Достижение рекордных показателей в GreenGraph500 для вычислительных систем на ПЛИС. Теория и практика

А.Д. Сизов, С.Г. Елизаров

Московский государственный университет имени М.В. Ломоносова

Современные мировые достижения в области разработки энергоэффективных программируемых логических схем (ПЛИС), обширный опыт применения реконфигурируемых спецвычислителей при разработке проблемно ориентированных суперкомпьютеров и уже продемонстрированные возможности создания на ПЛИС контроллеров памяти и коммуникационных процессоров со сверхнизкой латентностью, позволяют предполагать, что именно на такой элементной базе сегодня могут быть созданы вычислительные системы с рекордными на тесте GreenGraph500 показателями. В работе обсуждаются требования к вычислительной системе на ПЛИС с внешней памятью применительно к решению задачи поиска вширь по графу (Breadth first search – BFS), учитывающие имеющийся мировой опыт и особенности лучших существующих параллельных алгортмов BFS. Рассмотрен реальный вычислительный узел, содержащий ПЛИС Kintex Ultra Scale с 4-мя контроллерами памяти RLDRAMIII. Оценена производительность системы из 32-двух таких узлов, рассчитана энергоэффективность по критериям рейтинга GreenGraph500 и даны рекомендации по дальнейшей оптимизации аппаратуры.

1. Введение

Graph500 — мировой рейтинг суперкомпьютеров, предназначенных для решения задач, связанных с обработкой больших графов. Для ранжирования этих систем используется BFS – поиск в ширину в неориентированном разреженном графе. Этот тест в большей степени нагружает коммуникационную подсистему и контроллеры памяти, так как данный алгоритм подразумевает работу с большим объемом нерегулярных данных в противоположность Top500, ориентированному на вычисления над числами с плавающей точкой на тесте HPL Linpack. В дополнение к Top500 очень востребован Green500 – рейтинг энергоэффективности вычислительных систем на тесте Linpack. Предложенный в 2012 году GreenGraph500, сочетает указанные выше подходы и ранжирует системы из Graph500 по производительности в GTEPS (10^9 пройденных дуг в секунду) на Ватт электропотребления. Важность этого теста сложно переоценить, так как именно энергоэффективность и скорость работы со сверхбольшими объемами нерегулярных данных являются основными требованиями к суперкомпьютерам и центрам обработки данных будущего [1].

Современный опыт показывает, что один из наиболее удачных подходов к построению заказных проблемно ориентированных вычислительных систем (ПОВС) максимальной энергоэффективности – использование специальных ускорителей на базе программируемой логики (ПЛИС) [2]. С другой стороны, критическим фактором, ограничивающим производительность при решении графовых задач, является скорость случайного доступа в память. Показано [3], что производительность традиционных CPU/GPU архитектур, ориентированных на блочную работу с внешней памятью и использующих глубокие конвейеры команд в совокупности с несколькими ступенями кеширования данных, снижается на 1-2 порядка на задачах типа BFS. Однако на ПЛИС возможно реализовать специализированные контроллеры памяти, практически лишенные указанного недостатка, такие как, например, в системе Convey MX-100, входящей в первую сотню рейтинга Graph500 [4]. Производительность подсистемы памяти можно еще увеличить, перейдя к отличным от DDR3/DDR4 архитек-

турам [6]. Другой подсистемой, определяющей производительность в задаче BFS, является коммуникационная сеть, соединяющая вычислительные модули [7]. Известно, что наиболее быстрые и низколатентные коммуникационные сети для ПОВС на ПЛИС построены на мультигигабитных трансиверах и коммерчески доступных коммутаторах PCIe [8].

Предлагая ПЛИС, в качестве основного вычислительного узла, нужно принимать во внимание известные недостатки ПЛИС относительно узлов на основе CPU/GPU: номинальная рабочая частота ПЛИС составляет 300-600 МГц, которая в 5-10 раз уступает рабочей частоте современных коммерческих процессоров. Объем быстрой памяти, расположенной непосредственно на кристалле ПЛИС, ограничен 1-10 МБайт, что не позволяет использовать ПЛИС для решения задач большого размера без применения внешней памяти. Цена топовых ПЛИС на порядок превышает цену соответствующих CPU/GPU. Кроме того, создание ПОВС на базе ПЛИС предполагает для каждой конкретной задачи создание и отладку вычислителя на языке описания аппаратуры, сложность которой на порядок выше написания программы под традиционные архитектуры на языках высокого уровня.

В настоящей работе проводится анализ литературы и требований к аппаратной базе ПОВС для построения топовых решений в GreenGraph500. Выполняется расчет параметров оптимальной конфигурации, даются рекомендации для создания ПОВС для графовых задач различного размера. Проводится анализ применимости разработанного для данного ПОВС алгоритма поиска вширь. В рамках данной работы предполагается определение производительности одного узла на алгоритме BFS с помощью моделирования работы реального ПЛИС.

2. Общая память

Как сказано выше, BFS предполагает множество случайных обращения в общую память всей вычислительной системы. В работе [3] показано, что пиковая производительность контроллеров памяти в традиционных CPU/GPU архитектурах, рассчитанных на блочное чтение, достигается только при работе с большими 4 КБ и более блоками данных и снижается на порядки при чтениях отдельных машинных слов. Размер обрабатываемых алгоритмами BSF графов лежит в диапазоне от ГБ до ПБ, при том, что каждый запрос на чтение в BFS оперирует единицами машинных слов (4/8 байт на слово), адреса запросов практически случайны, поэтому эффективное чтение большими блоками невозможно. Таким образом, архитектура контроллера памяти в классических CPU/GPU архитектурах является фактором, ограничивающим общую производительность системы на тесте BFS. Это позволяет полагать, что переход к проблемно-ориентированным контроллерам памяти, на которых возможно достижение максимальных пропускных способностей на операциях доступа по случайным адресам, является одним из перспективных направлений в создании ПОВС для графовых задач.

Увеличение производительности подсистемы памяти возможно также при использовании других типов ОЗУ, так например производительность RLDRAMIII (Reduce latency DRAM) на случайных чтениях в 2-3 раза больше, чем для соответствующей DDR3. [6]

3. Коммуникационная подсистема

В статье [7] показано, что в вычислительных системах с многими узлами производительность алгоритма BFS определяется коммуникационной подсистемой, обеспечивающей обмен данными между вычислительными узлами, поэтому снижение количества пересылаемых данных позволяет значительно повысить производительность системы в целом. В работе [8] показана возможность построения и эффективной масштабируемости системы из нескольких ПЛИС, в которой коммуникационная подсистема построена на базе мультигигабитных трансиверов. Коммерчески доступной сетью такого типа является пакетная

сеть PCIe с топологией типа "звезда", которая в рамках стандарта PCIe Gen3 позволяет достигать пропускной способности до 16 ГБайт/с в дуплексном режиме.

4. Проект BFS для ПОВС

```
1.  for(i = 0; i < size(V); i++)
2.      lvl[v] = Inf;
3.  lvl[s] = 0;
4.  write_to_bfs_queue(n,s);   // write v to queue on chip n
5.  //On every chip, on every level:
6.  while(Q is not empty)
7.      for (all u in Q) //1 read
8.          for(all v in CSR[u])//3 reads
9.              if (v located in local_mem)
10.                 if (lvl[v] > lvl[u]) //2 reads
11.                     d[v] = u; //write
12.                     lvl[v] = lvl[u]; //write
13.                     //add v into local queue
14.                     write_to_bfs_queue(local,v);
15.             else
16.                 // send remote check request
17.                 write_to_check_queue(n,v);
```

Рис. 1. Проект распределенного алгоритма поиска вширь на ПОВС

Для ПОВС на ПЛИС требуется мультитредовый алгоритм, в котором разрешены только локальные чтения, операции глобальной синхронизации не требуют большого количества пересылок и в максимальной степени используются возможности ПЛИС и подсистемы памяти. Проект такого аглоритма приведен на рис. 1. Изначально, вершины в графе разбиваются между узлами таким образом, что ребра, соответствующие списку вершин, обрабатываемых на данном узле, находятся в локальной памяти соответствующего ПЛИС. В памяти каждого узла также хранится также локальный участок фронта. В качестве формата хранения графа используется Compressed Sparse Row (CSR) формат. Каждый локальный фронт на определенном уровне поиска обрабатывается независимо, причем, если в процессе поиска обрабатываемое ребро связывает локальную вершину с вершиной, данные о которой хранятся в удаленной памяти, информация о данной вершине посылается на удаленный вычислительный узел, где и происходит ее последующая обработка. Разрешение конфликтов между потоками внутри узла вычислителя осуществляется с помощью аппаратно реализованных на уровне контроллера памяти атомарных операций и full/empty признаков ячеек данных.

5. Оценка производительности ПОВС на ПЛИС

5.1. Оценка производительности узла

Предлагаемый ПОВС состоит из 32 вычислительных узлов, соединенных коммуникационной подсистемой из мультигигабитных трансиверов работающих по протоколу PCIe Gen3 4x. Каждый вычислительный узел представляет из себя кристалл ПЛИС Kintex Ultrascale XCKU095 емкостью 940 тыс. LUT, работающий на частоте 660 МГц, и четыре контроллера внешней памяти RLDRAMIII, работающий на частоте 800 МГц, емкостью 64 Мбайт каждый. Оценку производительности данного вычислительного узла будем проводить путем сравнения с существующими вычислительными системами на ПЛИС от компании Convey, производительность которых известна [4]. Система Convey MX-100 состоит из четырех вычислительных кристаллов V6 HX565T емкостью 585 тыс. LUT, работающий на частоте 550 МГц и подсистемы памяти из 32 каналов DDR3.

В работе [9] было показано, что использование алгоритма оптимизации по направлениям позволяет снизить количество обрабатываемых ребер графа до размера минимального остовного дерева, или в 16 раз для графа плотностью 16 ребер на вершину. В работе [4] используется способ хранения, который позволяет уменьшить количество запросов на чтение данных при обработке одного ребра до трех, что позволяет оценить требуемую пропускную способность памяти в системе MX-100 по формуле 14,6 [5] /16*3 = 2,7 GR/s (GR/s – 10^9 чтений в секунду). В соответствии с работой [6] производительность используемой в системе Convey памяти DDR3 на случайных чтениях может быть оценена в 0,6 GR/s для 32-х каналов. Рассматриваемое в [10] переупорядочивание запросов при доступе к памяти позволяет повысить производительность чтения в 4-5 раз относительно скорости случайного чтения. Проведенная нами моделирование работы контроллера RLDRAMIII показало, что производительность предлагаемой подсистемы памяти составляет 140*16 = 2,24 GR/s на случайных чтениях и 750*16 = 11,5 GR/s на последовательных чтениях при размере блока 18 байт, что позволяет говорить о создании подсистемы памяти с пропускной способностью до 10 GR/s.

Тогда, отталкиваясь от производительности подсистемы памяти, предлагаемый ПОВС сможет обрабатывать в 10/2,7 = 3,7 больше ребер в секунду, чем MX-100. Следовательно, производительность одного ПЛИС, предполагая линейную масштабируемость вычислителя, можно оценить в (14,6/4)*940 тыс. LUT*660 МГц/565 тыс. LUT*550 МГц = 7,3 GTEPS.

5.2. Возможности масштабируемости системы

В разделе 4 было показано, что в случае обработки ребра, которое соединяет локальную вершину с вершиной, находящейся в удаленной памяти, по коммутационной шине посылается запрос на удаленную обработку данной вершины. Этот запрос предполагает передачу 8 байт полезной информации – номер запрашиваемой вершины, номер запрашивающей вершины и ее уровень. При равномерном распределении вершин между узлами, учитывая максимально возможную производительность одного узла, оценка для которой дана в предыдущем разделе, необходимую пропускную способность можно рассчитать как 8 байт*(7,6 GTEPS/16) = 3,8 ГБайт/с для системы с достаточно большим кол-вом вычислительных узлов. В предлагаемом ПОВС для соединения вычислителей используется сеть, построенная на коммутаторах PCIe Gen3 4x, пропускная способность которой на запись из одного вычислителя в другой составляет 4 ГБайт/с. Однако, известно [11], что пропускная способность PCIe при передаче сообщений величиной 8 байт составляет приблизительно 30% от максимальной, более 90% при длине сообщения в 100 и более байт. Это значит, что для полноценной загрузки ПЛИС потребуется либо перейти к шине PCIe большей ширины, либо использовать механизм агрегации сообщений удаленной записи. Эти оптимизации позволят для системы из 4 узлов достичь практически линейной масштабируемости. Однако в рассматриваемом прототипе 4-х узловые блоки объединены PCIe Gen3 8x, проведя аналогичные выкладки, производительность ПОВС с 32 узлами может быть оценена в 7,6 GTEPS * 32 узла*(8 Гбайт/с/4) /3,8 Гбайт/с = 128 GTEPS или 200 MTEPS/W (оценивая энергопотребление в 20 Ватт на ПЛИС).

5.3. Сравнение с существующими устройствами

Рассматриваемая ПОВС по размеру решаемой задачи BFS должна быть по отнесена по классификации GreenGraph500 к разделу Small data, который в редакции от июля 2015 представлен в первой десятке 4-мя системами оригинальной архитектуры и 6-ю системами на базе микропроцессоров для сотовых телефонов и планшетов. Второй и третий десяток лидеров GreenGraph500 практически полностью заняты SMP системами на одном, двух или четырех топовых x86 процессорах Intel Sandybridge. Практически все лидеры используют предельно оптимизированные алгоритмы от группы GraphCREST [9]. Попадание в десятку

Таблица 1. Сравнение рассматриваемого ПОВС с существующими решениями.

Параметры	**Convey MX-100 [12]**	**Xperia Z1 [13]**	**Fermi GPU [14]**	**Cray XE6 Hopper [15]**	**Intel SB EP [13]**	**ПОВС 32 узла**
Возможный размер графа	29	20	20	31	28	22
Результат GTEPS	14,6	1,03	0,63	62	28,61	128
Результат MTEPS/W	146	235	2,6	0,15	61,48	200
Green Graph500 Small or Big Data Category	8 SD	2 SD	29 SD	19 BD	1 BD	5 SD
Graph500	79	153	171	54	70	46

требует энергоэффективности на уровне 130 MTEPS/W, которая, как показано выше, может быть достигнута на ПОВС в рассматриваемой в настоящей статье конфигурации при эффективности реализации аглоритма BFS на ПЛИС ПОВС на уровне систем Convey.

Отметим, что практически все представленные в разделе Small data вычислители имеют один узел и не допускают масштабирования, т.к. либо принципиально однопроцессорные (системы на Snapdragon и подобные), либо используют не масштабируемые решения (SMP система на 4-х Intel Sandybridge). В противоположность им – предложенная ПОВС многоузловая и уже содержит 32 вычислительных узла. Ее отнесение к Small data связано только с особенностью используемой RLDRAMIII (малая емкость модуля). В классе Big data попадание в первую десятку требует энергоэффективности на уровне 20 MTEPS/W, которая очевидно будет достигнута при переходе на более емкие модули памяти типа DDR3/4.

6. Заключение

Широкий класс задач, требующих нерегулярной работы с большими и сверхбольшими объемами данных, в том числе задача поиска вширь по графу, может эффективно решаться на массиве ПЛИС, оснащенных контроллерами памяти и связанными коммуникационной шиной PCIe. При совместной оптимизации алгоритма поиска, количества и типа используемых контроллеров памяти и параметров коммуникационной шины, могут быть разработаны системы с рекордными показателями в тесте GreenGraph500.

Литература

1. Francesquini, Emilio and Castro et al. //On the energy efficiency and performance of irregular application executions on multicore, NUMA and manycore platforms, Journal of Parallel and Distributed Computing, 2014, Elsevier.

2. Francisco, Phil and others //The Netezza data appliance architecture: a platform for high

performance data warehousing and analytics, IBM Redbooks, 2011.

3. Agarwal, Virat and Petrini, Fabrizio and Pasetto, Davide and Bader, David A //Scalable graph exploration on multicore processors, Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis, P. 1–11, 2010, IEEE Computer Society

4. Attia, Osama G and Johnson, Tyler and Townsend, Kevin and Jones, Philip and Zambreno, Joseph //CyGraph: A Reconfigurable Architecture for Parallel Breadth-First Search, Parallel & Distributed Processing Symposium Workshops (IPDPSW), 2014 IEEE International, P. 228–235, 2014, IEEE.

5. Graph500 List July 2015, URL: http://www.graph500.org/results_jul_2015

6. Avnet //Optimal Memory Interface Design with Xilinx 7 Series Xfest-2012 presentation, 2012, URL: http://www.em.avnet.com/en-us/design/trainingandevents/Documents/X-FEST%202012%20PRESENTATIONS/xfest12_pdf_memory_v1_2_may15.pdf

7. Checconi, Fabio and Petrini, Fabrizio //Traversing Trillions of Edges in Real Time: Graph Exploration on Large-Scale Parallel Machines, Parallel and Distributed Processing Symposium, 2014 IEEE 28th International, P. 425–434, 2014, IEEE.

8. Theodore Markettos, A and Fox, Paul J and Moore, Simon W and Moore, Andrew W //Interconnect for commodity FPGA clusters: standardized or customized?, Field Programmable Logic and Applications (FPL), 2014 24th International Conference on, P. 1–8, 2014, IEEE.

9. Yasui, Yuichiro and Fujisawa, Katsuki and Goto, Keisuke //NUMA-optimized parallel breadth-first search on multicore single-node system, Big Data, 2013 IEEE International Conference on, P. 394–402, 2013, IEEE.

10. Jin, Zheming and Bakos, Jason D //Memory Access Scheduling on the Convey HC-1, 2013 IEEE 21st Annual International Symposium on Field-Programmable Custom Computing Machines

11. Understanding Performance of PCI Express Systems Xilinx White paper October 2014, URL: http://www.xilinx.com/support/documentation/white_papers/wp350.pdf.

12. Convey //Convey MX Series Architectural Overview, White paper, URL: http://www.conveycomputer.com/files/5913/5266/3278/CONV-12-036.1MXarchOvrvwWeb.pdf

13. Yasui, Yuichiro and Fujisawa, Katsuki and Sato, Yukinori, //Fast and energy-efficient breadth-first search on a single numa system, Supercomputing, P. 365–381, 2014, Springer.

14. Hong, Sungpack and Oguntebi, Tayo and Olukotun, Kunle, //Efficient parallel graph exploration on multi-core CPU and GPU, Parallel Architectures and Compilation Techniques (PACT), 2011 International Conference on, P. 78–88, 2011, IEEE.

15. Beamer, Scott and Buluc, Aydin and Asanovic, Krste and Patterson, Dean, //Distributed memory breadth-first search revisited: Enabling bottom-up search, Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2013 IEEE 27th International, P. 1618–1627, 2013, IEEE.