Российский суперкомпьютер с глобально адресуемой памятью

Анатолий Слуцкин, Леонид Эйсымонт

В пылу массового увлечения вычислительными кластерами часто забывают, что эти системы непригодны для решения задач, требующих эффективной работы с глобально адресуемой памятью большого объема. Оказывается, что коэффициент полезного использования кластеров на таких задачах не превышает 5-10%, а в ряде случаев равен 0,1-1%. В силу характерных тенденций развития элементной базы данная проблема только обостряется, хотя количество задач такого класса растет. На Западе проблема создания вычислительных систем для эффективной работы с огромными массивами данных решается на уровне федеральных программ, а в России — в рамках работ по созданию перспективного суперкомпьютера с глобально адресуемой памятью.

Низкие значения коэффициента полезного использования (выраженная в процентах доля реальной производительности от пиковой) вычислительных кластеров означают, что имеется класс задач, которые просто не решаются на таких архитектурах. На фоне увеличивающегося разрыва между быстродействием процессора и памяти, а также роста количества задач, требующих эффективной работы с памятью огромного объема, эта проблема только усугубляется (к задачам такого класса обычно относятся задачи, связанные с обеспечением национальной безопасности). Как следствие, например в США, для создания вычислительных систем, решающих задачи упомянутого класса, принята федеральная программа DARPA HPCS.

Пиковая производительность компьютерной системы определяется по количеству характерных для некоторой задачи операций в секунду, которое может выполнить аппаратура, а реальная производительность — это количество тех же операций в секунду, которое на самом деле выполняется при решении этой задачи. Например, если микропроцессор может выполнять 6 миллиардов операций над числами в формате с плавающей точкой в секунду, а при решении некоторой задачи в течение одной минуты он успел выполнить лишь 36 миллиардов таких операций, то его реальная производительность равна 36/60 = 0,6 миллиарда операций в секунду — 10% от пиковой. Что тогда делал микропроцессор 90% времени? Либо он выполнял операции, которые не интересуют пользователя, либо простаивал, например, в ожидании выполнения операций с памятью.

Для создания систем с высокой реальной производительностью и эффективной глобально адресуемой памятью в ОАО «НИЦЭВТ» в инициативном порядке в 2000 году были начаты исследования, которые через два года переросли в государственный контракт — НИР «Беркут». Основные решения начального этапа работ по НИР «Беркут» [1] легли в основу оригинальной MTDF-архитектуры мультитредового суперкомпьютера, с управлением потоком данных и глобально адресуемой памятью большого объема. Наибольшее влияние на MTDF-архитектуру изначально оказали три разработки: проект HTMT создания гибридной мультитредовой системы петафлопсного класса [2]; работы группы В.С.Бурцева по машине, управляемой потоком данных [3]; мультитредовые системы Tera MTA и Cray MTA-2 [4]. Аналогичные работы в это же время были начаты и в США — в конце 2002 года появились первые сообщения о проекте DARPA HPCS [5], возведенном в ранг федеральной программы, который по сути решал те же проблемы, что и «Беркут». Последующие сообщения подтвердили, что и в России, и в США движение к суперкомпьютерам нового поколения идет с использованием примерно одинаковых архитектурных решений — различались лишь подходы к реализации процессора. В проекте «Беркут» был выбран вариант создания процессора на базе одной мультипроцессорной СБИС, перестраиваемой под разные мультитредовые вычисления (модель гетерогенной мультитредовости). Разработчики из США (проект Cray Cascade) могли позволить себе создание одновременно

нескольких СБИС для разного типа мультитредовых вычислений [5].

В 2005 году отечественные научно-исследовательские работы над суперкомпьютером продолжились в рамках проекта «Ястреб». В разрабатываемый процессор были дополнительно введены архитектурные возможности работы с крупнозернистыми статическими графами вычислений типа потоковых моделей (модели stream-based [6]). Сегодня прорабатывается поддержка мелкозернистых статических графов на реконфигурируемом поле функциональных устройств (данный подход используется в разработках MONARCH [7] и TRIPS[8]).

Сочетание мультитредовых и потоковых архитектурных средств позволяет сегодня говорить о возможности построения на их основе суперкомпьютеров от петафлопсного до экзафлопсного класса (1018 операций в секунду) [9].

Первый шаг

Прежде чем приступить к описанию архитектуры российского суперкомпьютера производительностью более 1 PFLOPS, надо сказать, что применяются два подхода к реализации мультитредовости процессора: большое количество ядер и малое число тредов в ядре; малое число ядер, но много тредов в ядре. Первый вариант реализуется в IBM Cyclops64, Sun Niagara, второй — в Cray MTA-2. В проекте «Ястреб» был выбран второй вариант, обеспечивающий большую толерантность к задержкам. Одновременно с процессором разрабатывалась специальная коммуникационная сеть с большой пропускной способностью при передаче коротких пакетов. Изучались варианты адаптивных бездедлоковых сетей, от N-тор до сетей Кэли и сетей Клоса [10, 11].

К 2007 году была построена потактовая имитационная модель будущей системы. На этой модели, работающей как на персональном компьютере, так и на вычислительном кластере из нескольких сотен узлов, велась отработка принципов работы, проводилось оценочное тестирование развиваемой реальной производительности суперкомпьютера, отлаживалось базовое программное обеспечение.

Сегодня работы над перспективным суперкомпьютером с глобально адресуемой памятью ведутся в рамках двух контрактов с Роснаукой и проекта «Ангара» ОАО «НИЦЭВТ» по созданию МТП-суперкомпьютера стратегического назначения (СКСН) мультитредово-потокового типа.

«Ангара»: цели, задачи, архитектура

Цель проекта «Ангара»— разработка компонентов и моделей суперкомпьютеров петафлопсного уровня с глобально адресуемой памятью большого объема для эффективного решения задач с различной пространственно-временной локализацией обращений к памяти. Среди целей проекта можно выделить следующие:

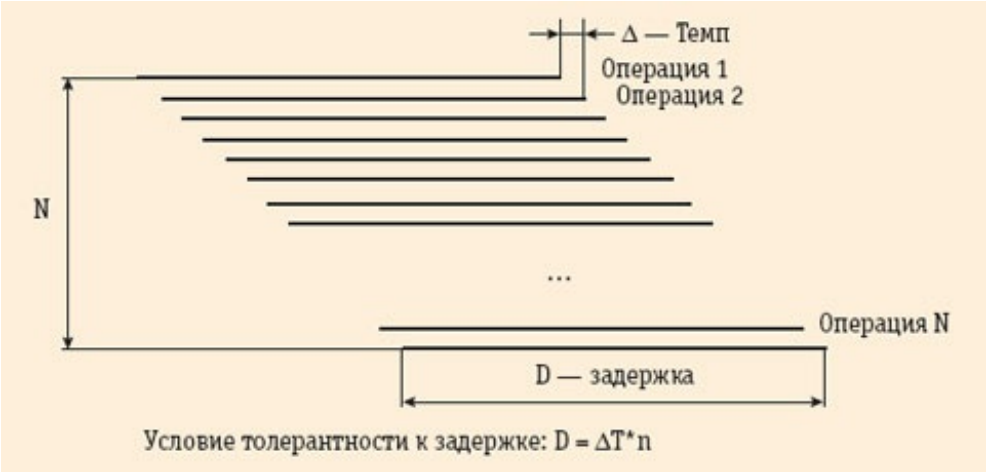
- разработка многоядерного мультитредово-потокового микропроцессора, работающего с распределенной общей памятью и адаптируемого к разным моделям организации вычислений;
- разработка коммуникационной сети с высокой конвейеризацией, адаптивностью, малым диаметром, обеспечивающей высокую пропускную способность передачи коротких сообщений и повышенную отказоустойчивость;
- разработка опытного образца суперкомпьютера с реальной производительностью 0,1 PFLOPS;
- разработка системного программного обеспечения нового поколения, ориентированного на адаптивную компиляцию программ, динамическую реконфигурацию в процессе счета, повышенную отказоустойчивость, разные модели вычислений;
- повышение на порядок продуктивности программирования в сравнении с современным

уровнем при обеспечении эффективного решения задач с интенсивной нерегулярной работой с памятью;
- разработка на базе многоядерного мультитредово-потокового микропроцессора универсальной высокопроизводительной платформы, в том числе и для бортовых применений.
- Исследования показывают, что современный перспективный суперкомпьютер должен строиться на основе нескольких общих архитектурно-программных принципов:
- **MT (multithreading).** Мультитредовая организация процессора и выполняемых им программ с целью обеспечения толерантности процессора к задержкам выполнения операций с памятью (100–500 тактов процессора) и коммуникационной сетью (1000 –10000 тактов).
- **DF (dataflow).** Управление вычислениями потоком данных с использованием статических и динамических графовых моделей.
- MD (message driven). Управление передачей сообщений для реализации потоковых моделей вычислений, а также для управляемой локализации посредством механизмов вызова процедур на удаленных узлах.
- **DAE (decoupled access/execute модель).** Организация работы процессора и программ, при которой выделяется вычислительная и обслуживающая ее невычислительная часть, выполняющая задачу обеспечения данными вычислительной части.
- **DSM (distributed shared memory).** Распределенная общая или глобально адресуемая память.
- **CMT.** Чип-мультипроцессорные технологии создания микропроцессорных СБИС типа «система на кристалле».
- **PIM (processor in memory) и NMP (near memory processor).** Технологии модулей памяти со встроенными процессорами обработки данных, либо с процессорами, размещенными в непосредственной близости к памяти.
- **RAS (reliability, adaptability, serviceability).** Обеспечение повышенной отказоустойчивости на аппаратном, системном и прикладном уровне.
- **PGAS и GAS (partitioned global address space languages).** Языки программирования для работы в разделенном на подразделы глобальном адресном пространстве, обладающие повышенным уровнем абстракции описания программ, ориентированных на описание сверхпараллельных и асинхронных вычислений, обладающих возможностями локализации вычислений на множествах данных.
- **AS (adaptive supercomputing).** Обеспечение высокого полиморфизма системы (наличие средств выполнения разнотипных программ) и ее реконфигурируемости (возможности выбора и настройки этих средств для эффективного выполнения программ разного типа).

Суть всех современных работ по СКСН— поиск оптимального сочетания и глубины реализации всех этих принципов.

Мультитредовость и толерантность к задержкам

Поясним, как обычно обеспечивается толерантность (нечувствительность) процессора к задержкам выполнения операций на некотором устройстве. Обычно это достигается за счет конвейеризации или распараллеливания выполнения на этом устройстве множества операций, запускаемых с высоким темпом. Несмотря на то, что в этом случае выполнение отдельной операции будет длительным, благодаря распараллеливанию среднее время выполнения операции будет определяться не задержкой, а темпом запуска операций (рис. 1).



Проблема в том, откуда взять такое большое количество операций. В векторных процессорах толерантность к задержкам памяти обеспечивается за счет того, что каждая операция выдается для элемента вектора. В современных векторных процессорах количество одновременно выполняемых операций с памятью может достигать нескольких тысяч [11]. Однако не все задачи векторизуются. В суперскалярных процессорах большое количество одновременно выполняемых операций с памятью появляется вследствие распараллеливания выполнения соседних команд программы, но это алгоритмически очень сложный и ресурсоемкий процесс, ограниченный по возможностям выдачи большого количества операций с памятью (около 30–40 обращений). Следует добавить, что, кроме задержек выполнения операций обращения к памяти, возможны задержки выполнения операций в арифметико-логических функциональных устройствах, задержки выполнения операций с коммуникационной сетью и др. По отношению к ним также необходима толерантность.

Наиболее популярный сегодня подход к обеспечению толерантности к задержкам разного типа — мультитредовая организация процессоров. Суть здесь в том, что такой процессор одновременно выполняет не одну, как раньше, а много программ-тредов, переключаясь с одной на другую без каких-либо больших затрат времени, за такт. Это позволяет одновременно выполнять много команд, поступающих от разных тредов.

Структура многоядерного мультитредового процессора МТП-суперкомпьютера представлена на рис. 2. Этот процессор содержит несколько мультитредовых ядер (МТ-ядер), в каждом из которых может выполняться несколько программ, содержащих, в свою очередь, много подпрограмм-тредов. Каждая такая программа представляется в ядре доменом защиты.

Домен защиты — это набор дескрипторов сегментов данных [12] и страниц команд, дескрипторы виртуальных тредов некоторой задачи и отведенных для нее тредовых устройств, регистры управления и контроля. Один из доменов защиты ядра всегда содержит выполняемую часть распределенной операционной системы МТП-суперкомпьютера.

Одно МТ-ядро содержит множество тредовых устройств (TU). Каждое такое устройство включает в себя счетчик команд выполняемой им программы-треда с признаками состояния этой программы, регистры управления и контроля, наборы разных архитектурных регистров — чисел с фиксированной и плавающей точкой, однобитовых признаков, регистры хранения адресов команд для передачи управления. Каждое TU имеет собственные наборы таких регистров. Единственный разделяемый TU ресурс — пул теневых регистров для обработки исключительных ситуаций.

Загрузка программ-тредов на TU инициируется либо явно, командой CREATE, либо неявно, по поступившему в МТ-ядро пакету-сообщению из коммуникационной сети или автоматически порожденному пакету внутри МТ-ядра из-за возникшей в нем исключительной ситуации. Снятие программы-треда с TU также инициируется явно, выполнением некоторых машинных

команд, или неявно, аппаратными средствами. Снятая с TU программа-тред может быть опять загружена на TU и продолжена с места ее останова. В некоторых случаях возможно даже перемещение в TU другого MT-ядра этого процессора или вообще в процессор другого вычислительного узла.

Выполнение мультитредовых программ в ядре происходит следующим образом. На каждом такте MT-ядра в блоке ISSUE (рис. 2) анализируется несколько команд, но по одной от каждого TU, на предмет готовности их выдачи на выполнение в функциональные устройства FU (FXU, FPU, LSU, SPU из рис. 2). Принадлежность TU к тому или иному домену защиты не имеет значения.

Команда готова к выдаче, если регистры-операнды не ожидают записи в них значений, регистр-результат готов для изменения его значения и есть FU, готовое выполнить эту команду. Количество анализируемых на готовность команд определяет параллелизм выполнения машинных команд в MT-ядре (ILP), мы ориентируемся на ILP=4, а в старших моделях — на ILP=8.

Если текущая команда некоторого TU не готова к выполнению, то она ожидает готовности, а следующие команды этого TU не анализируются до тех пор, пока не будет выполнена ожидавшая готовности команда. Таким образом, для одного TU принята достаточно простая дисциплина выдачи команд типа in-order (по порядку), однако с возможностью совмещения выполнения соседних команд. Завершение выполнения команд может производиться не в порядке их следования, то есть out-of-order, но буфер переупорядочивания команд при этом не используется. Одновременно в процессоре могут завершаться несколько команд от одного или нескольких TU.

Арифметико-логические команды после выдачи попадают в функциональные устройства FPU, FXU и SPU. Выдаваемый таким образом на эти устройства поток параллельных команд обеспечивает толерантность к задержкам выполнения операций в функциональных устройствах.

Команды обращения к памяти проходят сначала LSU, а потом попадают в блоки MMU, где происходит трансляция виртуальных адресов [12]. В зависимости от результата трансляции обращение может попасть в собственную локальную память узла, либо выяснится, что должно быть обращение к памяти другого узла через коммуникационную сеть. Во втором случае происходит запрос в блок MSU, который по этому запросу формирует пакет-сообщение, направляемый далее в коммуникационную сеть для передачи в узел, где находится требуемый участок памяти. Такой пакет, попав в целевой узел, принимается одним из блоков MSU этого узла, в этом блоке производится его дешифрация, после чего он передается на обработку в соответствующий блок MMU. Получается, что каждый блок MMU выполняет как собственные, так и наведенные удаленными обращениями из других узлов обращения к локальной памяти.

Таким образом, в памяти выполняется много операций, выданных разными тредовыми устройствами своего ядра, а также других процессоров суперкомпьютера. Количество тредовых устройств выбрано так, чтобы соблюдались условия толерантности по отношению к задержкам разного типа (рис. 1). По этим причинам выбирается большое количество тредовых устройств — до 128 на одно MT-ядро, что позволяет иметь в одном ядре до 1024 выполняемых обращений к памяти или сети (до восьми обращений разрешается выдать одному треду), а в восьмиядерном — более восьми тысяч.

Используется еще ряд приемов повышения толерантности и эффективности работы с памятью:

- вводятся команды обращений к памяти за короткими векторами;
- вводится кэш-память при модулях памяти узлов;
- применяются механизмы виртуальной адресации программ-тредов и TU, что позволяет

отрабатывать сверхдлинные задержки обращений к памяти;
- применяются методы реализации гетерогенных тредовых моделей, когда одни треды «специализируются» на обращениях к памяти с большими задержками, а другие— на «вычислениях»;
- введено большое количество атомарных операций с памятью.

Потоковость

В предыдущем разделе было показано, как решается проблема обеспечения толерантности процессора к задержкам обращений к памяти, но можно решать данные проблемы за счет сокращения количества этих обращений, что позволяют сделать применяемые в МТП-суперкомпьютере потоковые модели типа статического графа тредов. Это одна из двух потоковых моделей МТП-суперкомпьютера. Заметим, что такая потоковая модель фактически означает введение в процессор обработки векторов [6].

Другая потоковая модель - модель динамических графов, она нацелена на автоматическое порождение большого количества тредов и обеспечение большей асинхронности выполнения программ за счет сильно децентрализованного управления через механизм графов потоков данных.

Механизмы динамических графов потоков данных известны с конца 60-х годов и использовались в машинах, управляемых потоком данных (data-flow машины), однако необходимость в ассоциативной памяти и большие накладные расходы на организацию выполнения операций над данными долгое время препятствовали практическому внедрению машин такого типа. Вместе с тем за прошедшие годы были разработаны компромиссные схемы, которые сегодня допускают эффективную реализацию даже в программном варианте, используя огромную вычислительную мощность и параллелизм процессоров, особенно с мультитредовой организацией. Этот подход и применяется в МТП-суперкомпьютере. Более того, здесь используется еще и доработанная схема организации потоковых вычислений с динамическими графами на ассоциативной памяти [3].

Реализация динамических графов и данных с ассоциативным доступом в МТП-суперкомпьютере аппаратно-программная— используются механизм запуска тредов по поступившему в МТ-ядро сообщению, методы хеш-адресации для реализации ассоциативной памяти, встроенные микропрограммы стандартной обработки в тредах сообщений, специфических для моделей динамических графов. В порядке эксперимента исследуется и возможность, когда из блока MSU, получившего сообщение с простой операцией и данными, выдается заказ непосредственно в требуемое для выполнения этой операции FU. Таким образом, FU снабжаются операциями от двух источников— от TU мультитредовой части программы и от потоковой части через MSU. Это должно повысить развиваемую реальную производительность.

Управляемая локализация вычислений

Количество передач через коммуникационную сеть можно сократить, если вместо подкачки данных из удаленного узла просто переслать вычисление в узел, где находятся необходимые для него данные. Для этого используется механизм вызова процедуры на удаленном узле, который реализуется специальной машинной командой. При этом полагается, что во всех узлах загружена одна и та же программа. Эта команда имеет адрес процедуры (она есть во всех узлах) и ссылку на данные, с которыми она должна выполняться. В выдавшем эту команду узле происходит трансляция адреса данных и определяется узел, где эти данные находятся. Далее вызов процедуры передается именно в тот узел, где находятся эти данные.

Организация памяти

Для иллюстрации возможностей глобально адресуемой памяти МТП-суперкомпьютера имеет смысл провести сравнение с тем, как она организована в Cray XMT или более ранней Cray

MTA-2. Объем физически адресуемой памяти в МТП-суперкомпьютере равен 8 Пбайт (физическая адресация при переходе от Cray MTA-2 к Cray XMT Eldorado за пять лет увеличилась от 1 Тбайт до 256 Тбайт). Объем физически адресуемой локальной памяти узла МТП-суперкомпьютера равен 256 Гбайт (MTA-2 — 4 Гбайта, XMT — 32 Гбайта). Количество вычислительных узлов в МТП-суперкомпьютере — 32768 (MTA-2— 512, XMT— 8192). В системе XMT образца 2006 года максимальный размер сегмента данных по отношению к MTA-2 (2002 год) увеличился в сто тысяч раз— с 256 Мбайт до 32 Тбайт. Дополнительно размер быстрой таблицы дескрипторов (таблица TLB) в XMT был изменен так, что при использовании суперсегментов и таблицы такого размера удается перекрывать всю физически адресуемую память в 256 Тбайт. Очевидно, что за счет этого разработчикам удалось уйти от потерь производительности из-за возможных промахов обращений к TLB.

Все эти нововведения в архитектуре XMT говорят о том, что эта система действительно используется при решении задач, требующих объемов оперативной памяти, которые еще не так давно казались фантастическими. Ясно, что в МТП-суперкомпьютере должны были быть приняты адекватные меры, что и было сделано — введен размер суперсегмента до 256 Тбайт и перекрывающее всю память TLB в 2048 элементов.

В системах MTA-2 и XMT принята сегментная организация памяти с двумя уровнями адресации — виртуальными адресами задач и физическими адресами машины. Достоинство такой схемы — простота, которая оборачивается необходимостью использования сложных и заведомо неоптимальных алгоритмов централизованного отведения памяти под задачи. Поэтому в МТП-суперкомпьютере выбрана трехуровневая схема адресации, похожая на схему, применяемую в Cray T3E, Cray X1 и Cray BlackWidow [11].

При отображении виртуальной памяти на многомодульную физическую память обычно используют какую-либо схему «расслоения». Смысл ее в том, чтобы обеспечить для следующих друг за другом обращений к памяти попадание в разные модули. Если последовательно выдаваемые обращения к памяти имеют некоторую регулярность, что чаще всего происходит при обращении к элементам массива внутри гнезда циклов, то возможно попадание в один и тот же модуль, а это уже вызывает конфликт. Для снижения вероятности таких конфликтов обычно используется метод «зашумления» или «скремблирования» последовательно выдаваемых адресов. Такое скремблирование в MTA-2 выполняется всегда, а в XMT его можно отключать. В МТП-суперкомпьютере можно отключать скремблирование, однако здесь предусмотрен усиленный механизм управления отображением виртуальных адресов на модули памяти [12].

В программах, выполняемых МТП-суперкомпьютером, возможно использование физической и виртуальной адресации. Управление выдачей физических или виртуальных адресов производится путем установки специального бита в слове состояния треда. Физическая и виртуальная память адресуется до байта, однако основным рабочим объектом памяти является 64-разрядное слово (ячейка) базовый адресуемый элемент памяти, выровненный по 8-байтовой границе. Такая ячейка имеет теговые биты состояния, а ее адрес— теговые биты доступа. Выполнение операций доступа к памяти зависит от этих теговых битов, на этом построена низкоуровневая синхронизация при работе с памятью. Эти средства похожи на хорошо зарекомендовавшие себя средства, имеющиеся в MTA-2 и XMT.

Варианты и конфигурации микропроцессоров

В проекте «Ангара» предусмотрены два варианта реализации процессора МТП-суперкомпьютера — в виде микропроцессоров J7 (простой) и J10 (сложный, см. рис. 2). Первый поддерживает только гомогенную мультитредовую модель организации программ, работу с глобально адресуемой памятью и аппаратную обработку пакетов сообщений. Микропроцессор J10 дополнительно поддерживает гетерогенную мультитредовую модель, виртуализацию тредов и тредовых устройств, а также модели представления вычислений в виде статических и динамических графов потоков данных. Оба микропроцессора— заказные и зарубежных аналогов пока не имеют. Некоторые конфигурации

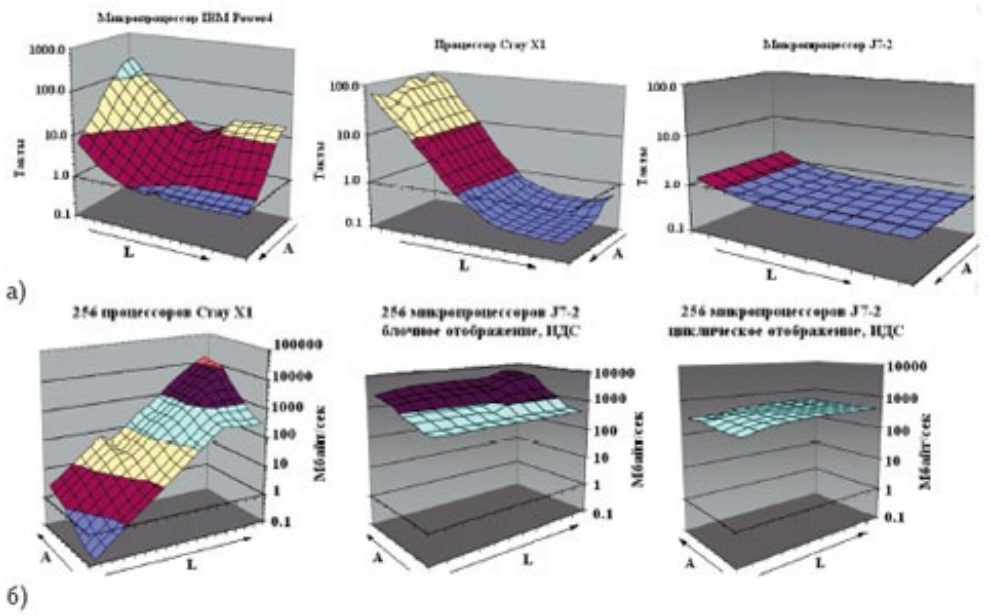
микропроцессоров J7 и J10 представлены в табл. 1.

Оценки реальной производительности

В проекте «Ангара» используется иерархическая методика оценочного тестирования, похожая на применяемую в программе DARPA HPCS [5].

На верхнем уровне методики используется тест APEX-MAP, позволяющий строить зависимость эффективности памяти от пространственно-временной локализации обращений к ней. Для одного процессора эффективность памяти представляется количеством тактов на одно обращение. Для мультипроцессорных конфигураций эффективность памяти измеряется ее пропускной способностью, приведенной к одному процессору. Для современных процессоров и суперкомпьютеров такие APEX-поверхности имеют форму «горки», что означает — обращения с плохой пространственно-временной локализацией выполняются крайне неэффективно. Это и есть основная причина низкой реальной производительности.

Главная цель проектов создания перспективных суперкомпьютеров состоит в повышении эффективности работы с памятью [5] — для таких суперкомпьютеров тест APEX-MAP должен строить плоские поверхности, расположенные на уровне лучших показателей, достигаемых при хорошей пространственно-временной локализации. На рис. 3 приведены APEX-поверхности для одного и для 256 процессоров J7-2 в сравнении с поверхностями, полученными для Cray X1 и для микропроцессора Power4.



Для мультипроцессорной конфигурации МТП-суперкомпьютера были получены две поверхности. Одна — для блочного отображения виртуальных адресов на физические, а другая — для циклического, и, как видно из рис. 3, их эффективность отличается на порядок, хотя обе имеют близкую к горизонтальной форму. Подъем поверхности для блочного распределения объясняется тем, что при улучшении временной локализации повышается эффективность использования кэш-памяти.

Опыт программирования разных оценочных тестов МТП-суперкомпьютера показывает, что полезны оба способа отображения. Блочное отображение удается особенно эффективно использовать для задач с хорошей и средней пространственно-временной локализацией. Основной прием эффективного программирования — организация на фоне счета подкачки данных в область глобально адресуемой памяти, отображенной на локальную память узла, и выполнение счета уже над подкачанными таким образом данными. Это похоже на подкачку в кэш-память обычных машин.

Следующий уровень оценочного тестирования — тесты набора HPC Challenge,

соответствующие крайним точкам APEX-поверхности: DGEMM — хорошая пространственная и временная локализация; RandomAccess — плохая пространственная и временная локализация; FFT — хорошая временная, но плохая пространственная локализация; STREAM — хорошая пространственная, но плохая временная локализация. Для одного узла результаты приведены в табл. 2 [12].

В табл. 3 приведены результаты тестирования для многопроцессорного варианта МТП-суперкомпьютера. В модели МТП-суперкомпьютера в данном случае использовалась потактовая модель реальной сети 4D-тор. Вместо теста DGEMM взяты тесты умножения матриц и HPL (высокоскоростной Linpack), которые имеют близкую хорошую пространственно-временную локализацию. Результаты реальной производительности на этих тестах можно сравнивать между собой.

Суперкомпьютер	MMULT (GFLOPS)	HPL (GFLOPS)	RandomAccess (GUPS)	FFT (GFLOPS)
X1E	Н/д	3190	0,0148	15,54
МТП / Ј7-2 / 4D тор	878	Н/д	18,56	258,56
МТП / Ј10-4 / 4D тор	26583	Н/д	~ 70	~ 3200

Приведенные результаты по оценочным тестам показывают возможность достижения поставленных задач в МТП-суперкомпьютере. Особенно следует обратить внимание на результаты теста RandomAccess— здесь имеется превосходство над обычными и даже векторными процессорами. С другой стороны, высокие результаты на тесте DGEMM показывают, что МТП-суперкомпьютер может эффективно решать и задачи с хорошей пространственно-временной локализацией.

Заключение

Проект «Ангара» был начат раньше, чем похожий по целям и решаемым задачам проект DARPA HPCS, и изначально эти проекты имели разную научно-техническую и инфраструктурную базу, несоизмеримо разные средства вкладывались в эти разработки и исследования. Бытует мнение, что практически значимые перспективные американские суперкомпьютеры появятся лишь к 2020 году, и поэтому сегодня можно не беспокоиться, однако это не так. Прототип СКСН в виде Cray BlackWidow будет уже в 2008 году [11], а Cray Baker— пилотная версия такого суперкомпьютера, годом позже. В одной и другой системе глобально адресуемая память приближается к одному Пбайт, причем уже известно (по ожидаемым показателям теста Random Access [11]), что с эффективным доступом.

В рамках отечественного проекта удалось реализовать принципы работы перспективного суперкомпьютера, которые программно реализованы, проверены и оценены в созданной параллельной имитационной модели. Модель в полном масштабе (все 32 тысячи узлов) имитирует работу создаваемого суперкомпьютера. Разработаны основные компоненты базового программного обеспечения инженерного уровня. Сейчас продолжаются исследования на тестовых наборах из новых областей приложений, создаются элементы системного программного обеспечения, включая библиотеки управления тредами, памятью, потоковыми моделями, блоки операционной системы. Вносятся необходимые коррекции в принципы работы. Разрабатывается микроархитектура основных блоков процессора и коммуникационной сети, ведется моделирование и макетирование. Начальный этап исследований завершен, и есть все предпосылки для развертывания опытно-конструкторских работ.

Литература

1. В.Митрофанов, А.Слуцкин, К.Ларионов, Л.Эйсымонт. Направления развития

отечественных высокопроизводительных систем. // Открытые системы, 2003, №5. www.osp.ru/os/2003/05/183021
2. T.Sterling, L.Bergman. A Design Analysis of Hybrid Technology Multithreaded Architecture for Petaflops Scale Computation. In Proceedings of International Conference on Supercomputing, June 20-25, 1999.
3. В.С.Бурцев. Система массового параллелизма с автоматическим распределением аппаратных средств суперЭВМ в процессе решения задачи. В сб. Вычислительные машины с нетрадиционной архитектурой. СуперЭВМ.Выпуск 2. М.: ВЦКП РАН, 1994г.
4. Cray/MTA Principles of Operation, Cray Inc., 28 November 2005.
5. А. Фролов, А. Семенов, А. Корж, Л.Эйсымонт. Программа создания перспективных суперкомпьютеров. // Открытые системы, 2007, №9.
6. W.J.Dally et al. Merrimac: Supercomputing with Streams. SC'03, November, 15–21, 2003.
7. W.D. Farwell, K.E. Prager. «Reconfigurable processor with alternatively interconnected arithmetic and memory nodes of crossbar switched cluster», United States Patent, Patent No: US 6,920,454 B2, Date of Patent: Jul.19,2005.
8. D.C. Burger, S.V. Keckler, K. Sankaralingman, R. Nagarajan. «Scalable processing architecture», United States Patent Application Publication, Pub.No: US 2005/0005084 A1, Pub.Date: Jan.6, 2005.
9. T. Sterling. «Architecture Paths to Exaflops Computing. Is Multicore the next Moor's Law? What about Memory?» Invited Presentation to the DOE E3SGS Town Hall Meeting. April 18, 2007.
10. R.Brightwell,K.T.Pedretti, K.D.Underwood, T.Hudson. SeaStar Interconnect: Balanced Bandwidth for Scalable Performance. IEEE Micro, May–June, 2006.
11. D.Abts, A.Batanieh, S.Scott, G.Faanes, J.Scwarzmeier, E.Lundberg, T.Jonson, M.Bye, G.Schwoerer, The Cray BlackWidow: A Highly Scalable Vector Multiprocessor. SC07, November 10–16, 2007.
12. А.Семенов, А.Соколов, Л.Эйсымонт. Архитектурные особенности и реализация глобально адресуемой памяти мультитредово-потокового суперкомпьютера. Подготовлена к печати, «Программирование», 2008.
13. И.Задыхайло, К.Ефимкин. Содержательные обозначения и языки нового поколения. // Информационные технологии и вычислительные системы, N 2, 1996.

Анатолий Слуцкин *(slutskin@nicevt.ru)— заместитель генерального директора ОАО «НИЦЭВТ», **Леонид Эйсымонт** (verger@nicevt.ru)— начальник отдела ОАО «НИЦЭВТ» (Москва).*

Программирование СКСН

Среда программного обеспечения СКСН делится на три уровня.

Первый уровень— инженерный, это создание на языках Фортран и Си с использованием библиотечных функций для явного программирования программ, использующих мультитредовые, потоковые и другие модели организации вычислений.

Второй уровень— уровень технологий параллельного программирования, при котором специфические для МТП-суперкомпьютера модели вычислений используются неявно, для реализации типичных средств параллельного программирования: MPI, Shmem, OpenMP, UPC, Co-Array-Fortran и Charm++.

Третий уровень— это уровень неявного параллельного программирования с использованием более высоких и непроцедурных понятий по сравнению с ныне принятыми. Для СКСН этот уровень должен по продуктивности отличаться от существующего на порядок, например, в DARPA HPCS такие надежды связываются с языками Chapel, X10 и Fortress, однако неизвестно, как они будут приняты программистским сообществом. Из отечественных разработок примером языка такого типа является Норма [13].

30.11.2007г.

Постоянный URL статьи: http://www.osp.ru/os/2007/09/4569294/

© 1992-2011 Все права защищены. Издательство "Открытые системы"