем использования и удешевления ресурсов, предоставляемых будущими центрами обработки данных как исследовательского, так и общего назначения.

Реконфигурируемые ускорители в среде облачных вычислений

А. Лукашин

**Популярность облачных сервисов Amazon положила начало развитию технологий облачных вычислений в частном сегменте и у публичных провайдеров.
В 2008 году появился некоммерческий проект Eucalyptus, который позволял развернуть частное облако с интерфейсами веб-сервисов, совместимыми с интерфейсами AWS. А это означало, что появилась возможность запускать множество приложений, разработанных для AWS в собственной инфраструктуре. Вслед за Eucalyptus появилось множество других облачных решений: OpenNebula, ownCloud, CloudStack, OpenStack. Отдельного внимания заслуживает сервис OpenStack. Ввиду плохой масштабируемости популярного тогда Eucalyptus и в силу его архитектурных ограничений компанией Rackspace совместно с NASA было принято решение о разработке нового облачного решения, которое может масштабироваться до тысяч узлов виртуализации.
Сотрудничество Rackspace и NASA со временем переросло в глобальное сообщество разработчиков программного обеспечения, сотрудничающих в сфере создаваемых облачных технологий на базе существующих операционных систем.
Главная цель сообщества – обеспечить любую организацию возможностью создавать и предлагать услуги облачных вычислений при условии работы со стандартными аппаратными средствами.
Проект OpenStack на сегодняшний день крайне популярен и привлекает все больше разработчиков со всего мира, из самых разных организаций.
Однако у проекта есть ряд ограничений и недоработок, которые требуют решения. Одна из них – ориентация на стандартное оборудование. Тем не менее существует множество задач, которые требуют поддержки нестандартных устройств в вычислительных ресурсах облака. Это могут быть как простые USB-устройства (например, для таблеток-активаторов лицензионного программного обеспечения), так и вычислители на базе ПЛИС и GPGPU или облака на базе процессоров, альтернативной по отношению к x86 архитектуре. Например, появляются облака, построенные на ARM-процессорах.**

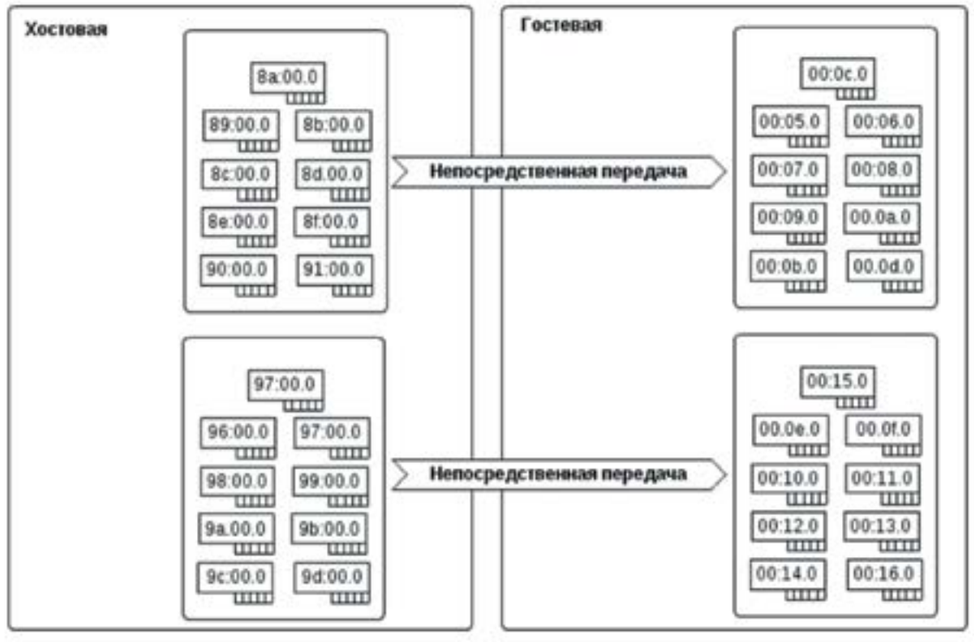


Рис 1. Схема передачи устройств в виртуальную машину

О моделировании процесса обледенения линий электропередач

Текст А.С. Поздняков, А.Ю. Чулюнин

В регионах со сложными климатическими условиями при строительстве инженерных сооружений необходимо учитывать ряд критериев, отвечающих за надежность и безопасность строительных объектов. Эти критерии, в частности, должны учитывать атмосферные и климатические факторы, которые способны негативно влиять на состояние конструкций и процесс эксплуатации сооружений. Одним из таких факторов является атмосферное обледенение.

Обледенение — процесс образования, отложения и нарастания льда на поверхностях различных объектов. Оно может возникать в результате намерзания переохлажденных капель или мокрого снега, а также путем непосредственной кристаллизации содержащегося в воздухе водяного пара. Опасность данного явления для строительных объектов заключается в том, что образовавшиеся на его поверхностях ледяные наросты приводят к изменению заложенных при проектировании характеристик конструкций (вес, аэродинамические характеристики, запас прочности и пр.), что влияет на долговечность и безопасность инженерных сооружений.

Особое внимание вопросу обледенения необходимо уделять при проектировании и строительстве линий электропередач (ЛЭП) и линий коммуникаций. Обледенение проводов ЛЭП нарушает их нормальную эксплуатацию и зачастую приводит к серьезным авариям и катастрофам (рис. 1).

Отметим, что проблемы обледенения ЛЭП известны давно, и существуют разнообразные методы борьбы с ледяными наростами. К таким методам относятся покрытие специальными антиобледенительными составами, плавление за счет нагрева электрическим током, механическое удаление наледи, зачехление, профилактический подогрев проводов. Но не всегда и не все эти методы бывают эффективны, зачастую они сопровождаются большими затратами электроэнергии.

Для определения и разработки более эффективных способов борьбы с обледенением необходимо знание физики этого процесса. На ранних стадиях разработки нового объекта необходимо проводить изучение и анализ влияющих на процесс факторов, характера и интенсивности отложения льда, теплообмена обледеневающей поверхности, определение потенциально слабых и наиболее подверженных обледенению мест в конструкции объекта. Поэтому умение моделировать процесс обледенения при различных условиях и оценивать возможные последствия данного явления является актуальной задачей как для России, так для мирового сообщества.

Рис. 1. Последствия обледенения ЛЭП

Роль экспериментальных исследований и численного моделирования в задачах обледенения

Моделирование обледенения ЛЭП – это масштабная задача, при решении которой необходимо учесть множество глобальных и локальных характеристик объекта и окружающей среды. К таким характеристикам относятся: протяженность рассматриваемого участка, рельеф окружающей местности, профили скорости воздушного потока, значение влажности и температуры в зависимости от расстояния над поверхностью земли, теплопроводность кабелей, температуры отдельных поверхностей и т. д.

Создание полной математической модели, способной описать процессы обледенения и аэродинамики обледененного тела, является важной и чрезвычайно сложной инженерной задачей. На сегодняшний день многие из существующих математических моделей построены на основе упрощенных методик, где заведомо вносятся определенные ограничения или не учитывается часть влияющих параметров. Основой подобных моделей в большинстве случаев являются статистические и экспериментальные данные (в том числе и стандарты СНИП), полученные в ходе лабораторных исследований и длительных натурных наблюдений.

Постановка и проведение многочисленных и многовариантных экспериментальных исследований процесса обледенения требует существенных финансовых и временных затрат. Кроме того, в ряде случаев получить экспериментальные данные о поведении объекта, например в экстремальных условиях, просто невозможно. Поэтому все чаще прослеживается тенденция дополнения натурного эксперимента численным моделированием.

Анализ различных климатических явлений с помощью современных методов инженерного анализа стал возможен как с развитием самих численных методов, так и с бурным развитием HPC-технологий (технологии высокопроизводительных вычислений – High Performance Computing), реализующих возможность решения новых моделей и масштабных задач в адекватные сроки. Инженерный анализ, проводимый с помощью суперкомпьютерного моделирования, обеспечивает получение наиболее точного решения. Численное моделирование позволяет решать задачу в полной постановке, проводить виртуальные эксперименты с варьированием различных параметров, исследовать влияние множества факторов на исследуемый процесс, моделировать поведение объекта при экстремальных нагрузках и т. д.

Современные высокопроизводительные вычислительные комплексы при грамотном применении расчетных инструментов инженерного анализа (в данной работе использовался STAR-CCM+) позволяют получать решение в адекватные сроки и в реальном времени отслеживать ход решения задачи. Тем самым значительно снижаются затраты на проведение

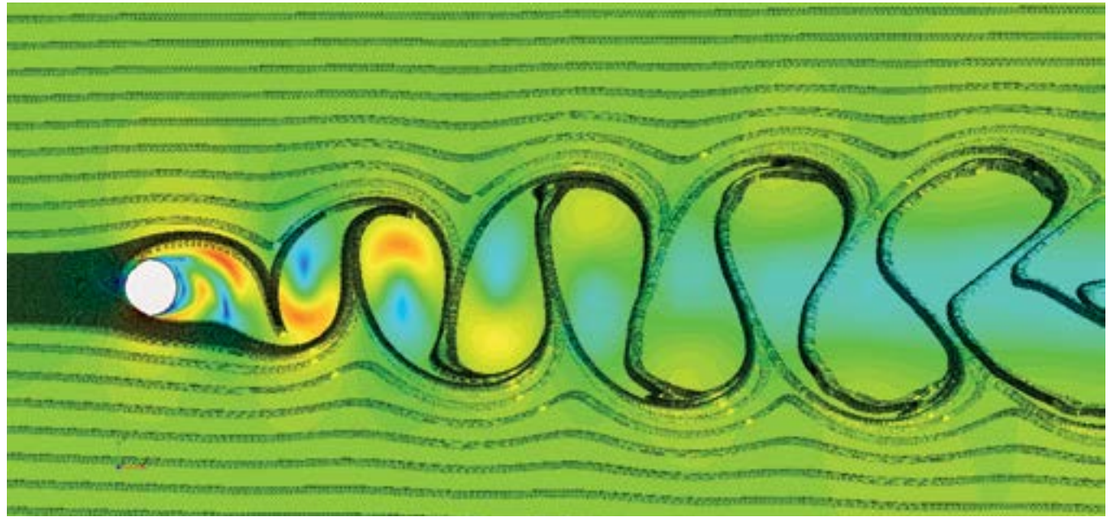


Рис. 2. Траектории капель и скалярное поле абсолютной скорости воздуха

многовариантных экспериментов с учетом многокритериальных постановок. Натурный эксперимент в данном случае можно использовать только на финальных стадиях исследований и разработок, в качестве верификации численно получаемого решения и подтверждения отдельных гипотез.

Компьютерное моделирование процесса обледенения

Для моделирования процесса обледенения используется двухэтапный подход. Первоначально проводится расчет параметров потока несущей фазы (скорость, давление, температура). После этого рассчитывается непосредственно процесс обледенения: моделирование осаждения капель жидкости на поверхность, расчет толщины и формы слоя льда. По мере роста толщины слоя льда происходит изменение формы и размеров обтекаемого тела и выполняется пересчет параметров потока с использованием новой геометрии обтекаемого тела. Вычисление параметров потока рабочей среды происходит за счет численного решения системы нелинейных дифференциальных уравнений, описывающих основные законы сохранения. Такая система включает уравнение неразрывности, уравнение количества движения (Навье-Стокса) и энергии. Для описания турбулентных течений пакет использует осредненные по Рейнольдсу уравнения Навье-Стокса (RANS) и метод крупных вихрей LES. Коэффициент перед диффузионным членом в уравнении количества движения находится как сумма молекулярной и турбулентной вязкостей. Для вычисления последней в настоящей работе используется однопараметрическая дифференциальная модель турбулентности Spallart-Allmaras, которая находит широкое применение в задачах внешнего обтекания.

Моделирование процесса обледенения осуществляется на основе двух заложенных моделей. Первая из них – модель плавления и затвердевания. Она не описывает явным образом эволюцию границы раздела «жидкость-лед». Вместо этого используется формулировка энтальпии для определения той части жидкости, в которой образуется твердая фаза (лед). При этом поток должен описываться моделью двухфазного течения.

Второй моделью, позволяющей спрогнозировать образование льда, является модель тонкой пленки, которая описывает процесс осаждения капель на стенки обтекаемого тела, тем самым позволяя получать поверхность смачивания. Согласно данному подходу в рассмотрение включается набор лагранжевых жидких частиц, которые обладают массой, температурой и скоростью. Взаимодействуя со стенкой, частицы в зависимости от баланса тепловых потоков могут либо увеличивать слой льда, либо уменьшать его. Другими словами, моделируется как обледенение поверхности, так и плавление ледяного слоя.

В качестве примера, иллюстрирующего возможности пакета STAR-CCM+ для моделирования обледенения тел, рассматривалась задача обтекания цилиндра потоком воздуха со скоростью U = 5 м/с и температурой T = –15 °C. Диаметр цилиндра составляет 19,5 мм. Для разбиения расчетной области на контрольные объемы использовался многогранный тип ячеек с призматическим слоем у поверхности цилиндра. При этом для лучшего разрешения следа после цилиндра использовалось локальное сгущение сетки. На первом этапе решения задачи с использованием модели однофазной жидкости были рассчитаны поля скоростей, давлений и температур для «сухого» воздуха. Полученные результаты имеют качественное согласование с многочисленными экспериментальными и численными работами по однофазному обтеканию цилиндра.

На втором этапе в поток инжектировались лагранжевы частицы, моделирующие наличие мелкодисперсных капель воды в потоке воздуха, траектории которых, а также поле абсолютной скорости воздуха представлены на рис. 2. Распределение толщины льда по поверхности цилиндра для различных моментов времени показано на рис. 3. Максимальная толщина ледяного слоя наблюдается около точки торможения потока.

Следует отметить, что время, затраченное на расчет двумерной задачи (физическое время t = 3600 с), составило 2800 ядро-часов при использовании 16 вычислительных ядер. Столько же ядрочасов необходимо, чтобы посчитать в трехмерном случае только t = 600 с. Анализируя временные затраты на расчет тестовых моделей, можно сказать, что для расчета в полной постановке, где расчетная область будет состоять уже из нескольких десятков миллионов ячеек, где будет учитываться большее число частиц и сложная геометрия объекта, потребуется значительное увеличение требуемых аппаратных вычислительных мощностей. В этой связи для проведения полного моделирования задач трехмерного обледенения тел необходимо применение современных HPC-технологий.

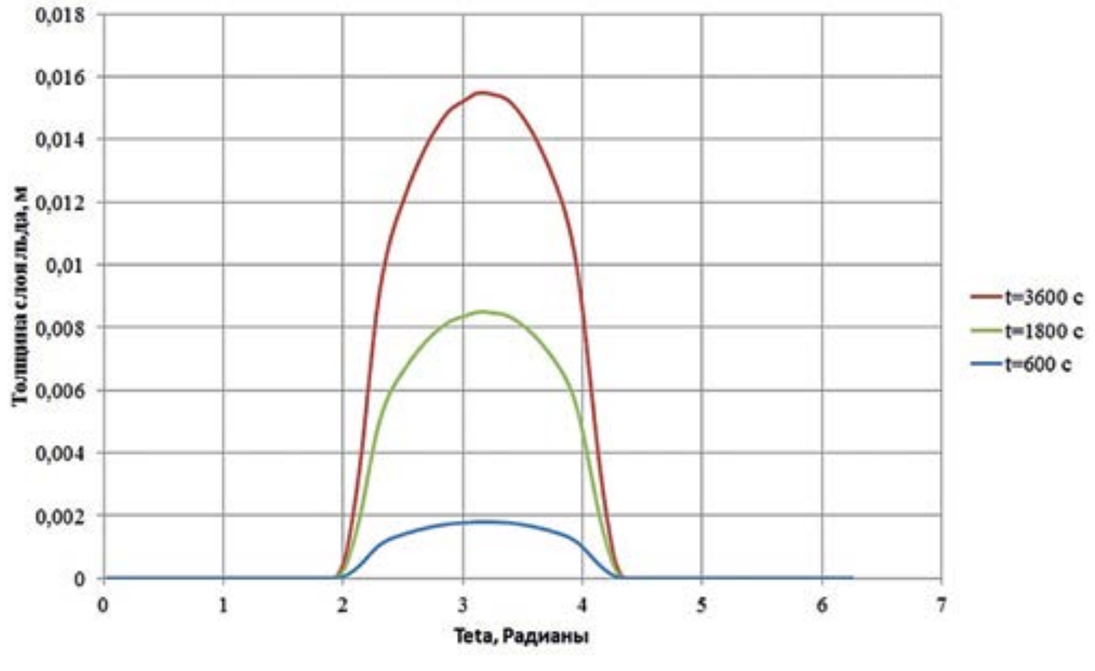


Рис. 3. Толщина слоя льда в различные моменты времени

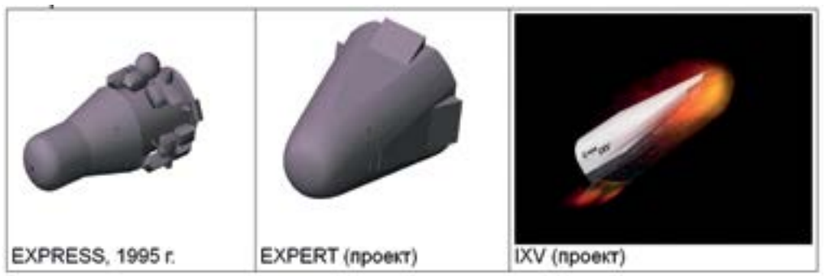
Наш журнал представляет победителей конкурса «GPU: серьезные ускорители для больших задач». Выбор журнала – два проекта: «Моделирование течения разреженного газа методом ПСМ на ГПУ», авторы – А. В. Кашковский, А. А. Шершнёв, П. В. Ващенков, и «GPU для решения СЛАУ: ускорение инженерных расчетов», авторы – Б. И. Краснопольский, А. В. Медведев.
Статью о первом проекте мы и публикуем в этом номере. Статью о втором проекте вы сможете прочитать в следующем.

Моделирование спуска с орбиты на GPU

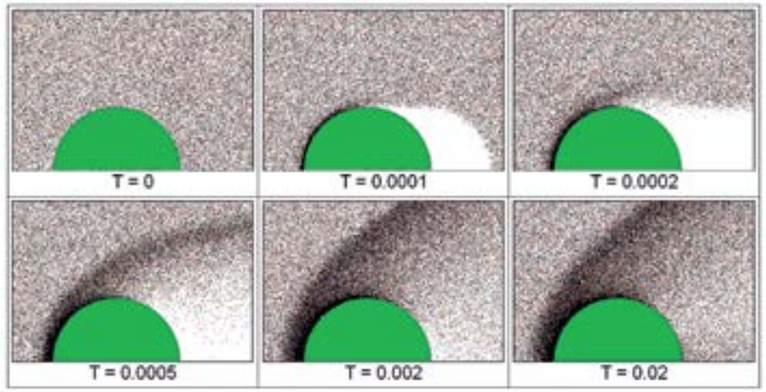
Текст А. Кашковский, А. Шершнёв, П. Ващенков

При спуске с орбиты возвращаемые космические аппараты (КА) должны уменьшить свою скорость с орбитальной (7.5–8 км/с) до посадочной (практически нулевой). Для этого используется аэродинамическое сопротивление, которое пропорционально квадрату скорости. На высотах 60–100 км, когда скорость КА еще достаточно велика, а плотность атмосферы уже существенно увеличилась, торможение наиболее интенсивно, а КА подвержен наибольшему аэродинамическому и тепловому воздействиям и максимальным перегрузкам. Обеспечить высокую экономическую эффективность и безопасность эксплуатации разрабатываемых КА невозможно без скрупулезного учета всех воздействий на конструкцию и выбора оптимальной траектории движения, которая снизила бы эти воздействия. В наземных условиях чрезвычайно тяжело смоделировать космические скорости в практически вакуумных условиях.

Полетный эксперимент, осуществляемый исследовательскими спускаемыми аппаратами (на рисунке экспериментальные КА, в разработке которых нам довелось участвовать), достаточно дорогой, так как необходимо создать и запустить такой аппарат.
Поэтому объем вычислений при разработке КА стремительно возрастает и требует более точного моделирования природных явлений и быстрого получения результатов. На высотах 70–100 км атмосфера сильно разрежена и постулат о непрерывности газового течения не выполняется, что не позволяет пользоваться методами, основанными на решении уравнений Эйлера или Навье-Стокса, и вынуждает применять кинетические подходы, трактующие течение как поток отдельных молекул газа.
«Лаборатория вычислительной аэродинамики» ИТПМ СО РАН давно занимается численными исследованиями течений разреженного газа на основе кинетических методов. Программные продукты, созданные в этой лаборатории, используются как в российских космических организациях (в



частности, в РКК «Энергия»), так и за рубежом. Опыт моделирования течений на этапе запуска, нахождения на орбите и спуска с орбиты (станции «Мир», «МКС», КА «Прогресс», «Союз», «Клипер», ППТС и многие другие) отчетливо показал необходимость значительного увеличения скорости вычислений при сохранении точности расчетов. Только в этом случае за разумное время обеспечивается большое количество расчетных вариантов, необходимых для оптимизации конструкции КА и траектории движения.



Метод прямого статистического моделирования (ПСМ, в английской литературе – Direct Simulation Monte-Carlo, DSMC) является наиболее известным из кинетических методов и ставшим «стандартом де-факто» для исследования течений разреженного газа. Метод ПСМ – это стохастический (вероятностный) численный метод решения кинетического уравнения Больцмана для конечного числа Кнудсена. Традиционно рассматривается как метод компьютерного моделирования движения большого количества частиц, представляющих газовое течение. Каждая частица является компонентой газа, имеет координаты в пространстве и скорость. Моделирование ведется по времени, дискретными интервалами Δt, на каждом из которых выполняется:

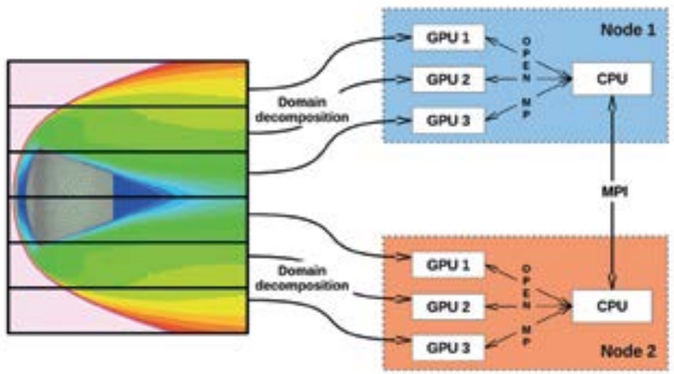
- *перенос каждой частицы с ее скоростью на шаг Δt;*
- *моделирование бинарных столкновений частиц, которые изменяют скорости частиц.*

Расчетная область прямоугольная и разбивается равномерной сеткой на достаточно малые ячейки. Сталкиваться могут только частицы, находящиеся в одной ячейке. Кроме того, в ячейках собирается статистическая информация о поле течения. Использование такой сетки существенно облегчает поиск ячейки, в которой находится частица. Вычисления методом ПСМ проводятся методом установления: начинаются с равномерного поля течения, которое, взаимодействуя с телом, изменяет свою структуру и постепенно приходит к стационарному течению. После выхода на стационарное состояние можно накапливать статистическую информацию. Например, на рисунке ниже представлена эволюция частиц при моделировании обтекания двумерного цилиндра 3-компонентным потоком разреженного газа на скорости 7500 м/с и высоте 90 км. Частицы N2 имеют черный цвет, O2 – красный, O – зеленый.
Метод ПСМ позволяет довольно подробно моделировать физико-химические процессы, происходящие при обтекании КА (неравновесность внутренней энергии молекул, диссоциацию, ионизацию и т. д.), но является чрезвычайно затратным по вычислительным ресурсам. Для моделирования течений газа вокруг КА на высотах менее 100 км требуются параллельные вычисления с использованием 500 и более процессоров. Такие вычисления можно проводить только на больших вычислительных серверах. Фактически, каждый такой расчет становится уникальным, что мешает массовости использования метода ПСМ. В связи с этим любые алгоритмы ускорения или повышения эффективности параллелизации всегда остаются актуальными.

Графические процессорные устройства (GPU, или Graphic Processor Unit) – один из наиболее перспективных подходов к увеличению скорости вычислений. GPU использует вычислительную технологию SIMD (одна команда – множество данных), которая подразумевает, что одна и та же инструкция одновременно применяется к множеству данных.
Это означает, что в случае, например, логического ветвления все вычислительные потоки будут выполнять оба списка вычислений, только не подходящие по условию результаты будут удаляться. В случае выполнения циклов все потоки будут проводить вычисления по числу итераций самого большого цикла, а те, у которых цикл заканчивается раньше, будут производить пустые вычисления. На ГПУ имеется несколько типов памяти, существенно отличающихся по скорости доступа и размеру. Это приводит к тому, что при многократном обращении к какому-либо набору данных выгоднее сначала скопировать их в более быструю память. Эти особенности накладывают специфику программирования на GPU.
Вычисления на GPU, разработанных компанией NVIDIA, удобнее производить с использованием программно-аппаратной архитектуры CUDA (Compute Unified

Device Architecture – унифицированная архитектура вычислительных устройств). Эта архитектура предоставляет программисту высокоуровневый интерфейс, скрывая от него низкоуровневые драйверы к реальным видеопроцессорам. Это позволяет работать с унифицированным виртуальным устройством и управлять вычислениями посредством широко распространенных языков (C/C++, Fortran), что существенно облегчает разработку программ.

Трехмерная программа моделирования разреженных течений методом ПСМ на GPU – наша инициативная разработка. В дальнейшем она была поддержана контрактом с ЦНИИмаш. Программа написана на языке C++/CUDA и рассчитана на применение на гетерогенном кластере, в составе которого имеется несколько вычислительных узлов; на каждом из них имеется несколько GPU. Между отдельными GPU используется метод параллелизации «разделение области» (domain decomposition). Вся



расчетная область разбивается на подобласти по числу GPU. Если в процессе перемещения частицы перелетают в подобласть другого GPU, они накапливаются в буфере, и после переноса всех частиц осуществляется отправка и прием буферов. Для GPU на одном узле пересылка осуществляется через память CPU, а между узлами – посредством MPI. Управление несколькими GPU в пределах одного узла осуществляется с помощью OpenMP. В пределах одного GPU параллелизация делается методом «распараллеливания по данным» (data parallelism).

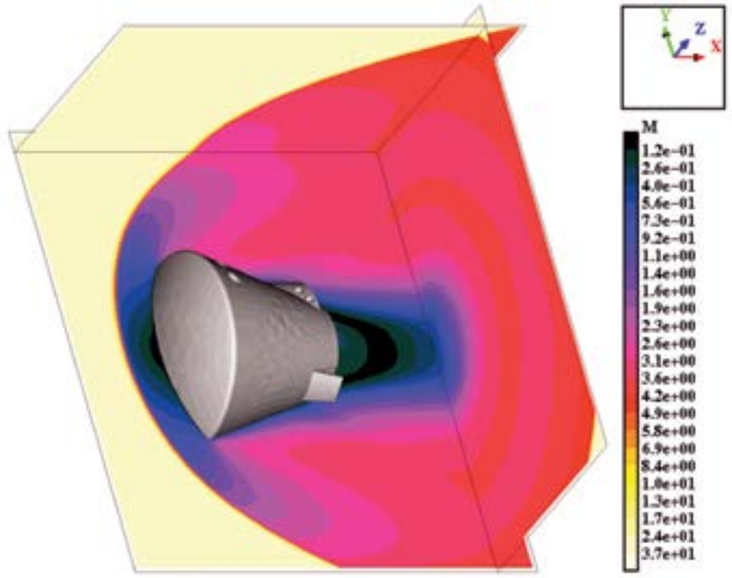
Вычисления представляют собой последовательный вызов GPU-функций (называемых ядро, или kernel), которые выполнят свою часть вычислений над заранее загруженными данными. Использование нескольких небольших kernel представляется более удобным, чем одна большая. Это упрощает отладку программы и поиск более оптимальных алгоритмов.

При создании данной программы были отработаны алгоритмы параллельной индексации частиц (составление списка частиц в каждой ячейке), параллельного удаления частиц, вылетевших из подобласти, автоматическая адаптация сетки под конфигурацию течения, динамическая балансировка загрузки GPU, пересылка частиц и многое другое. Все это позволило получить достаточно эффективную программу. Например, в процессе проектирования Перспективной Пилотируемой Транспортной Системы (ППТС) с помощью вычислительного программного комплекса SMILE был произведен расчет аэродинамических характеристик (АДХ) этого КА для высоты 80 км на 128 процессорах Intel Xeon E5420 @ 2.50GHz. Параллелизация в этом комплексе построена на методе «разделения области» с обменом частиц с помощью MPI. На рисунке ниже показано распределение чисел Маха в поле течения. Необходимо отметить, что данный расчет выполнялся в 2006 году на доступном оборудовании, и поэтому сильно загрублен: число ячеек и частиц примерно в 100 раз меньше от требуемого количества. Но тем не менее погрешность в АДХ не превышает 5–7%, что в любом случае точнее, чем использование инженерных методов. Данный расчет был повторен с использованием 24 GPU Tesla M2090. В таблице дано сравнение используемых ресурсов. На CPU использовалось больше памяти, потому что часть информации (геометрическая модель и т. п.) на всех процессорах дублируется. Большее число ячеек на GPU связано с различием алгоритмов адаптации сетки в CPU- и GPU-программах. Использование GPU позволило вместо более чем 2-суточных вычислений получить результат менее чем за 8 часов (одна ночь). При этом использовалось в 5 раз меньшее число вычислительных устройств. Экономия в процессоро-часах – почти в 36 раз. Расчеты других практических задач показали, что один GPU оказался эквивалентен 30–40 CPU. Таким образом, вычисления на GPU существенно ускоряют численные исследования высотной аэродинамики.

Использование GPU имеет ряд других принципиальных преимуществ.

1. Традиционно параллелизация метода ПСМ строится на принципе «разделения области» с пересылкой частиц с помощью MPI-протокола. Особенностью такого подхода является то, что число частиц в каждой подобласти постоянно меняется: они двигаются, перелетая из одной подобласти в другую. Поэтому вычислительная загрузка в одной и той же подобласти на каждом шаге разная, и всегда есть ожидание каким-либо процессором завершения работы всех остальных процессоров. Причем вполне возможно, что на одном шаге процессор 1 ожидает завершения работы процессора 2, а на следующем шаге, наоборот, процессор 2 ожидает завершения работы процессора 1.

С увеличением частиц в подобласти уменьшается статистическая флуктуация числа частиц (время ожидания), увеличивается вычислительная загрузка процессора по отношению к времени обмена. Это увеличивает эффективность параллелизации. Так как число GPU меньше, а число частиц на них больше, то большая скорость вычислений методом ПСМ обеспечивается не только за счет мощности мультипроцессоров, но и благодаря более высокой эффективности параллелизации.



	CPU	GPU	CPU/GPU
Устройств	128	24	5.3
Частиц, млн	215	216	
Ячеек, млн	18	42	
Время вычислений, ч	51	7.75	6.4
Процессоро-часы	6500	182	35.7
Память, Гб	175	127	1.4

2. Для корректного расчета рассмотренной выше задачи необходимо порядка 4 миллиардов ячеек и 20 миллиардов частиц. Экстраполируя данные, представленные в таблице, эту задачу можно было бы решить, используя 1600 CPU за 17 суток или 560 GPU за 32 часа. Видно, что, используя CPU, такую задачу более-менее реально можно решить, если бы они были в 8–10 раз быстрее – иначе время счета слишком велико для инженерного применения.

В то же время при использовании GPU данная задача выглядит вполне решаемой. Таким образом, GPU существенно расширяет возможности применения метода ПСМ.

3. Исторически сложилось так, что все космические программы развивались в режиме секретности. И до сих пор многим КБ запрещено использовать для вычислений внешние ресурсы, и вычисления проводятся только на внутренних серверах.

Очевидно, что кластер из, например, 4 узлов по 3 GPU будет дешевле чем, скажем, кластер из 60 восьмиядерных процессоров. Существенно дешевле будет и его техническая эксплуатация. Поэтому приобретение кластеров с GPU более выгодно таким организациям.

Использование GPU позволит при относительно небольших финансовых затратах получить возможность проводить численные исследования аэродинамики КА.

4. Видеокарта с 1 Гб памяти в составе офисного компьютера способна проводить вычисления методом ПСМ, используя до 2 миллионов частиц. Это позволит осуществлять моделирование течений от 90–95 км и выше, вообще не прибегая к услугам вычислительных центров. Это позволяет высвободить ресурсы вычислительных кластеров для других задач.

Таким образом, использование GPU является перспективным направлением вычислительной аэродинамики больших высот. Нет никакого сомнения, что в ближайшем будущем GPU будет использоваться в разработке новых КА.

Использование высокоточных таймеров в системах управления ветрогенераторами

Текст
И. С. Федотова, аспирант факультета ИВТ, СибГУТИ, г. Новосибирск
Э. Сименс, профессор Университета прикладных наук (Hochschule Anhalt), г. Кётен, Германия

Ветроэнергетика – бурно развивающаяся отрасль. Уже более 3% потребляемой электроэнергии в мире производится с помощью ветроэнергетических установок. В отдельных странах, где существует поддержка этого сектора государством, доля энергии, полученной из альтернативных источников, уже достигает более 30%. И одной из лидирующих стран в этой области является Германия.

Usage of high-precision timers in the wind turbines control systems

Text I. Fedotova, PhD student, SibSUTIS, Novosibirsk, Russia
E. Siemens, Prof. Dr.-Ing., Anhalt University of Applied Sciences, Koethen, Germany

The capacity of energy produced by wind turbines is growing every year and consequently the requirements for the development of an effective control system of wind power plant are growing as well. At this paper some possible solutions to reduce time response and improve the time accuracy of wind turbines control systems are suggested.

По итогам конца 2012 года Германия занимает 3-е место (после Китая и США) по объему установленной мощности ветряных электростанций. Всего в стране электричество производят уже 23 тысячи ветряков. Их общая мощность по итогам 2012 года выросла на 20%, достигнув 31,3 ГВт. При этом две трети ветряков, производимых в Германии, идет на экспорт, что, как следствие, вызывает высокий уровень конкурентной борьбы. И нередко в вопросах повышения эффективности производители обращаются к университету и научно-исследовательскому центру. Университет Hochschule Anhalt города Кетена сейчас сотрудничает с двумя компаниями по производству ветрогенераторов мощностью до 30 КВт и предоставляет возможность студентам и аспирантам российских вузов успешно выполнять свои научные диссертации в рамках данных проектов. Результаты работы позже внедряются на предприятиях. Университет находится в германской федеральной земле Саксония-Анхальт, где в 2012 году почти половина всей электроэнергии была выработана ветрогенераторами. Поэтому именно здесь вопросы повышения эффективности работы данной отрасли стоят особенно остро. Решить их возможно, в частности, при обеспечении продолжительного функционирования ветроэлектрических агрегатов, оптимизации локального использования энергоснабжения и стабилизации частоты электроэнергии, которую вырабатывают ветроустановки.

Основные направления разработки

Главным элементом системы автоматизации и управления является микроконтроллер. Он управляет многими процессами ветроустановки, такими как поворот лопастей, заряд аккумуляторов, защитные функции и т. д. Весьма перспективным в последнее время считается контроллер с ядром ARM, работающий под ОС Linux. Одна из текущих задач в лабораториях университета Анхальт – это переход на встроенные системы SoC (System on Chip) с архитектурой ARM-ядра, которое и будет непосредственно управлять турбиной. Система подключается к последовательному интерфейсу RS-485, и все параметры обрабатываются в реальном времени. Далее, через веб-интерфейс пользователь отслеживает состояние системы: направление и скорость ветра, направление челнока, количество вырабатываемой энергии и т. д. Далее происходит проверка эргономичности (так называемый usablity check). К примеру, если при достаточно сильном ветре поворот лопастей составляет 90 градусов, то через GSM поступает сообщение о возможных проблемах. На рис. 1 изображена упрощенная схема системы контроля турбины.

Таблица 1. Максимальное время ответа на Profibus при передачи 65 535 бит в секунду

Битрейд канала, бит/с	Максимальное время ответа, мс
9600	6826
19200	3413
187500	349
500000	131
1500000	43

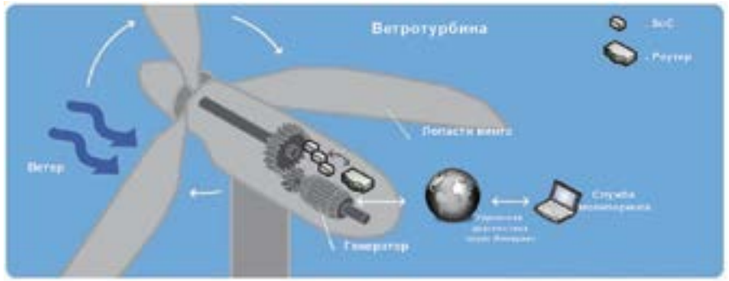


Рис. 1. Упрощенная схема системы контроля ветротурбины

С помощью разработанной на подобных принципах системы оповещения и контроля были обнаружены некоторые неточности в системе работы ветрогенератора. Так, например, система одной из партнерских компаний, пытаясь найти оптимальное положение против ветра, постоянно варьировала поворот на 30 градусов, в результате чего на выходе давала только 50–60% номинальной мощности.
Кроме того, разработка подобной микроконтроллерной системы управления ветрогенератора должна отвечать требованиям стандартов сети, которая связывает все основные устройства турбины. Так, например, согласно популярному стандарту промышленной сети Profibus скорость передачи данных варьируется от 9,6 Кбит/с до 12 Мбит/с. При этом максимальное время ответа зависит от значения битрейта канала.
А при отсутствии запрашиваемого ответа и реакционном цикле 52 бита гарантированное время отклика составит примерно 6 мс, со скоростью передачи данных 9,6 Кбит/с – и это не предел. Таким образом, требования к эффективным механизмам замера времени и контроля временных прерываний являются чрезвычайно высокими. Как видно из недавно опубликованного доклада ведущей компании по разработке электрооборудования для производителей возобновляемой энергии Freqcon, время реакции в 20 мкс при полной загруженности системы уже не является чем-то необычным.
Однако существует ряд проблем, затрудняющих решение задачи минимального использования ресурсов CPU при сохранении эффективности обработки процессов, в особенности на встроенных системах. Это связано, как правило, с особенностями механизма учета времени таймеров ядра ОС Linux. Хотя начиная с версии ядра Linux 2.6 произошли существенные изменения таймеров, промах ожидания по-прежнему составляет примерно 50 мкс. Поэтому часто для высокоточного сна используется метод ожидания в цикле на процессоре,

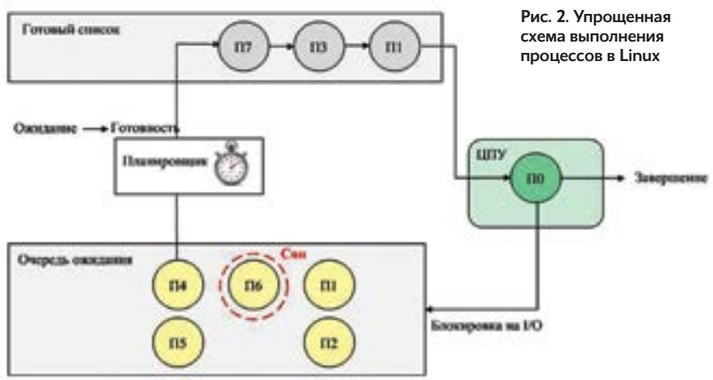


Рис. 2. Упрощенная схема выполнения процессов в Linux

Рис. 3. Промах системного сна

что, с точки зрения параллелизма, не является эффективным. В рамках библиотеки таймерной поддержки HighPerTimer предлагается новый подход оптимизации сна, который снижает использование ресурсов CPU до 1–1,89% относительно 100% ожидания в цикле на процессоре при сохранении минимальных потерь времени от точности пробуждения. Далее предлагается рассмотреть все существующие методы ожидания процесса на примере процессора Intel Core-i7.

Рис. 4. Промах ожидания в цикле на процессоре

Системный сон стандартной библиотеки языка Си

Стандартная библиотека языка Си ОС Linux предоставляет возможность приостановить выполнение процесса на заданное количество секунд, микросекунд или наносекунд. Объявленный в заголовочном файле unistd.h метод sleep() (подобно методам usleep() и nanosleep()) предоставляет простой способ заставить программу ждать заданный промежуток времени. На рис. 2 показана упрощенная схема исполнения системного ожидания. Очередь Run-time (также известная как Готовый список – Ready list) хранит список всех процессов, которые готовы к выполнению и не заблокированы на операциях I/O или других системных запросах. Записи в списке являются указателями на блок управления процессами, который хранит всю информацию о каждом состоянии. Когда выполнение какого-то процесса приостанавливается, сначала он попадает в очередь Run-time. После этого, в соответствии с его приоритетом, процесс попадает в очередь ожидания (Waiting queue), где ждет установленное время. После того как время сна истекло, процесс переходит в состояние готовности, попадает в список Ready list и завершает свое исполнение. Вся эта рутина контролируется планировщиком процессов.
Планировщик процессов является компонентом операционной системы и решает, какие задачи или процессы будут добавлены в очередь процессов, готовых к выполнению. Планировщик либо добавляет новый процесс в очередь готовых процессов, либо откладывает это действие. Планировщик также «будит» спящие процессы. Он вызывается операционной системой с частотой HZ, а это означает, что планировщик может проверить состояние процесса в очереди не чаще, чем каждые 1/HZ секунд. Таким образом, выполнение ожидания процесса происходит без использования ресурсов процессора, что является основным преимуществом системного сна. Однако промах сна сильно зависит от значения HZ и версии ядра. Под промахом имеется в виду отклонение или разница реального времени сна от ожидаемого. На рис. 3 показаны результаты промаха системного сна при задержке от 1 с до 1 мкс. Измерения проводятся на ядре с параметром HZ = 1000.

Сон в цикле ожидания на процессоре

Альтернативный способ исполнения сна – ожидание в цикле на процессоре (busy-waiting loop). При таком подходе процесс ожидает события, вращаясь все время в «плотном» цикле. Это позволяет сэкономить время и уменьшить промах, но недостаток данного подхода в том, что во время сна мы имеем 100% загрузку CPU.
В спецификации Intel рекомендуется использовать ассемблерную инструкцию pause. На более старых процессорах эта инструкция работает как nop. Инструкция nop не выполняет никаких действий и обычно используется вместе с инструкцией rep. Кроме того, например на ARM-архитектуре, не поддерживаются ни rep- ни pause-инструкции, поэтому в этом случае возможно использование только nop. Такой способ использует 100% производительности CPU, но в соответствии с рис. 4 имеет очень маленькое значение промаха. Измерения проводятся на ядре с HZ = 1000.
Следует также заметить, что значение промаха выравнивается после времени сна, равного 0,001953 с, и становится равным примерно 50 нс. Значения промаха здесь также зависят от величины 1/HZ. Когда время сна превышает параметр 1/HZ, значение промаха уменьшается прямо пропорционально к запланированному времени сна.
Когда время сна меньше, чем 1/HZ, величина промаха равна примерно 50 нс, что вполне приемлемо для высокоточного сна.

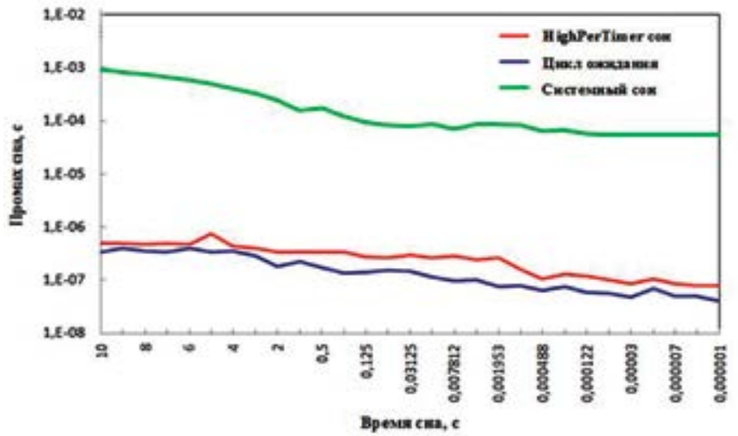


Рис. 5. Измерение зависимости значения промаха от времени сна

Таблица 2. Оценка величины промаха всех известных методов сна

		Системный сон	Цикл ожидания	HighPerTimer сон
Время сна ≥ 1/HZ	Мат. ожидание, мкс	61,9	0,160	0,258
Время сна < 1/HZ	Мат. ожидание, мкс	50,9	0,0701	0,0950
	Стан. отклонение, мкс	11,4	0,0400	0,0404

Таблица 3. Оценка загрузки процессора во время выполнения ожидания

	Системный сон	Цикл ожидания	HighPerTimer сон
Реальное время выполнения	835 мин 49.698 с	833 мин 18.146 с	833 мин 18.675 с
Время использования CPU	8.179 с	830 мин 38.679 с	15 мин 46.006 с
Процентное отношение	0.0160 %	99.7 %	1.89 %

Предлагаемый метод оптимизации ожиданий

HighPerTimer сон является новым комбинированным способом сна, который объединяет выполнение системного сна и ожидания в цикле на процессоре. При его разработке ставилась задача получить преимущества от использования каждого способа, т. е. минимальную загрузку процессора и минимальный промах. Основная идея заключается в разделении общего времени сна на две части, используя величину 1/HZ. Так как Linux не предлагает стандартизированного решения запроса значения HZ, используемого ядром, это значение приходится определять опосредованно. В качестве метода расчета HZ был выбран способ оценки загрузки процессора во время ожидания в цикле при едином компромиссном значении

сна для каждого вида платформ. Анализируя, как ядро замеряет время использования ресурсов процессора на разных платформах, были обнаружены зависимости, на основе которых рассчитывалась величина HZ.

На рис. 5 показаны результаты измерения значения промаха в зависимости от времени сна, которое варьируется от 10 с до 1 мкс. Когда время сна больше чем 1/HZ, наблюдается тенденция зависимости промаха от времени сна. Более того, подобная тенденция наблюдается как во время системного сна, так и во время ожидания в цикле на процессоре. В отличие от этого в интервале, когда время сна меньше, чем 1/HZ, промах остается нетеденциально константным и колеблется в диапазоне от 50 до 100 нс.

В таблице 2 показаны результаты измерений среднего значения промаха при каждом методе ожидания. Для измерения был использован цикл из 10 000 шагов, на каждом шаге которого были произведены замеры на время сна от 1 с до 1 мкс. Общее время теста составило 830 минуты.

В интервале, когда время сна больше, чем величина 1/HZ, в расчете стандартного отклонения нет смысла, т. к. промах зависит от времени сна тенденцией, проиллюстрированной на рис. 5. Когда время сна меньше, чем 1/HZ, значение промаха при выполнении сна HighPerTimer составляет в среднем 95 нс. Значение стандартного отклонения на этом интервале составляет около 404 нс. Кроме того, необходимо оценить HighPerTimer сон по параметру загрузки CPU. В таблице 3 представлены результаты процентного отношения использования процессорного времени в сравнении с системным сном и циклом ожидания.

Измерения проводились в цикле из 10 000 шагов, время сна на каждом шаге варьируется от 0,25 с до 1 мкс, при этом реальное время теста составило более 833 минут для каждого метода. Из таблицы заметно, что во время исполнения сна в цикле ожидания на процессоре загрузка CPU равна 99,7%. В сравнении с этим значением загрузка процессора в 1,89% при выполнении HighPerTimer сна выглядит существенным преимуществом. Следует отметить, что библиотека также обладает соответствующим механизмом прерывания этого спящего состояния.

Использование системного таймера ARM-процессора

Одним из последующих шагов является улучшенная поддержка системного таймера ARM-процессоров (одноядерных на базе семейств Cortex-A8 и двуядерных на базе Cortex-A9), на которых отсутствуют известные для ПК-платформ счетчики TSC и HPET. Согласно документации, реализация ARM должна поддерживать системный таймер GP Timer (Global Purpose Timer), отображенный в «закрытую» область памяти, значения которого могут быть считаны только из режима ядра. Для этого планируется создание собственного драйвера устройства и имплементация виртуального вызова mmap(). Это позволит отобразить физический адрес основного счетчика в программную область памяти. Т. е. доступ из области пользователя к аппаратному таймеру будет осуществляться через виртуальный адрес, возвращаемый вызовом mmap().

Промах системного сна на процессоре ARM Cortex-A8 сейчас варьируется от 125 до 435 мкс. Промах HighPerTimer ожидания при обращении к таймеру через системный вызов составляет примерно 3–9 мкс. Предположительно, прямое обращение к аппаратному таймеру может позволить сэкономить несколько дополнительных микросекунд и существенно улучшить производительность выполнения ожиданий.

От Motorola к анклавам

Пит Бекман – один из самых авторитетных представителей HPC-сообщества. Более 10 лет он работает в Аргонской Национальной Лаборатории, участвуя в крупнейших суперкомпьютерных проектах. Сейчас Пит возглавляет Exascale Technology and Computing Institute при Лаборатории и руководит группой разработчиков операционной системы будущего – Argo.

Редакция: Помните ли Вы, каким было отношение к параллельным вычислениям в начале вашей карьеры? Вы начинали сразу с параллельного программирования?

Пит Бекман: Я начинал аспирантом в Университете Индианы в 1985 году. В то время параллельные вычисления были экзотикой, о них говорили от силы на двух-трех семинарах по вычислительной технике. К счастью, на факультете появился Деннис Гэннон, а с ним и теория параллельных вычислений. Мы работали на системе BBN Butterfly GP-1000 с 32 процессорами Motorola 68020.

В те времена, когда первопроходцы исследовали параллельные алгоритмы теоретически, опыт написания и отладки таких программ был большой редкостью. Мы изобретали параллельные языки. Сам я работал над языком pC++, то есть параллельным C++. Проект финансировался DARPA.

С тех пор многое изменилось, но некоторые проблемы, с которыми мы тогда столкнулись, актуальны и теперь. Как реализовать параллельные алгоритмы и как эффективно исполнить реальный машинный код – над этим бьются и сейчас.

Р.: Есть ли у Вас чувство, что основные направления достижения масштабов экзаскейла уже более или менее определились и речь идет теперь о способах реализации похожих идей?

П. Б.: Я думаю, что сейчас со многими ключевыми вопросами основательно разобрались, теоретическая база и программные средства готовы к тому, чтобы двигать HPC вперед.

На заре параллельного программирования надо было быть одновременно и прикладным математиком, и системным программистом, и архитектором систем, и разрабатывать самим компиляторы, языки и средства разработки. Сейчас есть уже сложившиеся сообщества и финансируемые программы исследований в каждой из этих отраслей.

Конечно, эта ситуация порождает и новые проблемы. Многие программисты с трудом понимают, как на самом деле работает железо. В наш век виртуализации пропасть между абстрактными вычислениями и электроникой становится все шире.

В будущем наше промежуточное ПО и среда выполнения будут обладать способностями к интеллектуальной адаптации, к заполнению этой расширяющейся пропасти между программой и вычислительной платформой. В проекте Argo мы как раз создаем новую ОС и среду выполнения для масштабов экзаскейла, осваивая это промежуточное пространство.

Мы уже приближаемся к экзавычислениям, но я не думаю, что уже определены все трудности и понятно, что будет просто «куском кода, который надо запрограммировать», а что потребует глубоких исследований, которые, не исключено, будут длиться не один год. Два примера: модели программирования и отказоустойчивость. Последние несколько лет мы работали вместе со всеми основными производителями железа, исследуя, какими должны быть программы, чтобы они эффективно работали на их оборудовании, на их процессорах. Выяснилось, что производители не могут прийти к консенсусу по поводу общей для всех модели вычислений. Как же писать переносимый код, если нет единства даже в абстрактных принципах будущих параллельных систем (сейчас я не говорю о специфических программных моделях и языках программирования)?

Сегодня в качестве базовой абстракции вычислительной системы у нас есть: SPMD (принцип «одна программа – много данных»); память,

разделяемая узлами; передача сообщений между узлами. А что будет базовой абстракцией завтра? Этого мы не знаем. К тому же сейчас много дискуссий по поводу отказоустойчивости, а понимания того, насколько сложна эта проблема, нет. Достаточно ли будет просто расширить возможности существующей модели, или придется полностью менять сами основы научных алгоритмов?

Я думаю, что, пока мы не в состоянии оценить уровень сложности проблем, эти области будут широко открыты для исследований. Но и с этими неопределенностями вполне можно двигаться вперед, вместе определять сценарии отказов, договариваться об интерфейсах для оповещения об ошибках. Сейчас не существует стандарта, по которому HPC-приложение получает оповещение о сбое системы, и нет стандартной схемы определения типов отказов. Это та область, где кооперация может способствовать быстрому прогрессу.

Р.: Какие концепции новой ОС проекта Argo наиболее революционны? Что такое «анклавы» (enclaves) и «объединительная панель» (backplane)?

П. Б.: Мы начали с того, что определили новую абстрактную машинную модель для потоков, доменов когерентной памяти и доступа к удаленной памяти load/store. Когда мы приступили к проектированию ОС и среды выполнения для экзамасштабов, то скоро поняли, что для нашей архитектуры нужен фундамент. Поверх машинной модели мы построили концепцию параллельных анклавов. Эта концепция напоминает принцип разбиения на партиции на современных машинах, но с новой функциональностью и понятным управлением интерфейсами.

Сейчас пользователям просто предоставляют набор узлов, соединенных по MPI. Когда мы стали разбираться с управлением питанием и с отказоустойчивостью, то обнаружили, что нужна более формальная модель, которая будет поддерживать управление энергопотреблением, балансировку нагрузки и реакцию на отказ на уровне отдельного задания. Допустим, приложение, работающее на нескольких узлах, управляет энергопотреблением и реагирует на системные ошибки. В существующих моделях вычислений пользователю придется сначала вычленить такую функциональность из партиции, получив к ней доступ через планировщика заданий, и после этого попытаться запустить эти функции, каждая из которых будет невидима для более общего системного ПО.

В нашей модели с анклавами все эти механизмы видны, вся система имеет возможность контролировать потребление энергии, получать сообщения о сбоях и взаимодействовать с приложениями. Объединительная панель – коммуникационный слой в парадигме publish/subscribe, поддерживающий эти взаимодействия. Без этой объединительной панели у глобальной системы не было бы возможности отправить информацию о сбое или данные о балансировке нагрузки в анклавы и дальше в отдельные вычислительные узлы. К тому же новая модель дает анклавам возможность динамически реагировать на запросы системы на реконфигурацию или корректировку ресурсов.

Р.: Какие системные функции останутся за ОС в эру экзаскейла, какие перейдут к среде выполнения, какие к приложениям? Какая часть интеллектуальных операций будет делаться на аппаратном уровне?

П. Б.: В нашем проекте Argo мы очень тесно связали то, что можно назвать низкоуровневой операционной системой, со средой выполнения. Для себя мы вообще называем это OSR, то есть OS – операционная система – плюс среда выполнения. В прошлом наши единицы согласованности был тяжеловесными – системные процессы и потоки ядра. Теперь, когда мы думаем об экзамасштабах, параллелизм внутри узла стремительно увеличивается. Поскольку основная ОС необходима для того, чтобы обрабатывать входящие сообщения и управлять системной памятью, уже нет практического смысла сохранять слои легковесных потоков пользовательского уровня, которые создавались для массовой согласованности в отдельном и отдаленном программном интерфейсе. Поэтому в нашей системе интерфейсы сверхлегковесных потоков тесно связаны с нижними уровнями ОС. И мы ожидаем, что в будущем аппаратный уровень поддержит активацию чрезвычайно легких потоков – примерно как это сделано с устройствами «бодрствования» (wake-on) в суперкомпьютерах IBM BlueGene/Q.

Р.: Возникает вопрос: что будут представлять собой ОС в будущем? Не останется ли название «операционная система» просто для удобства, являясь на самом деле чем-то существенно другим по отношению к тому, с чем мы имели дело раньше?

П. Б.: В операционных системах мы видим два серьезных изменения: во-первых, тесная связь низкоуровневой ОС и среды выполнения. Во-вторых, движение к тому, что я назвал OSR. Сейчас, когда люди говорят об OSR, они в основном имеют в виду то, что касается кода, исполняемого на узле. Или просто узла OSR, который по каким-то причинам связан с другими узлами. В старой модели вычислений HPC-система строится из узлов OSR как самостоятельных единиц, которые планировщик запускает, определяя их взаимодействие. Но мы считаем, что для того, чтобы управлять энергопотреблением, обеспечить отказоустойчивость и динамическое исполнение, нужно, чтобы была доступна общая картина того, что происходит в системе.

Вот поэтому анклавы и объединительная панель так важны для нашего проекта. На верхнем уровне системы энергопотребление можно будет оптимизировать по всем анклавам примерно так же, как энергопотребление оптимизируется внутри анклава по узлам, в него входящим. Рекурсия и иерархия – очень важные концепты масштабирования для стабильных систем, так что мы думаем, что системы в эру экзаскейла будут не просто наборами легковесных ядер. Но хотя наш проект ориентирован на экза-масштабы, мы верим, что наш подход будет полезен и при работе на классических Linux-кластерах и станет ориентиром для переносимых приложений, которые не будут сводиться лишь к классическому MPI-программированию.

Р.: Что Вы думаете о подходе к системному ПО группы, возглавляемой Томасом Стерлингом?

П. Б.: Томас всегда был первопроходцем – и тогда, когда строил простые Linux-кластеры, и сейчас, когда разрабатывает модели исполнения. Мы согласны с ним в том, что по мере того, как масштабы и сложность систем будет расти, среда выполнения будет более динамичной и реактивной. Программисты будут разрабатывать алгоритмы, которые предполагают слои среды выполнения со сложной функциональностью, поддерживающие выполнение потока заданий.

Р.: Можете ли вы отметить важные, революционные идеи и разработки в Европе, Японии и Китае?

П. Б.: Конечно! В Европе удалось создать несколько команд, которые исследуют, как нужно писать приложения для вычислений экзамасштаба. Работы по OpenACC и OpenMP очень помогли мировому сообществу разработчиков. Есть очень интересные проекты, исследующие HPC-системы на процессорах с низким энергопотреблением, таких как ARM. Я бы очень хотел получить доступ к прототипам таких систем, это бы принесло пользу проекту Argo.

В Японии команды, ведущие академические исследования, работают в тесном контакте с их разработчиками чипов, чтобы исследовать новые сверхбольшие системы. Fujitsu K Computer – замечательное достижение, я думаю, что Япония будет лидером в разработке архитектур экзасистем.

Р.: Могли бы Вы назвать мощные и успешные интернациональные проекты эры экзаскейла?

П. Б.: Я думаю, что экзамасштабные проекты стран «восьмерки» – это потрясающие примеры совместной работы. Я представляю, какой бы эффект был, если бы политики и фонды смогли распространить такую модель более широко. Что касается нашей группы, то мы в своей работе активно сотрудничаем с учеными из Японии, Франции, Китая. У нас постоянный поток студентов и аспирантов из этих стран. Они приезжают к нам в Аргоннскую Национальную Лабораторию и работают вместе с нами. Хороший пример – проект MPICH, но и над файловой системой, над вводом-выводом, над ZeptoOS (нашей предыдущей ОС), как и над Argo, работа происходит в том же интернациональном духе.

Р.: Сейчас все говорят о перспективах концепции Co-Design. Вы видите в ней что-то новое? И что в ней наиболее важно?

П. Б.: Во времена, когда Сеймур Крей разрабатывал машины, системные архитектуры были сразу разработчиками и чипов, и параллельных программ. Современные архитектуры, если считать, что в них входят и компиляторы, и слои обмена сообщениями, и приложения, из-за их сложности невозможно уложить в одной голове, даже гениального архитектора. Сейчас, как я говорил, есть несколько команд, которые успешно работают в самых разных областях. Главное в Co-design – отдавать себе отчет в том, что нужны какие-то организационные усилия, чтобы разработки в области системного ПО, аппаратной архитектуры и прикладных программ соответствовали друг другу.

Если разработчики чипов будут просто «выпекать» новые процессоры со все большим количеством ядер и все меньшей скоростью доступа к памяти, то ничего не получится. Нужен структурированный диалог между разработчиками архитектуры чипов, учеными, придумывающими новые алгоритмы, и программистами, исследующими новое низкоуровневое системное ПО. Пример из нашего опыта в Argo – механизмы активации легковесных потоков. Разработчики чипов, которые думают прежде всего о флопсах и низком энергопотреблении, никогда бы сами не стали добавлять функциональность координации легковесных потоков, если бы не услышали об этом от разработчиков OSR и прикладных математиков.

Р.: Вы принимаете участие во множестве интернациональных инициатив, таких как IESP, BDEC, EESI и другие. Какой из результатов их деятельности самый важный?

П. Б.: Эти проекты породили много интернациональных проектов по всему миру. Их координация оказала немалое влияние на направление деятельности разработчиков. Работа во взаимодействии с самыми выдающимися учеными, из какой бы страны они ни были, очень ускоряет решение сложных проблем. Например, на этой неделе мы работаем над загрузкой новых легковесных ядер, разработанных в Японии, чтобы посмотреть, смогут ли они стать основой некоторых наших разработок. Мы хотим, чтобы научная часть сверхбольших систем продвигалась вперед как можно быстрей. Миру нужно решать важные научные проблемы: новые ресурсы энергетики, не загрязняющей окружающую среду, проблемы здоровья, надо улучшить методы предсказания природных катастроф. Я верю, что это становится возможным в том числе благодаря усилиям IESP, BDEC и EESI.

Кристалл для «Ангары»

Текст И. Жабин,
Д. Макагон,
А. Симонов,
Е. Сыромятников,
А. Фролов,
А. Щербак

В октябре компания «НИЦЭВТ» представила СБИС EC8430 маршрутизатора отечественной высокоскоростной сети «Ангара» для кластеров и суперкомпьютерных комплексов. Что представляет собой сеть «Ангара» и разработанная СБИС? На этот вопрос мы ответим в данной статье.

Сотни тысяч вычислительных ядер в суперкомпьютерах – это уже реальность. Очевидно, что от того, насколько эффективно обеспечивается обмен данными между ядрами, в конечном итоге и зависит, насколько эффективно их можно использовать одновременно для решения одной задачи. В этом смысле коммуникационная сеть является ключевым компонентом суперкомпьютера. Медленная сеть, не способная эффективно подстраиваться под возникающие отказы оборудования, которые при наличии миллионов компонентов перестают быть маловероятными, становится «узким местом», что для многих задач с большой долей сетевых обменов может быть весьма критичным. Фактически мы говорим о потенциале масштабируемости, который напрямую определяется именно характеристиками используемой коммуникационной сети.

Современные высокоскоростные сети

Прежде чем перейти к предметному обсуждению, необходимо определиться с терминологией. Коммуникационная сеть (интерконнект) состоит из узлов, в каждом из которых есть сетевой адаптер, соединённый с одним или несколькими маршрутизаторами, которые, в свою очередь, соединяются между собой высокоскоростными каналами связи (линками). Структура сети, определяющая, как именно связаны между собой узлы системы, задается топологией сети. В настоящее время наиболее распространены топологии «многомерный тор», fat tree, dragonfly. Архитектура маршрутизатора определяет структуру и функциональность блоков, отвечающих за передачу данных между узлами сети, а также необходимые свойства протоколов канального, сетевого и транспортного уровней, включая алгоритмы маршрутизации, арбитража и управления потоком данных. Архитектура сетевого адаптера определяет структуру и функциональность блоков, отвечающих за взаимодействие с хост-системой: процессором и памятью. На этом уровне, в частности, может осуществляться поддержка операций библиотеки MPI, обработка исключительных ситуаций, агрегация пакетов, поддержка механизма RDMA (Remote Direct Memory Access), обеспечивающего прямой доступ к памяти другого узла без участия его процессора.
Если посмотреть на статистику списка TOP500 (top500.org), то можно заметить, что большинство представленных в нем систем используют коммерчески доступные сети InfiniBand и Gigabit Ethernet. Однако суперкомпьютеры из первой десятки списка – китайские системы

Tianhe-2 и Tianhe-1A, японский K Computer, американские Cray Titan, IBM Blue Gene/Q – используют собственные уникальные («заказные») коммуникационные сети, разрабатываемые в составе этих вычислительных систем и доступные только совместно с ними. То есть, хотя в отличие от коммерчески доступных сетей, «заказные» сети занимают гораздо меньшую долю рынка, именно они используются в наиболее мощных суперкомпьютерах. Это, конечно же, неслучайно. Основная причина здесь в том, что для получения высокой производительности, сеть должна быть максимально интегрирована с вычислительной подсистемой и программным обеспечением (ОС на вычислительных узлах, библиотеки параллельного программирования, средства мониторинга и управления ресурсами системы). Более того, часто «заказные» системы (и их сети) подстраиваются под определённые классы целевых задач и, соответственно, предполагают необходимость тесного сотрудничества с организациями, решающими задачи в интересах государства.
Приобретение подобных машин в России в ряде случаев затруднено, а часто является просто невозможным. В то же время коммерчески доступные сети InfiniBand и Ethernet далеко не всегда подходят для эффективной реализации систем со столь высокими требованиями по масштабируемости, надежности и производительности. Можно также заметить, что в списке TOP500 нет сетей, использующих ПЛИС (FPGA). В связи с этим крайне актуальным является вопрос разработки отечественной высокоскоростной сети, сравнимой с западными «заказными» аналогами.
В таблице приведены основные характеристики сетей, используемых в наиболее мощных суперкомпьютерах, а также для сравнения – характеристики сети «Ангара» на базе ПЛИС и СБИС.
Необходимо отметить, что ряд отечественных организаций также достиг определенных успехов в разработке коммуникационных сетей для суперкомпьютеров, в том числе РФЯЦ ВНИИЭФ, ИПС РАН, ИПМ РАН и НИИ «Квант».

Проект разработки высокоскоростной сети «Ангара»

В сети «Ангара» маршрутизатор и адаптер находятся в одном кристалле (в отличие, например, от сети InfiniBand). Упрощенная блок-схема показана на рис. 1.
Топология сети – «многомерный тор» (до 4 измерений). Поддерживается надёжная передача пакетов по линку, детерминированная и адаптивная маршрутизация. Для предотвращения взаимных блокировок (deadlocks) детерминированной маршрутизации используется комбинация двух методов: метода «порядка направлений» (direction order) и «правило пузырька» (bubble-rule). Поддерживаются три RDMA-операции: асинхронные записи в память удаленного узла, асинхронные чтения и атомарные операции с удаленной памятью. Отдельный виртуальный канал используется для доставки ответов на чтения, чтобы предотвратить возникновение логических взаимных блокировок, обусловленных взаимозависимостью запросов и ответов. Эффективная работа с сетевым адаптером многоядерных процессоров поддерживается с помощью нескольких инжекционных конвейеров.
Взаимодействие вычислительного узла (т. е. кода, исполняемого на центральном процессоре) с маршрутизатором осуществляется путем записи данных по адресам памяти,

ОСНОВНЫЕ ХАРАКТЕРИСТИКИ СОВРЕМЕННЫХ «ЗАКАЗНЫХ» КОММУНИКАЦИОННЫХ СЕТЕЙ И СЕТИ «АНГАРА»

СЕТЬ (СУПЕРКОМПЬЮТЕР)	TH Express-2 (Tianhe-2)	Cray Gemini (Titan)	IBM BlueGene/Q (Sequoia)	Tofu (K Computer)	InfiniBand FDR (Stampede)	Cray Aries (Cray XC30)	Ангара (ПЛИС)	Ангара (СБИС)
ГОД СОЗДАНИЯ СЕТИ	2013	2010	2011	2011	2011	2012	2010	2013
ТОПОЛОГИЯ	fat tree	3D-тор	5D-тор	6D-тор	fat tree	dragonfly	2D-тор	4D-тор
ПС ИНТЕРФЕЙСА С ХОСТ-СИСТЕМОЙ, ГБ/С	8 PCIe 2.0 x16	9,6 HyperTransport 3	~20 Custom	6,25 Custom	16 PCIe 3.0 x16	16 PCIe 3.0 x16	2 PCIe 1.0 x8	8 PCIe 2.0 x16
ПС ЛИНКА, ГБ/С	~4,55	9,375	2	5	6,8	5,25	0,625	7,5
ЗАДЕРЖКА МЕЖДУ СОСЕДНИМИ УЗЛАМИ, МКС	н/д	1,4	< 1,0	< 1,0	1,0	< 1,0	2,5	1,0
ТЕХПРОЦЕСС КРИСТАЛЛОВ СЕТ. АДАПТЕРОВ	90 nm	90 nm	45 nm Integrated into Compute chip	65 nm	65 nm	40 nm	(65 nm) FPGA	65 nm

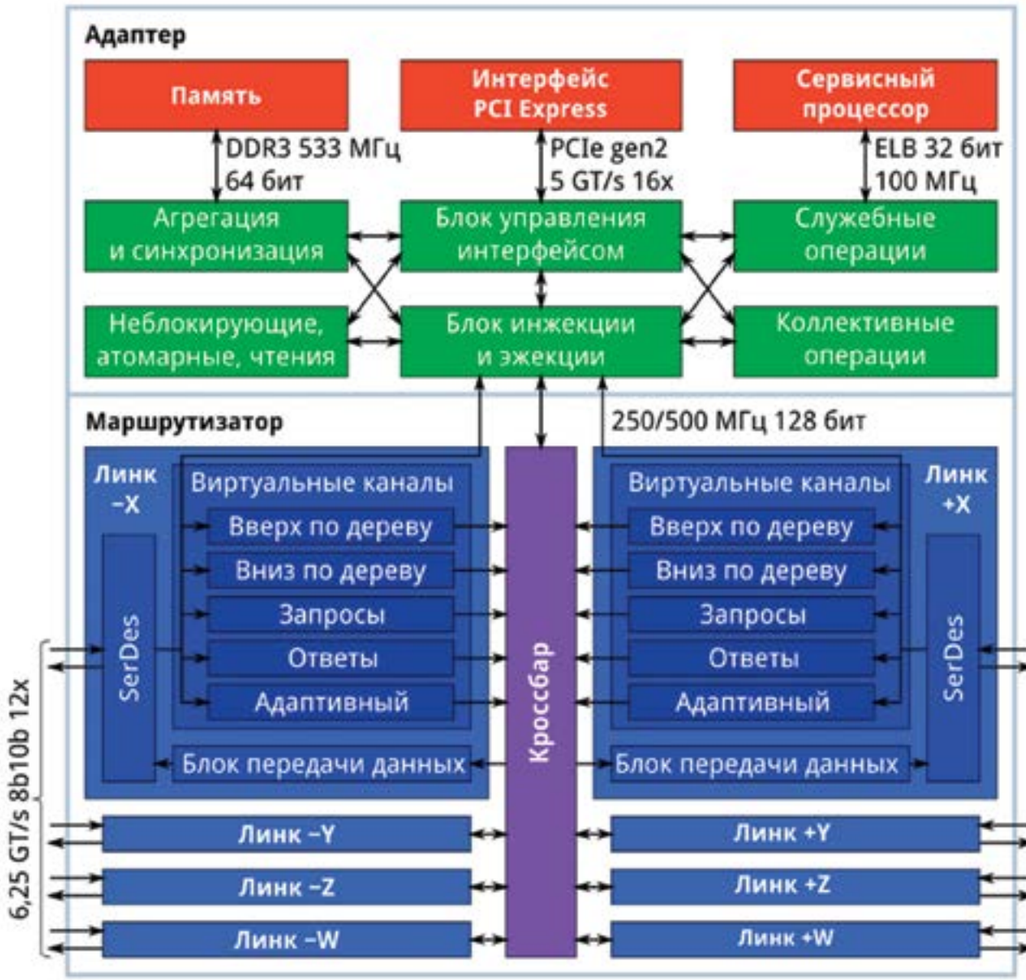


Рис. 1. Упрощенная блок-схема СБИС EC8430

которые отображены на адреса ресурсных регионов маршрутизатора (memory-mapped input/output). Это позволяет приложению взаимодействовать с маршрутизатором без участия ядра ОС, что снижает накладные расходы при отправке пакетов, поскольку переключение в контекст ядра и обратно занимает существенное время, в сравнении со временем отправки пакета. Аппаратная поддержка коллективных операций (например, broadcast – один узел рассылает данные группе узлов) реализуется на базе основной сети с топологией «многомерный тор», при этом используются отдельные виртуальные каналы, образующие виртуальную подсеть с древовидной топологией. В дереве задается корень, относительно которого вводятся два возможных направления движения по дереву: от корня и к корню. Каждому из направлений соответствует свой виртуальный канал. Чтобы предотвратить появление взаимных блокировок, дерево строится с учетом порядка измерений (dimension order). Для достижения большей эффективности было принято решение исключить из рассмотрения случаи, когда две разные задачи используют пересекающиеся группы узлов. С учетом этого каждый узел может относиться только к одной вычислительной задаче. Это позволяет исключить накладные расходы, связанные с использованием виртуальной памяти, избежать интерференции задач, упростить архитектуру маршрутизатора за счет отсутствия необходимости в полноценном MMU и избежать всех связанных с его работой коммуникационных задержек, упростить модель безопасности сети, исключив из нее обеспечение безопасности процессов различных задач на одном узле. Принятое решение не повлияло на функциональность сети, поскольку она предназначена в первую очередь для задач большого размера. Аналогичное решение было принято в IBM Blue Gene, с той разницей, что там ограничение на единственность задачи вводится для раздела. Основным режимом программирования для сети «Ангара» является совместное использование MPI, OpenMP и Shmem. Также поддерживаются библиотеки и языки параллельного программирования GASNet, UPC, ARMCI, Charm++.

Для обеспечения эффективного ввода-вывода используется параллельная файловая система Lustre. Во время подготовки к выпуску СБИС были добавлены функции для поддержки мониторинга, отладки и профилирования. В числе прочего был проработан механизм генерации прерываний для оповещения хост-системы о возникновении той или иной нештатной ситуации, например, об отказе какого-либо блока адаптера или о получении некорректного пакета из сети. Также были добавлены несколько сотен счетчиков производительности и подсистема конфигурирования отдельных блоков.

СБИС EC8430

Сеть «Ангара» — первый в России проект высокоскоростной сети с маршрутизаторами на основе СБИС отечественной разработки.

Микросхема EC8430 стала итогом семилетней работы подразделения ОАО «НИЦЭВТ» – разработчика высокоскоростной сети «Ангара». СБИС выпущена на фабрике TSMC с использованием технологии 65 нм. Размер кристалла – 13,0×10,5 мм, количество транзисторов – 180 миллионов. Кристалл размещен в корпусе flip-chip BGA, имеет 1521 вывод, размер подложки – 40×40 мм. СБИС работает на частоте 250/500 МГц и потребляет 36 Вт. Поддерживается топология сети «четырёхмерный тор», каждый сетевой узел может иметь до 8 соединений с соседними узлами, пропускная способность каждого соединения – 75 Гбит/с (12 линий по 6.25 Гбит/с, кодирование 8b10b). Взаимодействие с вычислительным узлом осуществляется через интерфейс PCI Express 2.0×16.

Размещение функциональных блоков в СБИС показано на рис. 2. Выделены следующие блоки:

- **PCIe** — *блок интерфейса PCI Express 2.0 (16x5 бит/с);*
- **BUI** — *блок приема и передачи данных через PCIe;*
- **PE** *и* **NI** — *блоки инжекции и эжекции сетевых пакетов;*
- **CROSSBAR** — *блок коммутации для передачи пакетов между линками и блоками инжекции и эжекции;*
- **LINK[0-7]** *и* **LINK[0-7] SerDes** — *блоки приема и передачи данных через линки в соседние узлы (8 линков по 12x6,25 Гбит/с);*
- **DDR3** — *интерфейс к DRAM- памяти (ширина 64+8 ECC = 72 бита, частота 533 Мгц (1066 MT/с));*
- **MIDP** — *блок приема и передачи данных через DDR3;*
- **GPIO** — *служебные интерфейсы конфигурирации (SPI Flash), отладки (JTAG), подключения сервисного процессора (SBUS);*
- **PLL** — *блок фазовой автоподстройки частоты.*

СБИС EC8430 используется в сетевых адаптерах «Ангара» в формфакторе плат расширения PCI Express для кластерных систем с коммерчески доступными процессорами (рис. 3). В настоящее время ведутся работы по интеграции СБИС в разрабатываемую в ОАО «НИЦЭВТ» вычислительную платформу для отечественного суперкомпьютерного комплекса «Ангара». Работы по созданию суперкомпьютера «Ангара», включая разработку СБИС EC8430, выполняются при финансовой поддержке Министерства промышленности и торговли РФ.

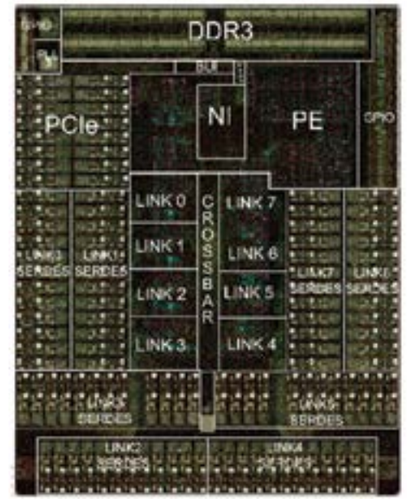


Рис. 2. Размещение функциональных блоков в СБИС EC8430

Рис. 3. Сетевой адаптер сети «Ангара» на базе СБИС EC8430

Высокопроизводительные вычисления в Desktop Grid

Текст Евгений Ивашко

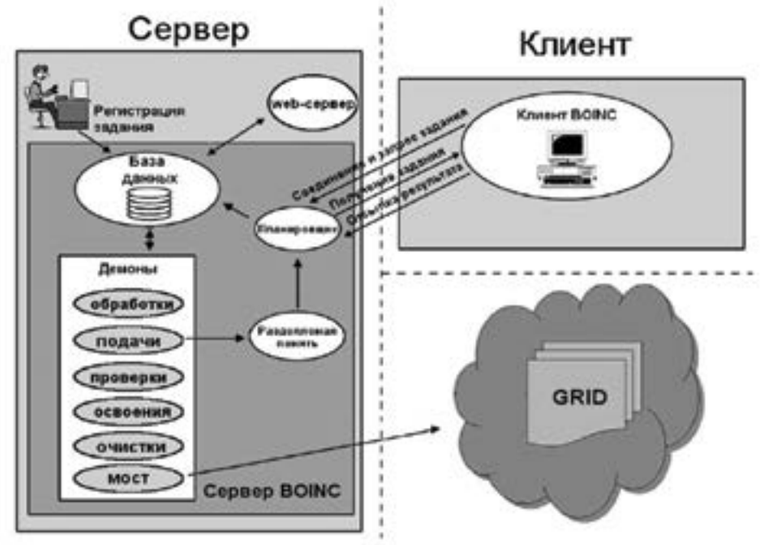
Высокопроизводительные вычисления играют большую роль при проведении научных исследований, разработке новых видов промышленной продукции и в социальной сфере. Они востребованы как инструмент для решения задач квантовой химии, молекулярной биологии, гидродинамики и прочих областей наук, требующих проведения большого количества ресурсоемких вычислительных экспериментов и обработки больших объемов данных. Без высокопроизводительных вычислений не обходится ни одна современная технологичная новинка или масштабное событие в жизни страны – будь то КаМАЗ и Sukhoj SuperJet 100, Олимпиада-2014 в Сочи, Бурейская и Зейская ГЭС или новые лекарственные препараты...

Как правило, высокопроизводительные вычисления ассоциируются с суперкомпьютерами. Действительно, большое количество как проприетарных, так и свободно распространяемых расчетных программ создается для суперкомпьютеров. Однако далеко не все задачи требуют использования такого дорогостоящего и специализированного оборудования. Моделирование методом Монте-Карло, поиск решения в конечном пространстве кандидатов, случайный поиск, имитационное моделирование, анализ и обработка независимых наборов данных – как правило, эти задачи не задействуют основное достоинство вычислительных кластеров – сверхбыстрый интерконнект, связывающий множество вычислительных узлов в логически единую систему.

Desktop Grid и BOINC

Большой вычислительный потенциал кроется в персональных компьютерах, имеющихся «под рукой» практически у каждого: дома, на работе, в интернет-кафе. В мире насчитывается более миллиарда компьютеров, которые, даже по самым скромным подсчетам, обладают суммарной пиковой производительностью более 120 Экзафлопс (для сравнения: суммарная пиковая производительность всех суперкомпьютеров списка TOP500 на июнь 2013 года составляет «лишь» 325.8 Петафлопс). При этом в среднем персональный компьютер загружен лишь на 10–15% своей мощности, а значит, простаивающие ресурсы можно использовать для научных расчетов.
Именно эта идея легла в основу технологий Desktop Grid: географически удаленные друг от друга неспециализированные вычислители (персональные компьютеры, ноутбуки и даже смартфоны) с помощью особого программного обеспечения объединяются в единую вычислительную сеть. Примеров такому промежуточному программному обеспечению достаточно много: Condor, X-Com, OurGrid, XtremeWebHEP, BOINC и другие.
Добровольные вычисления (Volunteer Computing) – наиболее яркий пример использования простаивающих ресурсов персональных компьютеров в интересах науки. В этом относительно новом для распределенных вычислений направлении масштабные расчеты в течение долгого периода времени (месяцы, годы) проводятся «волонтерами» (volunteers). Волонтеры, или добровольцы, – это обычные люди, живущие по всему миру и предоставляющие собственные компьютеры для решения разнообразных научных задач. В расчетах используются только свободные ресурсы (когда компьютер простаивает), поэтому участие в добровольных вычислениях не мешает работе пользователя.

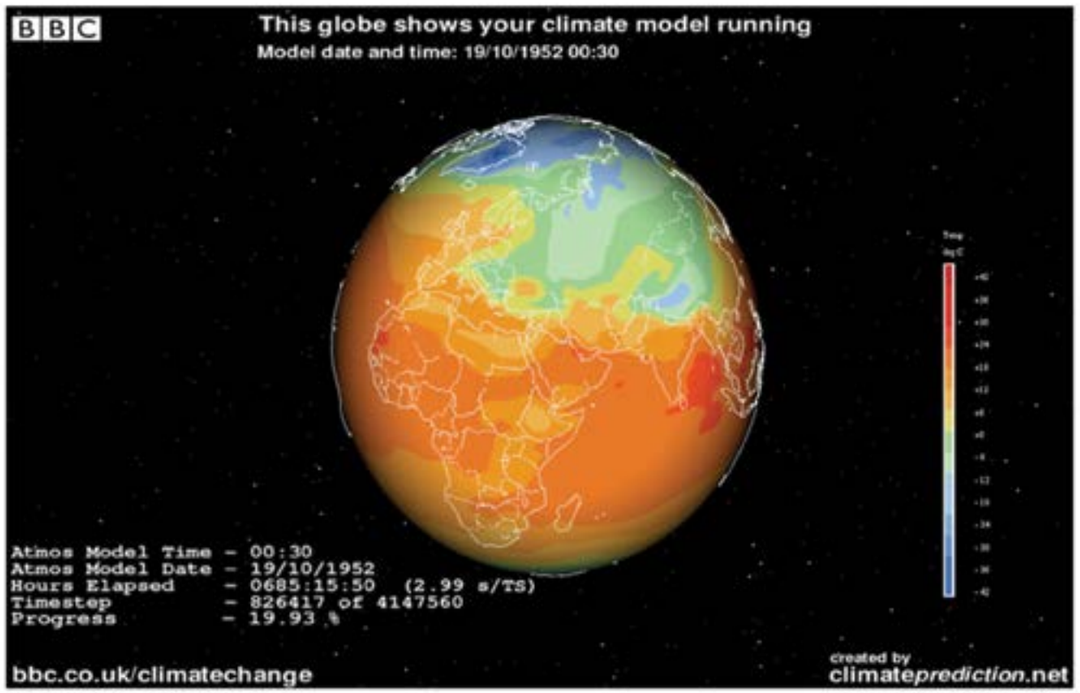
Наиболее популярной платформой организации добровольных вычислений является BOINC (Berkeley Open Infrastructure for Network Computing). Из порядка 100 крупных действующих проектов добровольных вычислений около 80 основаны на BOINC, т. е. данная платформа дефакто является стандартом для проектов добровольных вычислений.
Ярким примером BOINC-проекта в сети Интернет, использующего свободные вычислительные ресурсы компьютеров добровольцев, является проект SETI@home, цель которого – анализ космических радиосигналов. Средняя производительность вычислений этого проекта по состоянию на 31 октября 2013 года составляет 1.7 Петафлопс. Аналогичные показатели пиковой производительности имеют суперкомпьютеры, находящиеся во второй десятке списка TOP-500 (по состоянию на июнь 2013 года). При этом в проекте SETI@home участвуют более миллиона пользователей, а виртуальная вычислительная система состоит из 3.5 млн узлов.
Платформа BOINC является активно развивающимся открытым (Open Source) программным обеспечением и предоставляет богатую функциональность. При этом она отличается простотой в установке, настройке и администрировании, а также обладает хорошими возможностями по масштабируемости, простоте подключения новых вычислительных узлов, использованию дополнительного ПО, интеграции с вычислительными кластерами, другими грид-системами и т. д.
Платформа BOINC имеет архитектуру «клиент-сервер», при этом клиентская часть может работать на произвольном количестве компьютеров с различными аппаратными характеристиками и набором программного обеспечения. Сервер поддерживает одновременную работу большого числа независимых проектов; каждый клиент системы может одновременно производить вычисления для нескольких BOINC-проектов. BOINC предоставляет возможность гибкой настройки клиентской части, регулируя максимальный размер загружаемых файлов, время выполнения рабочих заданий, загрузку CPU или GPU, выделяемый объем оперативной памяти и дискового пространства.
Рабочий процесс в грид-системе, основанной на платформе BOINC, организован следующим образом. Вычислительные узлы, имеющие свободные ресурсы, обращаются к серверу для получения новых рабочих подзаданий в рамках проекта. Сервер BOINC рассылает клиентским приложениям экземпляры рабочих подзаданий, клиенты выполняют расчеты и отсылают обратно результаты. После получения результатов сервер проверяет и обрабатывает их, например, занося в базу данных или автоматически создавая на их основе новые рабочие подзадания.



Архитектура BOINC

BOINC: проекты и достижения

Стремительное развитие каналов связи (и соответствующее увеличение покрытия, скорости и пропускной способности) сети Интернет, а также рост производительности персональных компьютеров делают Desktop Grid все более популярными и перспективными. Вычислительноемкие расчеты с использованием этих технологий составляют важную часть отрасли высокопроизводительных вычислений. В рамках Седьмой рамочной программы ЕС (FP7) получил финансирование проект DEGISCO по развитию технологий Desktop Grid, была создана International Desktop Grid Federation (IDGF), крупнейшие исследовательские проекты на базе Desktop Grid выполняют ведущие мировые университеты (SETI@HOME, Rosetta@Home, ClimatePrediction@Home и другие).
Многие исследования, выполняемые с помощью добровольных



Climateprediction.net: моделирование температурных изменений на планете

вычислений, уже добились значительных результатов. Например, в рамках проекта Einstein@home на основе анализа космических радиосигналов обнаружено несколько десятков радиопульсаров. Проектом PrimeGrid@home поставлен целый ряд рекордов по нахождению простых чисел особого вида. Расчеты проекта LHC@home помогают стабилизировать пучок частиц в экспериментах на Большом адронном коллайдере.

Существует большое количество проектов в самых разных областях научных исследований – анализ структуры белков в биоинформатике, предсказание климата, материаловедение, ядерная физика, теория чисел и многие другие. Среди них отдельно стоит упомянуть российский проект SAT@home, созданный совместно сотрудниками Института динамики систем и теории управления Сибирского отделения РАН (г. Иркутск) и Института проблем передачи информации РАН (г. Москва). Проект нацелен на решение задач выполнимости булевых формул (SAT).

Платформа BOINC является активно развивающимся открытым (Open Source) программным обеспечением и предоставляет богатую функциональность. При этом она отличается простотой в установке, настройке и администрировании, а также обладает хорошими возможностями по масштабируемости, простоте подключения новых вычислительных узлов, использованию дополнительного ПО, интеграции с вычислительными кластерами, другими грид-системами и т. д.

Развитием технологий Desktop Grid на международном уровне занимается IDGF. Эта организация объединяет экспертов из всех областей, связанных с Desktop Grid. На сайте организации размещены различные технические и обучающие материалы, материалы конференций, а также новости и информация о прошедших и будущих конференциях. Ключевой конференцией в мире, пожалуй, является BOINC Workshop, на которой ежегодно собираются ученые и представители крупных проектов со всего мира, включая основателя и руководителя проекта BOINC Дэвида Андерсона (David Anderson). Российское отделение IDGF, как нетрудно заметить по сайту организации, является одним из наиболее активных.

BOINC:FAST – первая российская конференция по BOINC

В сентябре этого года недалеко от г. Петрозаводска, на берегу Онежского озера была проведена первая специализированная российская конференция «BOINC:FAST 2013. Высокопроизводительные вычисления на базе BOINC: фундаментальные исследования и разработки». Организаторами конференции выступили Институт прикладных математических исследований (ИПМИ) Карельского научного центра РАН, Институт проблем передачи информации РАН, Петрозаводский государственный университет и Центр высокопроизводительной обработки данных Карельского научного центра РАН.

География участников широка – Иркутск и Пенза, Москва, Курск и Петрозаводск, доклад представил аспирант из Китая Тянь Бо, проходящий обучение в МГУ, для участия в конференции в Петрозаводск вернулась аспирантка ИПМИ Наталия Никитина, проходящая стажировку в г. Любек (Германия). Специально для конференции записал свой видеодоклад руководитель и главный разработчик проекта BOINC Дэвид Андерсон (США), в котором он представил современное состояние отрасли и планы на будущее. В режиме телеконференции выступил председатель IDGF Роберт Ловас (Венгрия), рассказавший о деятельности IDGF. Не менее интересны были и доклады, рассказывающие об опыте реализации BOINC-проектов, использовании BOINC-грид в рамках одной организации, интеграции BOINC с вычислительными кластерами, повышении производительности проектов на основе специальных математических моделей, обработке больших наборов данных в BOINC-грид, моделировании телекоммуникационных сетей и многие другие. Подробнее с докладами конференции можно ознакомиться на сайте конференции www.boincfast.ru.

Участники конференции наслаждались прекрасной погодой и живописной карельской природой

Не остался в стороне от конференции и журнал «Суперкомпьютеры»: каждый участник получил по два номера журнала среди раздаточных материалов. Финансовую поддержку предоставил Российский фонд фундаментальных исследований и Петрозаводский государственный университет (Программа стратегического развития). К конференции было приурочено и объявление о начале сотрудничества с мировым лидером GP-GPU – компанией NVIDIA, которая безвозмездно предоставила свои видеокарты и учебно-методическую поддержку для организации в Петрозаводском государственном университете учебных курсов по программированию на языке CUDA.

В рамках конференции был организован круглый стол «Проекты добровольных вычислений: диалог разработчиков и участников», в котором приняли участие как исследователи, разработчики проектов, так и представитель информационного портала BOINC.ru — крупнейшего российского ресурса, объединяющего русскоговорящих добровольцев, участвующих в различных BOINC-проектах. На конференции также прошло заседание российского отделения International Desktop Grid Federation, на котором участники обсудили приоритетные направления сотрудничества, в том числе международного, меры по развитию и повышению популярности Desktop Grid в России, создание новых проектов и привлечение пользователей и ученых к использованию Desktop Grid.

Следующая встреча исследователей, где будут широко обсуждаться вопросы развития Desktop Grid, – Национальный суперкомпьютерный форум, который пройдет в конце ноября в г. Переславле-Залесском.

Суперкомпьютеры и мультиагентные технологии

для решения сложных задач управления ресурсами в реальном времени

Текст О. Граничин, П. Скобелев
Иллюстрация Владимир Камаев

Рис. 1. Картина известного Европейского художника Яка Депчика, иллюстрирующая наше интуитивное понимание сложности окружающего мира как самоорганизующегося множества разнообразных организмов

На пороге века сложности

По мнению выдающегося ученого современности, физика и космолога, профессора Стивена Хоукинга, построившего термодинамику поведения «черных дыр», новый XXI век будет «веком сложности», сменяющим «век физики» и «век биологии» (Stephen Hawkins says the 21st centure will be the century of complexity. blogsscientificamerican.com).
Ускорение научно-технического прогресса, ставка крупных корпораций на непрерывные инновации, появление все новых возможностей и предложений для удовлетворения все более разнообразных и индивидуальных потребностей клиентов ведут к переходу от массового к индивидуальному производству, обострению конкуренции, росту априорной неопределенности и динамики изменений спроса и предложения и возрастанию других факторов сложности.
Такая растущая сложность окружающего нас мира приводит к кризису современного менеджмента и сокрушает фундаментальные устои традиционных иерархических структур управления корпорациями, которые в силу сложившихся иерархий и присущей им бюрократии, точь-в-точь как вымершие динозавры, часто оказываются не способными умно, точно и быстро реагировать на важные, быстро происходящие события изменений в среде, что приводит к потере эффективности, утрате конкурентных преимуществ и очевидным дальнейшим неотвратимым последствиям.
Что же должны делать управленцы в современном мире, чтобы уйти от «черных дыр» экономики, стать продуктивными и эффективными и выполнять больше заказов меньшим числом ресурсов?

Новая теория сложности

С эпохи Возрождения и до второй половины XX века в науке царило «линейное мышление» и стремление к редукционизму, т. е. желанию упрощать модели рассматриваемых объектов и процессов.
В основе линейных представлений лежит убеждение, что результат суммарного воздействия на систему есть сумма воздействий, а эффект прямо пропорционален воздействию.
Это утверждение справедливо для большого числа случаев, а именно для систем, которые находятся вблизи состояния равновесия.
Классические законы Ньютона (механика), Ома (электричество), Гука (теория упругости), Мальтуса (рост популяций), даже Максвелла (электродинамика) и Шредингера (квантовая механика) – линейны.
Из математических свойств линейных систем следует однозначный детерминизм. Следствие однозначно определяется причиной. Существует единственно правильное решение. «Линейная наука» изучает только устойчивые процессы, воспроизводимые в эксперименте.
На базе линейной науки развилась механика (в том числе небесная механика), строительное дело, электротехника. Триумф линейной науки – космические полеты.
Отклонения описывали в качестве малых нелинейных добавок. Но что дальше?
К сожалению (а точнее, к счастью), далеко не все явления природы линейны, устойчивы и воспроизводимы. Так, все живые системы, от клетки до человечества, и эволюционирующие неживые системы находятся в состоянии, далеком от равновесия. Не укладываются в ли-

нейные теории зарождение атмосферных вихрей, движения горных масс, образование галактик, пятен планктона в океане, процесс производства белков в клетках, формирование городов, написание статей, творческая деятельность, многие социальные, психологические и другие процессы.

Современный кибернетический словарь выделяет различные виды сложности. Наиболее сложной «сложностью» является динамическая сложность, которая может возникнуть даже в простых системах. Сложность обычно является результатом взаимодействий в системе, происходящих с некоторым временным лагом. Существование многократно взаимодействующих обратных связей в системе означает, что трудно изолировать влияние интересующих факторов. Многие факторы изменяются одновременно, осложняя интерпретацию системного поведения и снижая эффективность каждого цикла исследований. Задержки также приводят к неустойчивости динамических систем, задерживая отрицательную обратную связь и увеличивая тем самым тенденцию к колебательным процессам в системе. Колебание и неустойчивость уменьшают возможность управления связанными друг с другом переменными (т. е. возможность различать причину и следствие), что также осложняет изучение системы.

Таблица 1. Характеристики традиционных и новых предприятий

Традиционные предприятия	Новые предприятия
Централизация функций	Децентрализация функций
Иерархическая структура, жесткие связи	Сетевая структура, переменные связи
Закрытость к среде	Открытость к среде
Объем знаний, используемых в принятии решений, строго фиксирован, решения принимаются по формальным правилам бизнес-процессов	Объем знаний не фиксирован, приоритет имеет приобретение новых знаний, решения принимаются не формально, а по существу ситуации
Назначение сотрудников: плановый подход, все ресурсы распределены заранее	Назначение сотрудников: рыночный подход, ресурсы распределяются по мере необходимости и потребности
Распределение ресурсов статическое, на основе штатного расписания, статуса и должностных инструкций	Распределение ресурсов динамическое, на основе знаний и опыта, компетенций, конкуренции и кооперации
Выдача команд «сверху вниз» по жесткой иерархии	Переговоры «равный с равным», круг не ограничен (каждый с каждым), необходимые участники выбираются по ситуации
Пакетное жесткое планирование, следование регламентам и инструкциям	Гибкое планирование, поиск компромиссов, принятие решений по ситуации в реальном времени
Полная определенность	Полная неопределенность
Коммуникации регламентированы	Коммуникации не регламентированы
Тотальный внешний контроль	Внутренняя мотивация
Постоянная месячная оплата	Переменная (сдельная или почасовая) оплата

Где-то уже зреет новая теория сложности, которая будет оперировать с какими-то новыми атрибутами и понятиями. Изменения, безусловно, затронут и организацию работы как простых вычислительных узлов, так и суперкомпьютеров. Еще в 2004 году А.С. Нарьяни в статье «Модель или алгоритм: новая парадигма информационных систем» предсказывал появление вычислительных устройств, в которых вычисления строятся не на традиционных алгоритмах, а на моделях. Сейчас многим понятно, что в будущем супервычисления будут реализовываться асинхронными взаимодействиями различных достаточно сложных динамических систем (моделей) вместо огромных наборов простых триггеров.

Новые тенденции в управлении

Кризис современного менеджмента, обусловленный ростом сложности окружающего мира, во многом до сих пор базирующегося на ключевых идеях идеальной бюрократии М. Вебера, наиболее отчетливо проявляет себя в условиях, когда окружающий мир становится все более неопределенным, неустойчивым и

быстроизменяющимся. Причина такой ситуации – в стремлении менеджеров к сохранению иерархии власти и игнорировании личных качеств подчиненных сотрудников, превращающем их в рядовых «винтиков», что рано или поздно оказывается неприемлемо при организации любой новаторской созидательной деятельности: предпринимательской, научно-исследовательской, инженерной и любой другой.

Впечатляет статистика обычно тщательно скрываемых реалий, представленная в исследованиях компании Standish Group, посвященных эффективности проектной деятельности: из всех проектов, завершенных в 2010 году, только 32% проектов являются успешными, 44% – спорными (имеющими перерасход средств, превышение бюджета, другие недостатки), а 24% – провальными.

Факт, что основные резервы нашего общества для повышения эффективности деятельности любой организации следует искать не в совершенствовании устаревшей по своей сути бюрократии, а в более полном использовании интеллектуальных и волевых ресурсов людей, постепенно осознается и пионерами бизнеса.

Контуры форм новых организаций будущего уже вполне отчетливо проявляют себя в сравнении с существующими (табл. 1).

Идеализированная модель любой новой будущей организации – сообщество сотрудничающих на равной основе, продуктивных и эффективных, хорошо организованных людей, фактически представляющих собой самостоятельные автономные компании, где каждый «сам себе менеджер» (или добровольно выбравших себе менеджера) и где каждый, в зависимости от своей результативности, может одновременно участвовать в деятельности многих организаций.

Новый подход к управлению требует кардинально новых моделей, методов и программных средств поддержки принятия решений, в которых осуществляется постоянная проактивная коммуникация с пользователями и решение ищется в виде баланса интересов, обеспечивающих консенсус участников процессов принятия решения.

Научное развитие новой управленческой парадигмы ведется в работах проф. В.А. Виттиха из Института проблем управления сложными системами РАН (г. Самара) уже с начала 2000-х годов.

Похожая метаморфоза должна произойти и в суперкомпьютинге. Жесткая иерархия связей, централизованное планирование выполнения операций и т. п. будут не в состоянии обеспечить адекватный рост производительности при переходе к следующей – экзафлопсной – эре.

Рис. 2. Отличительные особенности мультиагентных систем

Смена вех в компьютерной индустрии

Этот глобальный вызов не мог быть не услышан и не принят в мире компьютерных технологий. Новый этап в развитии информационных технологий в ближайшем будущем связывается с мультиагентными системами (МАС), которые по своей значимости постепенно выходят на уровень критических нано- и биотехнологий, отмечается на сайте Ассоциации AgentLink Европейского союза, объединяющей разработчиков и пользователей таких систем.

Этой Ассоциацией представлен план (дорожная карта) развития этого направления до 2020–2030 годов, имеющий девиз «Вычисления как взаимодействия» (Computing as Interactions), который выражает важную суть данной технологии, позволяющей от централизованных, монолитных и последовательных программ с заранее фиксированной структурой перейти к распределенным сообществам автономных программ, работающих асинхронно и квазипараллельно, способных самостоятельно формировать требуемые структуры

и взаимодействовать для решения поставленных задач.
Причина стремительного развития направления МАС во многом связана с возможностью создания компьютерных систем нового поколения, использующих принципы самоорганизации и эволюции, характерные для поведения живых систем, например, колонии муравьев или роя пчел. Новый класс таких систем даже получил особое название Swarm Intelligence («Интеллект роя»), – подчеркивая ту особенность, что интеллект в таких системах не содержится в каких-то специальных компонентах (например, модулях дедукции или индукции), а рождается в ходе тонких взаимодействий совершенно простых, но автономных программ (агентов), не имеющих развитых интеллектуальных способностей.

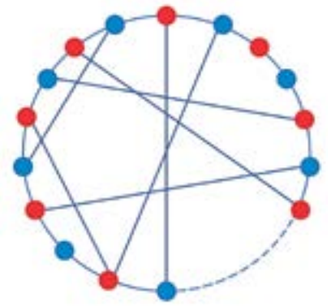


Рис. 3.
Топология вычислительного кластера

Историческая справка

Принято считать, что мультиагентные технологии появились в конце 1980-х годов на стыке достижения в области объектного программирования, параллельных вычислений, искусственного интеллекта, интернет-технологий и телекоммуникаций.
В рамках общего направления по разработке мультиагентных систем существует множество направлений, в числе которых особо выделяется направление по поиску новых методов решения сложных задач, не решаемых или плохо решаемых классическими математическими методами, на основе принципов самоорганизации и эволюции. Первоначально считалось, что это первый шаг в использовании биологических принципов при разработке программных систем, в связи с чем данное направление изначально называли bio-inspired («вдохновленным биологией»).
Но постепенно стало ясно, что этот подход открывает совершенно новую парадигму самоорганизующихся систем, с присущими им собственной феноменологией и большими перспективами вырасти в особый новый класс интеллектуальных систем так называемого «эмерджентного интеллекта» («вспыхивающего интеллекта»), возникающего спонтанно, в заранее непредвиденные моменты времени, которая, по образному выражению Европейской программы развития мультиагентных технологий, пока еще находится в «эмбриональном состоянии». Последние открытия в области супрамолекулярной химии, на стыке неорганической и органической химии, объясняющие процессы образования все более сложных молекул из простых за счет взаимодействия и «переговоров» молекул на своем особом химическом языке, показывают огромную фундаментальную роль процессов самоорганизации в раскрытии такого ключевого феномена, как «жизнь», и помогают разрабатывать новые принципы устройства и работы мира агентов.

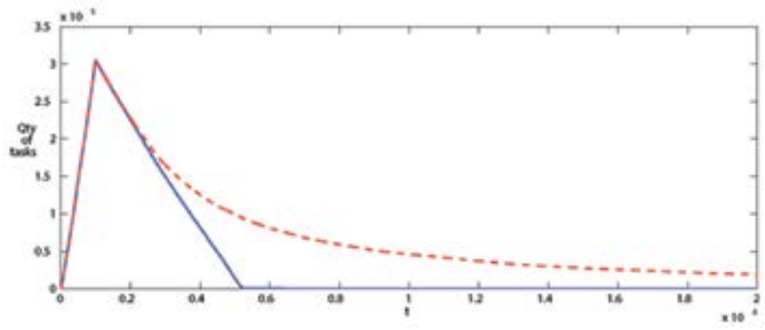


Рис. 4. Зависимость общей длины очереди заданий от времени

Мультиагентная система состоит из автономных программных агентов, способных воспринимать ситуацию, принимать решения и взаимодействовать с себе подобными для согласования своих решений. Решение любой сложной задачи в такой системе зарождается и самоорганизуется (возникает и развивается), при этом формируясь эволюционным путем, когда принятые ранее решения могут многократно пересматриваться и самоулучшаться (при наличии времени), за счет постоянного взаимодействия десятков и сотен тысяч агентов, непрерывно конкурирующих и кооперирующих друг с другом.
В отличие от традиционных громоздких централизованных, «монолитных», последовательных программных систем, МАС представляет собой распределенное сообщество агентов, действующих параллельно и на основе переговоров и потому способных гибко и быстро реагировать на любые события, разрешая конфликты и перестраивая сеть из десятков и сотен тысяч связей в реальном времени.
Это позволяет решать задачи самой высокой сложности, неподдающиеся решению другими способами, например, в области планирования и оптимизации ресурсов, распознавания образов, понимания текстов и ряда других.
Важно отметить, что мультиагентная система должна и может решать сложные задачи в реальном времени, когда качество и эффективность решения напрямую зависит от самого момента времени: например, для управления ресурсами, где чуть задержался с решением – и потерял заказ, или поспешил с решением – и образовался простой ресурсов.
Эти преимущества выгодно отличают такой подход от обычных пакетных решений по планированию и оптимизации ресурсов, не чувствующих особенностей момента времени и самого хода времени, когда время для принятия решений может просто уйти.

Балансировка загрузки элементов супервычислителя

Показательный пример преимущества мультиагентных технологий перед традиционными алгоритмами балансировки узлов вычислительных сетей представлен на международной конференции IEEE по управлению и принятию решений CDC-2013 (N. Amelina, O. Granichin, A. Kornivetc. Local voting protocol in decentralized load balancing problem with switched topology, noise, and delays, 2013).
Для кластера из 1024 серверов была смоделирована задача о распределении 1 млн заданий с неизвестной заранее трудоемкостью. Задания поступали на разные серверы (выбор производился случайным образом). Время поступления заданий и их длительности моделировались распределениями Пуассона. Серверы были соединены в кольцо с помощью 1023 связей, кроме того, для ускорения балансировки можно было на каждом такте пользоваться еще 1024 связями, устанавливаемыми p2p случайным образом (см. рис. 3).
В работе исследовалась применимость протокола локального голосования типа стохастической аппроксимации, который имеет мультиагентную природу: каждый узел «забирает» или «отдает» задания на основе локальных решений по сравнению своей загрузки только с зашумленными и, может быть, запоздалыми данными о «соседних» узлах (с теми, с которыми в данный момент есть связь).
Полученные строгие математические результаты об эффективности алгоритма хорошо иллюстрируются результатами моделирования на рис. 4. После прекращения поступления новых заданий система с перераспределением заданий по протоколу локального голосования заканчивает выполнение всех очередей за «линейное» время, определяемое вторым собственным значением матрицы топологии сети. На том же рис. 4 показано экспоненциальное затухание общего времени выполнения заданий в системе без их динамического перераспределения.
В настоящее время МАС активно и успешно используются для решения сложных задач управления ресурсами в реальном времени в проектах Группы компании «Генезис знаний» (Самара):

- *Smart Aerospace – мультиагентная система построения программы полета, планирования грузопотока и расчета ресурсов МКС;*
- *Smart Truck – мультиагентная система управления междугородними грузоперевозками;*
- *Smart Field Service – мультиагентная система управления аварийными и ремонтными бригадами;*
- *Smart Airports – мультиагентная система управления наземными сервисами аэропорта;*
- *Smart Factory – мультиагентная система управления цехами машиностроительных предприятий;*
- *Smart Supply Chain – мультиагентная система управления цепочками поставок товаров в магазины;*
- *Smart Railways – мультиагентная система управления ресурсами РЖД;*
- *Smart Satellites – мультиагентная система управления группировкой спутников.*

Эти системы помогают предприятиям осуществить переход к принятию решений в реальном времени, что требует поддержки автономного цикла управления ресурсами, включающего реакцию на события, распределение и планирование ресурсов, оптимизацию решения (при наличии времени), согласование с пользователями, мониторинг и контроль выполнения построенного плана, а также перепланирование при расхождении плана и факта, – цикла, присущего любым живым организмам. Оцениваемый выигрыш предприятий от перехода к экономике реального времени – 20-40% (при том же числе ресурсов).
Направления дальнейших исследований и разработок связываются с развитием принципов достижения многокритериального консенсуса группой ЛПР в рамках ситуационного управления, созданием сетецентрических систем для поддержки механизмов принятия и согласования решений на основе развития логики и протоколов взаимодействия для виртуального круглого стола, использованием онтологий, обучением из опыта, а также переходом к высокопроизводительным вычислениям в облачных приложениях.
В СПбГУ начали разработку новой адаптивной мультиагентной операционной системы реального времени, основными метафорами которой будут «ресурсы» и «задачи» вместо традиционной файловой системы.

Рейтинг TOP50: куда крадемся?

Текст Д. А. Никитенко, н. с. НИВЦ МГУ, dan@parallel.ru

В первый день работы Международной суперкомпьютерной конференции «Научный сервис в сети Интернет: все грани параллелизма», проводимой Российской академией наук и Суперкомпьютерным консорциумом университетов России при поддержке Российского фонда фундаментальных исследований 24 сентября 2013 года состоялось объявление новой, девятнадцатой редакции списка TOP50 наиболее производительных суперкомпьютеров России и СНГ. Сайт рейтинга: http://top50.supercomputers.ru.

Девятнадцатая редакция списка продемонстрировала незначительные изменения: за полгода с момента публикации предыдущей редакции рейтинга появилось всего 3 новых системы в списке, и одна система была обновлена. Суммарная производительность систем на тесте Linpack за полгода выросла чуть более чем на 1% – с 3355.9 Тфлопс/с до 3393.9 Тфлопс/с, а суммарная пиковая производительность систем списка – на 3% и составила 5877.8 Тфлопс/с (5707.4 Тфлопс/с в предыдущей редакции списка).

Лидеры списка остались неизменными: на вершине – суперкомпьютер МГУ «Ломоносов» производства компании «Т-Платформы», чья пиковая производительность составляет 1700.21 Тфлопс/с, а производительность на тесте Linpack – 901.9 Тфлопс/с. На втором месте – суперкомпьютер МВС-10П производства Группы компаний РСК, установленный в Межведомственном суперкомпьютерном центре Российской академии наук, с пиковой производительностью 523.83 Тфлопс/с и производительностью на тесте Linpack 375.70 Тфлопс/с. На третьем месте – суперкомпьютер производства Hewlett-Packard с пиковой производительностью 317.40 Тфлопс/с и производительностью на тесте Linpack 160.90 Тфлопс/с.

Подробная статистика отражена в соответствующем разделе сайта рейтинга. Приведем некоторые особенности текущей редакции:

- *порог вхождения в список составляет 13.5 Тфлопс/с (12.36 Тфлопс/с в предыдущей редакции);*
- *продолжает расти число систем, построенных с использованием ускорителей;*
- *появились системы с малым числом основных CPU, ориентированные на достижение производительности за счет ускорителей;*

не осталось систем, содержащих менее тысячи вычислительных ядер;

- *в качестве основной коммуникационной сети сохраняется тенденция к использованию Gigabit Ethernet (32% систем в списке), хотя лидирует все же InfiniBand (64% систем в списке);*
- *IBM и HP сравнялись по числу установок – представлено по 17 систем;*
- *в текущей редакции нет обновлений от отечественных производителей (Группа компаний РСК, компания «Т-Платформы»);*
- *минимум изменений самих систем – три новых и одна обновленная.*

Сравнимые темпы обновления имели место в осенних редакциях 2009 и 2006 годов, когда было зарегистрировано 5 и 7 изменений в списке соответственно.

Продемонстрированные темпы роста – чуть ли не самые низкие за всю историю ведения рейтинга. Подобная ситуация имела место лишь осенью 2009 года – в 11-й редакции списка прирост пиковой суммарной производительности всех систем составил менее 2%. Однако следует отметить, что последовавшая за тем 12-я редакция весной 2010 года уже продемонстрировала мощный прирост производительности, более чем вдвое увеличилась производительность первой пятерки списка: среди новых систем и обновлений был введен в строй флагман отечественных высокопроизводительных вычислительных систем – суперкомпьютер «Ломоносов», до сих пор (претерпев серию обновлений) остающийся на вершине списка.

Еще два примера возросшей активности после затишья – осенние редакции 2006 и 2012 годов, за которыми следовали достаточно яркие весенние редакции. Во многом это объясняется тем, что ввод систем в строй зачастую реально происходит к концу календарного года. Будем надеяться, что весна 2014 года порадует интересными обновлениями в списке, тем более что грядущая редакция списка будет юбилейной – двадцатой по счету.

Напомним, что первая редакция списка TOP50 была опубликована в декабре 2004 года. Хочется отметить, что за почти 10 лет ведения рейтинга, безусловно, многое изменилось. Изначально выбранные методы описания вычислительных систем и доступный инструментарий требуют доработки по целому ряду причин. Здесь и новые масштабы суперкомпьютеров, и полная их неоднородность, появление и широкое использование ускорителей и так далее. Неслучайно, что за это время появились новые тесты и рейтинги (Graph500, Green500 и др.). Быть может, было бы интересно видеть результаты, которые та или иная система достигает на других тестах помимо Linpack, какие места занимает в текущих редакциях других рейтингов.

Особенно следует обратить внимание на отсутствие в рейтинге сведений по энергопотреблению систем, ведь в современных условиях энергоэффективность становится одной из ключевых характеристик суперкомпьютеров. Востребованность удобного, функционального, содержательного ресурса, конечно, крайне высока как среди разработчиков, так и среди держателей систем. В связи с этим следует подчеркнуть, что составители рейтинга, имея ряд собственных соображений по его доработке, абсолютно открыты для предложений насчет того, какую функциональность хотелось бы добавить, какой информацией было бы желательно дополнить существующие описания вычислительных систем.

К тому же более удачный момент для обновления самого рейтинга и функционала его сайта, чем сейчас, на рубеже юбилейной редакции, и подобрать-то сложно.

Следующая, двадцатая редакция списка TOP50 самых мощных компьютеров СНГ будет объявлена в апреле 2014 года на Международной научной конференции «Параллельные вычислительные технологии (ПаВТ) 2014».

TOP500

Текст Игорь Лёвшин

Ноябрьский список TOP500 2013 небогат сенсациями. В десятке единственная новость – ворвавшийся на 6-ю строчку швейцарец Piz Daint. Безусловно, это выдающаяся система последнего поколения Cray XC30 с Intel Xeon E5-2670 – вместо традиционных для Cray Opteron; с сетью Aries и ускорителями NVIDIA K20x. Но сенсацией это никоим образом не стало: об этой машине отечественные медиа говорят давно (им полюбилось название). Этот год – не лучший для российских участников TOP500, особенно если смотреть по критерию количества систем в списке. Важным для нас этот показатель является потому, что для России характерна большая централизация ресурсов, которая вряд ли идет на пользу стране и отрасли HPC. Пять систем в последнем списке – это, конечно, маловато. Подобный провал был года полтора назад, когда Россию обогнала Польша (при этом отстав по суммарной производительности). Потом положение немного выправилось (8 систем). Ну а золотые дни были в начале 2010-х, когда в списке бывало по 11–12 российских машин. Конечно, нелепо было бы считать, что 10 машин против 5 говорит о том, что положение было в два раза лучше, но и поводов для оптимизма маловато. Да и место по этому показателю какой-то нехорошее – 13-е.

Падение происходит на фоне взлета Индии. За два года число индийских машин в TOP500 выросло с 2 (период глубокого спада) до 12. За это время мы слышали громкие заявления о том, что Индия первой достигнет знаменитого экзаскейла, о том, что кругленькие суммы выделяются на грандиозные программы поддержки компьютерщиков, физиков и на космос. Но Индия начинает отнюдь не с нуля – около пяти лет назад в списке бывало двузначное число индийских машин. В противоположность Китаю, поражавшему неумолимым ростом с нуля. В последнем списке, кстати, Китай вышел на плато и даже немного завалился вниз. Но на фоне 1-го места Tianhe-2 с ее 33 с лишним ПФлопс этого можно и не заметить. Возможно, это просто коррекция: в предыдущих списках было немало уж очень странных строчек, посвященных китайским суперкомпьютерам.

Похожую на индийскую картину роста можно увидеть в Канаде. Из минимума в 2 системы два года назад страна прыгнула до 9, потом до 11, а теперь и до 12 участников TOP500.

Отрыв Tianhe-2 от преследователей впечатляет. И не видно пока, кто готов дать бой. 33.86 Пфлопс против 17.59 у Titan – это, справедливости ради надо сказать, отрыв отнюдь не рекордный (в относительных единицах, конечно). В истории TOP500 бывали отрывы куда более удивительные. В 2005-м у IBM BlueGene/L производительность была в 3 раза выше ближайшего конкурента – 91.3 IBM BGW.

Вообще, изучение списка со статистикой по странам дает любопытные результаты. Один из них такой: количество стран с первых лет списка (ему исполнилось 20 лет) и до наших дней удивительно своим постоянством: это 27–28 стран. Сами страны меняются, появляются и исчезают Ирландия, Бельгия, Словения, Белоруссия, Венгрия, Греция, но их общее количество стабильно. Скорее всего, можно говорить о том, что на мировом уровне централизации ресурсов или их децентрализации не происходит. К тому же можно предположить, что, ощутив себя участником большого суперкомпьютерного мира, страны находят в себе силы предпринять что-то для возвращения в престижный список.

Удивителен пример Японии. Во времена, когда список только еще появился на свет, у этой страны были трехзначные цифры. Далее последовал чудовищный спад – бывало и меньше 20. Титаническими усилиями (мощными правительственными финансовыми вливаниями в суперкомпьютерную индустрию) положение хоть как-то выправляется, удалось закрепиться в районе 30, заведомо впереди европейских лидеров: Великобритании, Германии и Франции.

Интересный критерий – порог вхождения в список. Но захотелось глянуть на порог вхождения в первую страницу, то есть в первую сотню списка. Можно предположить, что сравнение порога мирового списка TOP500 с порогом десятки российского TOP50 должно коррелировать с количеством российских систем в TOP500. Действительно, отставание нашего порога десятки от мировой полутысячи было совсем маленьким в 2010-м году. Сейчас же – больше чем в полтора раза.

Все-таки делать какие-то заключения общего характера по 50 строчкам намного более рискованно, чем по 500. Поэтому вернемся к мировым данным.

Наблюдая за порогами сотни и полутысячи, можно заметить, что экспонента дает сбой. Раньше за 5 лет производительность подрастала примерно в 20–30 раз. Показатель первой строки, пороги сотни и полутысячи росли почти синхронно. Сейчас порог сотни подрос немного больше чем в 10 раз, а полутысячи – даже меньше, чем в 10. При этом Tianhe-2 быстрее чемпиона 2008 года – Roadrunner – в 30 раз (у него 1105.0 Пфлопс). Гиганты растут быстрее, чем крепкие середнячки. Обозы не успевают подтягиваться к передовой HPC.

Новости

Петафлопс на квадратном метре

На конференции SC'2013 в Денвере РСК установила мировой рекорд вычислительной плотности в 1 Пфлопс на одну стойку при занимаемой площади 1 м^2 и объеме 2.2 м^3. Такой результат был показан на решении RSC PetaStream с прямым жидкостным охлаждением. Одна стойка RSC PetaStream состоит из 1024 вычислительных узлов и обеспечивает отвод более 400 кВт тепловой мощности. Это мировой рекорд энергетической плотности. Каждый узел системы построен на базе 60-ядерного сопроцессора Intel Xeon Phi 5120D с 8 ГБ высокоскоростной памяти DDR5.

Все 1024 вычислительных узла объединены между собой высокоскоростными соединениями на базе технологии InfiniBand FDR, обеспечивая революционное сверхплотное высокопроизводительное решение с более чем 250 000 потоков в одной стойке на базе архитектуры x86.

Для обеспечения максимальной плотности упаковки, высокой пропускной способности ввода/вывода, а также надежности вычислений и управляемости несколько узлов RSC PetaStream группируются в один модуль, обеспечивающий узлам жидкостное охлаждение, высокоэффективное электропитание и преобразование электроэнергии, а также возможность агрегации узлов в стойку.

Суперкомпьютинг-13

Текст С. Соболев, Д. Никитенко, А. Антонов

25-я Международная конференция и суперкомпьютерная выставка серии SC проходила в Денвере, штат Колорадо, с 17 по 22 ноября. Юбилейное мероприятие было проведено с особым размахом, которому не помешал даже осенний «шатдаун» – локальный кризис государственного финансирования, из-за которого пострадали многие проекты и в суперкомпьютерной области. Здесь хотелось бы отметить чуткость организаторов: понимая, что с проблемами могла столкнуться существенная часть участников бюджетного сектора, например сотрудников университетов и национальных лабораторий, по специальному запросу они были готовы продлить сроки льготной регистрации, организовать бесплатную отмену брони в отелях и т. д. К счастью, за месяц до начала Суперкомпьютинга кризис был разрешен. Конечный итог – свыше 10 тысяч участников, 400 стендов на выставке и крайне насыщенная научная программа, в которой порой тяжело было сориентироваться – столь велик был выбор параллельно проходящих докладов, тренингов, семинаров.

Лейтмотив – конечно же, приближающийся экзаскейл во всех его проявлениях: проблемы эффективного использования сотен тысяч процессорных ядер, новые модели программирования, надежность и отказоустойчивость сверхбольших систем и выполняющихся в них приложений.

Одним из заметных событий американского Суперкомпьютинга стало объявление очередной, 42-й по счету редакции списка TOP500 самых мощных суперкомпьютеров со всего мира. Впрочем, чуда не произошло: по сравнению с предыдущей редакцией в первой пятерке – ни одного изменения. Лидерство по-прежнему удерживает китайский суперкомпьютер Tianhe-2 («Млечный путь») с производительностью 33.86 Петафлопс на тесте Linpack. Зато в первую десятку ворвалась 6.27-петафлопсная система Cray XC90 Piz Daint (Пиц Даинт), названная в честь одной из гор Швейцарских Альп и установленная в Швейцарском национальном суперкомпьютерном центре. Самый мощный российский суперкомпьютер «Ломоносов» отступил на шесть позиций вниз и теперь занимает 37-е место международного рейтинга.

На выставке обращало на себя внимание обилие представителей Азии: крупные и мелкие промышленные компании, университеты и суперкомпьютерные центры Японии, Китая, Южной Кореи (предлагалась даже специальная экскурсия по азиатским стендам). Пожалуй, больше всего было представлено именно японских организаций, зато китайская компания Inspur, при участии которой был создан Tianhe-2, провела свое мероприятие «Диалог Китая и США по HPC». Основные тренды выставки – решения для предоставления высокопроизводительного ресурса как сервиса в рамках парадигмы Cloud Computing; научная визуализация, да не простая, а трехмерная – специальные очки лежали, наверное, у каждого второго большого экрана; водяное охлаждение (например, всем желающим предлагали перерезать трубку с охлаждающей жидкостью и увидеть, как жидкость моментально уходит из системы, не проливаясь внутрь дорогостоящей аппаратуры). Мелькало упоминание Big Data: действительно, обработка больших объемов данных рано или поздно потребует суперкомпьютерных ресурсов. Многие компании, не ограничиваясь статичным показом своих продуктов, прямо на месте проводили семинары по использованию своих решений. А на стенде IT-гигантов, таких как Intel или NVIDIA, семинары шли непрерывно, по сути дополняя программу конференции. Компанию NVIDIA на выставке хотелось бы отметить особо: лекторами на ее стенде были известнейшие в HPC-мире люди вроде Томаса Стерлинга, Сатоши Мацуока и Джека Донгарры; здесь можно было не только узнать о нововведениях CUDA 6, но и посмотреть видеоклип с летающей змеей и демонстрацией ее модели; ну а гости стенда в шарфах фирменного зеленого цвета были заметны по всему Денверу.

Российских представителей на выставке в этом году было всего два. Группа компаний РСК представила аппаратное решение, позволяющее получить производительность до 1 Петафлопс в стандартной стойке. А сотрудники Южно-Уральского государственного университета на стенде РСК наглядно демонстрировали свой опыт в решении практических задач на суперкомпьютерах – и делились почти настоящими кусочками челябинского метеорита. Традиционно свой стенд был у МГУ. Крупнейший российский университет рассказывал о государственном проекте «Суперкомпьютерное образование», приглашал пройти обучение суперкомпьютерным технологиям по магистерской программе на английском языке, устанавливал контакты для новых совместных проектов и исследований в области комплексного анализа поведения суперкомпьютеров, приложений, потоков задач.

Следующий американский Суперкомпьютинг пройдет в ноябре следующего года в Новом Орлеане. Ну а сейчас самое время начинать готовиться к Лейпцигу – до очередной конференции и выставки ISC время пробежит незаметно...