ВЫЧИСЛИТЕЛЬНАЯ МОДЕЛЬ ЛОКАЛЬНО-РЕКУРСИВНЫХ НЕЛОКАЛЬНО АСИНХРОННЫХ АЛГОРИТМОВ ДЛЯ ГЕТЕРОГЕННЫХ СИСТЕМ

И.А. Горячев, В.Д. Левченко

На современном этапе развития высокопроизводительных вычислительных систем существенная роль отводится графическим ускорителям. Использование GPU позволяет существенно повысить производительность и эффективность вычислений. Однако ключевым фактором разработки программ и приложений для гибридных систем остается реализация высокопроизводительных алгоритмов.

Для задач моделирования физических процессов конечно-разносными методами такими алгоритмами являются локально-рекурсивные нелокально-асинхронные алгоритмы (LRnLA) [1, 2], разрабатываемые на базе института ИПМ им. Келдыша РАН. При внедрении LRnLA алгоритмов на гетерогенные системы важно помнить о тонкостях, связанных с доступом к памяти, с разными уровнями параллелизма GPU как в программной, так и в аппаратной реализациях, чтобы получить высокопроизводительный код.

Связь между иерархической структурой подсистемы памяти и различными уровнями параллелизма можно отразить с помощью вычислительной модели для гетерогенных систем, создание которой является целью данной работы. Это позволит корректно охарактеризовать необходимые вычислительные ресурсы — время выполнения, объём памяти, а также ограничения алгоритмов или компьютера — и выбрать верную стратегию в разработке LRnLA алгоритмов для гетерогенных систем.

LRnLA алгоритмы базируются на парадигме “разделяй-и-властвуй” — решаемая задача — определенный объем данных и связанные с ним операции — разбивается на подзадачи меньшего размера. Разбиения выполняются до тех пор, пока все подзадачи не окажутся элементарными. При этом доступны разные способы разбиения, в классификации LRnLA алгоритмов обозначаемые как классы, формы и виды[1]. В совокупности это даёт многопараметрическое множество способов разбиения на подзадачи.

Подбор оптимальных параметров разбиения конкретной задачи для конкретной вычислительной системы будем вести в модели «песочных часов». Это позволит нам сохранить универсальность выбора как решаемых задач, так и вычислительных систем, в том числе и с разной архитектурой. С одной стороны «песочных часов» располагается множество всех решаемых задач, с другой — все доступные вычислительные ресурсы. При этом выбранная задача согласовывается с графом зависимостей LRnLA, а вычислительная система согласовывается с моделью вычислений.

«Узким горлышком песочных часов» и одновременно связью между графом зависимостей и моделью вычислений будет граф разбиения — ориентированный граф, под вершинами которого подразумевается выполнение подзадачи несколькими однородными вычислительными элементами, а под направленными ребрами — декомпозиция данных и связанных с ними операций с последующим переходом к подзадаче меньшего размера. Граф разбиения вложим в трехмерное пространство, где по одной оси откладывается объем обрабатываемых данных, по второй оси — коэффициент локальности — отношение числа операции к числу данных, по третьей оси — степень параллелизма. Каждой вершине назначается своя координата, показывающая число независимых подзадач одного типа, объем связанных с ними данных и количество операций, которые проводятся с этими данными.

Для примера приводится кластерная система К-100, установленная в ИПМ им. М.В. Келдыша РАН. На обозначенном гибридном кластере К-100 в распоряжении имеются до 64 узлов. На одном узле находится 3 GPU устройства, 2 CPU Intel Xeon X5670 и 96 ГБайт DDR3 SDRAM. GPU устройство имеет марку Tesla 2050 из архитектуры Fermi, на котором присутствуют 16 мультипроцессоров, 3 ГБайт глобальной памяти, и позволяет запускать до 1024 программных тредов на одном мультипроцессоре(SM). Выполнение программных тредов тесно связано с аппаратными полуварпами (warp), состоящими из 16 аппаратных потоков.

Разные программные уровни параллелизма согласуются с аппаратной реализацией иерархической подсистемой памяти кластерной системы. LRnLA алгоритмы учитывают подобную специфику, что позволяет существенно ускорить вычисления. Если выполнение каждой подзадачи может производится отдельно и независимо от других подзадач, это позволяет воспользоваться распараллеливанием обрабатываемых процессов по разным вычислительным ресурсам — узлам кластера, графическим картам, мультипроцессорам и ядрам. Помимо всего прочего, часть подзадач можно выполнить штатными CPU процессорами стандартными инструментами (MPI, OpenMP, SSE2), что полностью подчеркнет гибридность системы.

Для обозначенной кластерной системы на рис. [model] на основе технических характеристик построены две ступенчатые диаграммы, отображающие иерархии подсистемы памяти для CPU (синий цвет) и GPU (красный цвет) устройств. Ширина каждой ступени диаграммы (начиная от нуля) показывает допустимый диапазон объема данных, который можно локализовать на данном уровне иерархии памяти; высота ступени показывает пиковую пропускную способность между соседними уровнями.

В качестве решаемой задачи рассматривается перемножение двух квадратных плотных матриц. Под выполнением задачи в этом смысле является получение элементов результирующей матрицы согласно формуле

линейной алгебры. На рис. [model] приводится двухмерная проекция графа разбиения для такой задачи, где вершинами являются одинакового типа подзадачи, обрабатываемые асинхронно однородными вычислительными элементами. Это позволяет равномерно распределить выполнение подзадач по вычислительным ресурсам кластерной системы для последующего параллельного и асинхронного разбиения с дальнейшим выполнением подзадач меньше размера.

Вычислительная модель построена для 5-уровневого алгоритма [3] перемножения матриц на гетерогенной системе К-100. На рис. [model] показана проекция графа разбиения на плоскость «размер данных — коэффициент локальности». По причине проецирования однородные вычислительные единицы будут находится в одной вершине графа.

Размер матриц варьировался от 1*215 до 7*215 элементов, что объясняет разное количество вершин (VII), а также равномерное разбиение подзадач (грани VII-VI) по разным узлам кластера. При дальнейшем разбиении (грани VI-IV) на одном кластерном узле подзадачи распределяются по графическим ускорителям. Для равномерной нагрузки вычислительной системы небольшая часть подзадач (грани VI-10) отводится для штатных CPU. Синим цветом указывается ветвь, выполняемая на CPU, красным цветом обозначена ветвь для GPGPU. При каждом последующем разбиении решение подзадач выполняется на конкретном уровне параллелизма, а необходимые для этого данные должны локализоваться на более высоких уровнях иерархии памяти.

Для отдельного графического GPGPU устройства можно выделить три уровня параллелизма — треды (вершина I), блоки (вершина III), варпы (вершина II), — также три уровня иерархии памяти — глобальная память (грани VI-III), share-memory (грани III-II), регистры (грани II-I). Поэтому дальнейший алгоритм решения задачи перемножения матрицы на одном GPU устройстве описывается в терминах 3-уровневого графа разбиения.

Для 3-уровневого алгоритма на графике [titan] представлена достигнутая производительность на одном устройстве GPU марки NVIDIA GeForce GTX Titan в зависимости от размера решаемой задачи (синяя линия). Помимо этого произведено сравнение с решением этой же задачи с использованием проприетарной библиотеки CUBLAS (красная линия) и открытой библиотеки Magma (зеленая линия).

Из графика [titan] наглядно видно, что LRnLA алгоритмы превысили 50% от пиковой производительности. Таким образом, ожидаемый для этих алгоритмов уровень эффективности достигнут. Стоит подчеркнуть независимость формулировки LRnLA алгоритмов, которая носит абстрактный характер и не привязывается к конкретной вычислительной машине или решаемой задаче. Для задачи умножения матриц адекватное сравнение по мнению авторов стоит производить с открытой библиотекой MAGMA, которая использует все известные способы алгоритмической оптимизации, но не включает ряд низкоуровневых оптимизаций, специфичных для конкретной архитектуры Kepler. Сравнивая графики производительности для MAGMA и CUBLAS, виден потенциал подобной низкоуровневой оптимизации (до трёх раз). При помощи дизассемблирования удалось выяснить, что в проприетарной библиотеке CUBLAS используются недокументированные команды ассемблера, позволяющие обойти конфликты по доступу к банкам регистрового файла. Из сравнения результатов 3-ех уровневого алгоритма для блоков разных размеров, обрабатываемых в каждом треде на нижнем уровне разбиения (тонкие синие линии) делается вывод, что указанные конфликты являются основными ограничениями для нашей реализации. К сожалению, инструментарий NVidia CUDA Toolkit 5.0 не содержит средств решения данной проблемы.

Проведенное исследование позволило определить оптимальные программные параметры для вычислений на гетерогенных системах. Это позволило сделать первые шаги в реализации конечно-разносных схем численного моделирования сейсморазведки на основе LRnLA алгоритмов.

ЛИТЕРАТУРА:

1. В.Д. Левченко «Декомпозиция данных и вычислений в локально-рекурсивных нелокально-асинхронных (LRnLA) алгоритмах»// Международная суперкомпьютерная конференция «Научный сервис в сетиИнтернет: все грани параллелизма» , Абрау-Дюрсо, 2013.
2. В.Д. Левченко «Асинхронные параллельные алгоритмы как способ достижения эффективности вычислений» // Информационные технологии и вычислительные системы, 1/2005, с.68-87
3. И.А. Горячев «Оценка вычислительной производительности локально-рекурсивныхнелокально-асинхронных алгоритмов на гетерогенных системах» // Международная суперкомпьютерная конференция «Научный сервис в сетиИнтернет: поиск новых решений» , Абрау-Дюрсо, 2012