

НУ И ЧТО ЖЕ НАМ ТЕПЕРЬ ДЕЛАТЬ С ЭТИМ МНОГООБРАЗИЕМ СУПЕРКОМПЬЮТЕРНЫХ МИРОВ? ПОПЫТКА КОНСТРУКТИВНОГО СИСТЕМАТИЧЕСКОГО ПОДХОДА К СОПОСТАВЛЕНИЮ СУПЕРКОМПЬЮТЕРНЫХ АРХИТЕКТУР

С.С. Андреев, С.А. Дбар, А.О. Лацис, Е.А. Плоткина, Г.П. Савельев

Институт прикладной математики им. М. В. Келдыша РАН

Введение.

Обсуждение экзафлопсной проблемы в ее первоначальной постановке (не более 20 МВт, не позднее 2018г.) сделало очевидными серьезнейшие проблемы, которые возникают на пути эволюционного развития традиционной однородно-многоядерной архитектуры суперкомпьютеров [1]. Сегодня в этом уже никого не требуется специально убеждать. Новые архитектуры очень нужны, причем «нужны вчера». Новость последних примерно двух лет заключается в том, что мы и сегодня, как несколько лет назад, когда проблема новых архитектур только возникла, понятия не имеем, какими именно эти новые архитектуры могли бы быть.

За эти годы мы прошли этап эйфории от появления GPGPU, разочарования сложностью и вычурностью их программирования, и несколько лет ждали появления Xeon Phi, который, наконец-то, эту сложность преодолет, позволит без всей этой кошмарной зауми и ухищрений ускорить старый программный код хотя бы в несколько десятков раз. Сегодня мы знаем, что чуда не произошло. GPGPU во многом не оправдали возлагавшихся на них ожиданий. Xeon Phi, если рассматривать его как альтернативу GPGPU, позволяющую радикально упростить программирование, похоже, не оправдал их ни в чем [2]. Мы теперь гораздо лучше, чем еще пару лет назад, понимаем, что по-старому жить нельзя, но еще лучше мы понимаем, что жить по-новому тоже не получается. Все придется начинать с самого начала.

Если готовых ответов пока не найдено, то тем более важно привести в порядок вопросы, на которые предстоит ответить. Этому, в основном, и посвящен настоящий доклад.

1. Что мы твердо знаем о проблематике новых суперкомпьютерных архитектур.

1). Новые суперкомпьютерные архитектуры, пригодные на практике, могут быть созданы только совместными усилиями трех категорий специалистов отрасли: прикладных программистов, системных программистов и схемотехников (инженеров по «железу»). Все три перечисленные категории специалистов объединяет одно фундаментально общее качество: подход к компьютеру на том или ином логическом уровне абстракции. Этим все они в равной степени отличаются, например, от специалистов по микроэлектронике, для которых компьютер есть сущность физическая, а не логическая. Фундаментальные различия между тремя упомянутыми категориями специалистов формулируются в терминах традиционной архитектуры (схемотехник – это тот, кто «внутри» процессора, программист – тот, кто «снаружи»). Если мы хотим уйти далеко от традиционного процессора, создав совсем новую архитектуру, то привычное за многие годы разделение на три корпорации по признаку отношения к традиционному процессору обречено исчезнуть. На время придется всем снова стать универсалами, и научиться разговаривать на общем, равно понятном всем, языке, который для этого предстоит создать.

2). Проблема новой архитектуры – это не только и не столько проблема «железа», сколько проблема алгоритмов и численных методов. Архитектура есть конкретный способ организации перемещения данных между вычислительными устройствами и устройствами памяти. Традиционная архитектура, грубо говоря, реализует простейший из таких способов: равную удаленность всех элементов данных от всех устройств обработки. В этом ее простота и привлекательность, в этом же ее неэффективность, в этом же – надежда на то, что ее можно улучшить, создав что-то новое: улучшить можно только плохое, идеальное улучшить нельзя. Оптимизировать перемещение данных между функциональными устройствами вычислителя можно только путем создания условий для локализации максимального объема обработки на минимальном объеме памяти [3]. Но эту проблему, вообще говоря, нельзя решить в достаточной мере на уровне «железа». Ее надо решать на уровне даже не алгоритма и программы, а уже на уровне численного метода. В создании вычислителя, способного воплотить локализацию обработки в быстрое и энергоэффективное выполнение расчетов, нет ничего невозможного. Напротив, создание вычислителя, который волшебным образом придумает за программиста, как эту локализацию обеспечить, вставив в текст старой программы пару-тройку директив, вряд ли стоит рассматривать как работоспособную альтернативу.

3). На нынешнем этапе архитектурного кризиса вряд ли следует рассчитывать на появление новых решений исключительно в сфере промышленности изделий массового выпуска, как это было когда-то в случае кластеров и совсем недавно – в случае GPGPU. Стоящие перед нами сегодня задачи слишком сложны и слишком специфичны именно для суперкомпьютерной отрасли. Иными словами, «новых, хороших GPGPU/Xeon Phi» мы не дождемся, дело надо брать в свои руки. Какой-то части специалистов отрасли придется

обратиться к технологиям программируемой логики, чтобы на практике, буквально своими руками, создавать новые вычислители и/или их действующие модели. Это единственный на сегодня способ перевести решение проблемы в режим «народной стройки», резко расширить как качественно, так и количественно круг вовлеченных в поиски решения. Без этого нам, похоже, с места не сдвинуться. В том, как конкретно это сделать, есть масса сложностей, вопросов и ответов на них, но это могло бы быть темой отдельного доклада.

2. Чего мы пока не знаем до конца, и как нам это узнать.

В предыдущем разделе мы показали, что важнейшее условие успеха в поиске новых архитектур – междисциплинарный диалог, совместный анализ численных методов и «железа». Но такой анализ невозможен без соответствующего измерительного инструмента. Как воздух необходима система тестовых модельных приложений, к которой предъявляются очень жесткие требования, а именно:

1). Она должна быть конструктивной, то есть иметь прямое отношение к реальной жизни, и требовать такой техники параллельной реализации, которая была бы непосредственно пригодной для реальных задач. Иными словами, демонстрировать быстродействие только в сочетании с инженерной методикой его достижения, а не «в принципе», как HPL или другие синтетические тесты.

2). Она должна быть не просто набором, а именно системой тестов, обладать внутренней стройностью, систематически охватывать ту или иную (те или иные) область (области) актуальных приложений.

3). Как техника параллельной реализации тестовых приложений, так и место именно этой системы тестов в общей суперкомпьютерной картине мира, должны быть равно понятны всем трем категориям суперкомпьютерных специалистов, упомянутым в предыдущем разделе. Схемотехник должен взять на себя труд понимания того, чем именно индексная сетка отличается от неструктурированной на уровне алгоритма и метода, равно как и прикладной программист должен заставить себя осознать, что может, а что не может дать расслоение памяти, и почему не всякий конвейер можно сделать произвольной ширины. Система тестов должна всемерно этому способствовать.

4). Помимо собственно набора тестовых приложений, система тестов должна содержать способ передачи другим людям тех знаний и умений, которые требуются для реализации этих тестов на нетрадиционных архитектурах, в том числе – тех, которым еще только предстоит появиться. Эффективная реализация приложений, например, на гибридно-параллельных машинах – дело очень сложное и трудоемкое. Описание этих реализаций в терминах, скажем, псевдокода явно недостаточно, а описание в терминах конкретной технологии (скажем, OpenCL) слишком сложно для прикладного программиста, и во многом бесполезно при переходе на другой тип ускорителя.

В докладе предложен некоторый конкретный способ решения перечисленных проблем, возможно, не идеальный, но работоспособный в первом приближении. Так, систему тестов предлагается строить путем систематизации итерационных методов решения СЛАУ с разреженной матрицей, причем тестов будет всего четыре (в докладе объясняется, почему). Идея именно такой системы тестов, конечно, подсказана тестом HPCG [4], и сам тест HPCG вошел в эту систему с минимальными изменениями как один из четырех.

В качестве инструмента накопления и передачи знаний о технике гибридно-параллельной реализации предлагается использовать модельную библиотеку методов и инструментов (в данном случае – методов и инструментов решения СЛАУ с разреженной матрицей итерационными методами). Любая доступная в исходных текстах библиотека готовых параллельных «решателей» и «строительных блоков» для построения своих собственных проходит в своем развитии эту «модельную» фазу, но на ней не останавливается. Постепенно разрастаясь и усложняясь, библиотека совершенствуется как «черный ящик», но теряет внутреннюю простоту и понятность первоначальных версий, делающую ее пригодной в качестве как учебника, так и исходной точки для написания собственных программ. Предлагаемая идея состоит в том, чтобы построить реализации модельных приложений как приложения, разработанные с помощью иерархически структурированной библиотеки, разработанной для поддержки создания приложений именно такого рода. Саму библиотеку предполагается ни в коем случае не усложнять и не наращивать, а сознательно «заморозить» на модельной стадии развития, чтобы она оставалась пригодной для обучения методом ведения за руку. Отдаленным прототипом такой библиотеки могло бы служить подмножество самых первых, ныне не поддерживаемых, версий PETSc [5].

3. Место технологий программируемой логики в предлагаемом плане действий.

Сделанное выше предложение обратиться к технологиям программируемой логики для создания принципиально новых суперкомпьютеров своими руками звучит почти столь же устрашающе, сколь и бессодержательно. Необходимы уточнения. Перечислим конкретные направления возможных работ в этом направлении.

Начать следует с самого простого – с гибридно-параллельных кластеров, сопроцессоры которых выполнены на ПЛИС. В докладе отмечается и анализируется фундаментальное свойство таких сопроцессоров – малый объем внутренней синхронной памяти, пригодной для высокоэффективной реализации конвейерных вычислений.

Использование сопроцессоров на ПЛИС возможно двумя способами: или для прямой схемной реализации вычислений, или же для создания программируемых вычислителей нетрадиционной архитектуры. На втором направлении наиболее интересно было бы проверить идею мультитредового процессора с легкими

ядрами [6]. Также не лишена интереса идея построения программируемого вычислителя по идеологии VLIW. Заметный успех на любом из этих двух направлений мог бы значительно упростить техническую сторону адаптации приложений к вычислителям новой архитектуры.

Что же касается идеи прямой схемной реализации вычислений, то она была и остается потенциально наиболее многообещающей с точки зрения эффективности, в том числе – энергоэффективности. Главным препятствием в получении на этом пути практически значимых результатов является вовсе не сложность программирования ПЛИС, как это принято считать, а недостаточная проработанность проблемы с точки зрения локализации вычислений на малом объеме внутрикристалльной памяти [3], то есть проблема опять-таки не «железная», а методико-алгоритмическая. Что же касается технологии программирования ПЛИС как таковой, то в докладе приводятся некоторые сведения на этот счет, а также первые результаты практической оценки выпущенной примерно год назад системы программирования ПЛИС на Си от Xilinx.

ЛИТЕРАТУРА:

1. В.С. Горбунов, Г.С. Елизаров, Л.К. Эйсымонт. Обзор проектов экзафлопсных суперкомпьютеров за рубежом и в России, ограничения и перспективы роста». Презентация доклада на 4-м Московском суперкомпьютерном форуме (МСКФ-2013). <http://www.ospcon.ru/files/media/Eysymont.pdf>. Дата обращения 31.03.2014
2. В.Т. Жуков, М.М. Краснов, Н.Д. Новикова, О.Б. Феодоритова. Параллельный многосеточный метод: сравнение эффективности на современных вычислительных архитектурах. Препринт ИПМ РАН №31, Москва, 2014.
3. С.С. Андреев, С.А. Дбар, А.О. Лацис, Е.А. Плоткина. Макет гибридной реконфигурируемой вычислительной системы и реализация на нем вычислений с повышенной точностью. Тезисы докладов 2 национального суперкомпьютерного форума НСКФ-2013, Переславль-Залесский, 2013.
4. Jack Dongarra, Michael A. Heroux. Toward a New Metric for Ranking High Performance Computing Systems. Sandia Report. SAND2013-4744. Printed June 2013. <http://www.sandia.gov/~maherou/docs/HPCG-Benchmark.pdf>
5. <http://www.mcs.anl.gov/petsc/> Дата обращения 31.03.2014.
6. В.С. Горбунов, С.Г. Елизаров, В.В. Корнеев, А.О. Лацис. Футурология суперкомпьютеров. «Суперкомпьютеры» №1(17)весна 2014. с. 24-28.