23. Гомогенные и гетерогенные вычислительные системы. Взаимодействие процессоров общего назначения и ускорителей вычислений на примерах Nvidia CUDA и Intel MIC.

Из ВИКИ

Гетерогенные вычислительные системы — электронные системы, использующие различные типы вычислительных блоков. Вычислительными блоками такой системы могут быть процессор общего назначения (GPP), процессор специального назначения (например, цифровой сигнальный процессор (DSP) или графический процессор (GPU)), со-процессор, логика ускорения (специализированная интегральная схема (ASIC) или программируемая пользователем вентильная матрица (FPGA)).

В общем, гетерогенная вычислительная платформа содержит процессоры с разными наборами команд (ISA). Спрос на повышение гетерогенности в вычислительных системах, частично связан с необходимостью в высоко-производительных, высоко-реакционных системах, которые взаимодействуют с другим окружением (аудио/видео системы, системы управления, сетевые приложения и т. д.).

Основным методом получения дополнительной производительности вычислительных систем является введение дополнительных специализированных ресурсов, в результате чего вычислительная система становится гетерогенной. Это позволяет разработчику использовать несколько типов вычислительных элементов, каждый из которых способен выполнять задачи, которые лучше всего для него подходят.

Добавление дополнительных, независимых вычислительных ресурсов неизбежно приводит к тому, что большинство гетерогенных систем рассматриваются как параллельные вычислительные системы или многоядерные системы.

Ещё один термин, который иногда используется для этого типа вычислений «гибридные вычисления». Hybrid-core computing — форма гетерогенных вычислений, в которой асимметричные вычислительные устройства сосуществуют в одном процессоре.

Сайтик

Использование графических ускорителей (Graphical Processing Unit, GPU) наряду с классическими микропроцессорами (Central Processing Unit, CPU) сегодня является распространенной практикой в сфере параллельных вычислений на гетерогенных платформах, сочетающих в себе вычислительные элементы различного типа. Однако GPU и CPU изначально

разрабатывались как различные устройства, нацеленные на решение своих задач, и их совместное использование осложняется рядом проблем. Например, они имеют раздельную память и разные адресные пространства, поэтому приложение должно явным образом копировать данные в память GPU и обратно в основную память компьютера. Отправка вычислительного задания в очередь исполнения устройства осуществляется через стек драйверов с использованием системных вызовов и ядра операционной системы, что вносит большую задержку и зачастую делает нецелесообразным выполнение маленьких заданий на отдельном ускорителе. Кроме того, отправка заданий одного ускорителя другому или CPU затруднительна.

Heterogeneous System Architecture (HSA) — это архитектура гетерогенных систем с общей когерентной памятью для разнородных вычислительных элементов, поддержкой управления очередями задач, широкого спектра оборудования и языков высокого уровня.

В общем случае поддержка когерентности представляет собой сложную задачу. Правда, способность процессоров различных типов (CPU, GPU, DSP) работать с одними и теми же данными в общей памяти позволяет избавиться от лишних операций копирования и повысить производительность и энергоэффективность (рис. 1). В среде HSA приложения, работающие на CPU, могут выполнять отдельные задания на GPU/DSP так же легко, как и на CPU. Для этого приложению достаточно предоставить указатели на данные в общей памяти и обновить соответствующие очереди задач. При традиционном подходе приложение должно собрать все необходимые данные и, задействовав драйверы ОС, выполнить операции ввода-вывода для перемещения данных на вычислительное устройство, а затем запустить вычислительный процесс. HSA позволяет разработчикам ПО создавать приложения без необходимости глубоко вникать в специфику работы различных ускорителей, имеющихся на целевой системе, таких как GPU, DSP, видеокодеры/декодеры и прочие акселераторы.

Общая виртуальная память

Для упрощения работы ОС и приложений на платформе HSA используется единый набор таблиц виртуальных страниц для всех типов процессоров, что позволяет обмениваться указателями между любыми вычислительными устройствами — одинаковые указатели преобразуются в одинаковые физические адреса. Более того, данный подход избавляет ОС и менеджер

памяти от необходимости вести учет нескольких независимых наборов таблиц страниц для каждого устройства. Если CPU и GPU используют одинаковые участки памяти, то для традиционных GPU используются отдельные адресные пространства, и драйвер графического адаптера должен сбрасывать кэши при совместном использовании памяти с CPU.

Платформа HSA предоставляет разработчику полностью когерентную память, что упрощает создание приложений — где вычислители с разной архитектурой используются в рамках единых шаблонов программирования, позволяя применять привычные алгоритмы, например производитель-потребитель. Однако HSA поддерживает и некогерентную память, что бывает необходимо для получения максимальной производительности, когда нет необходимости совместного использования данных процессорами разного типа.

Операционная система, за счет механизма подкачки страниц, позволяет пользовательским процессам работать с объемами памяти, превышающими размер физической, однако обычные GPU могут работать только с невыгружаемой основной памятью — драйвер и ОС должны выделить участок в физической памяти, пометить его как невыгружаемый и создать отдельное виртуальное адресное пространство специально для работы ускорителя. HSA позволяет избавиться от невыгружаемой памяти и разделенных пространств адресов, позволяя всем GPU за счет механизма страничных прерываний использовать такое же большое адресное пространство, как и у CPU, с возможностью подкачки страниц.

Имеется множество задач, требующих памяти больше, чем могут предоставить современные графические ускорители, даже с учетом использования невыгружаемой основной памяти. В этом случае разделение задачи и данных на независимые части и реализация подкачки данных требует значительных усилий со стороны программиста, поэтому такие большие задачи редко переносились на GPU. Единое адресное пространство и страничные прерывания, обеспечиваемые архитектурой HSA, позволяют использовать ускорители и для подобных задач, что существенно повышает производительность, причем без усложнения программного кода. В дополнение к этому общее адресное пространство позволяет использовать структуры данных, содержащие указатели, такие как связные списки и деревья, одновременно доступные и CPU, и GPU. При традиционном подходе подобные случаи требуют от программиста отдельного рассмотрения и

зачастую являются основной причиной невозможности переноса алгоритмов на GPU.

Очереди и диспетчеризация задач

Работа ядра операционной системы зачастую была причиной появления узких мест в массово-параллельных системах — HSA позволяет снизить время, затрачиваемое на диспетчеризацию вычислительных задач, давая возможность процессам напрямую работать с очередями задач в пользовательском режиме без необходимости вызова системных функций и переключения в режим ядра. Такой подход минимизирует накладные расходы, вносимые системными драйверами. Кроме того, платформа HSA предоставляет ускорителям возможность аппаратного переключения контекста задач, стоящих в очереди, без необходимости вызова функций ядра ОС — специальный аппаратный планировщик способен существенно ускорить переключение контекста и сократить энергозатраты. Тем не менее ОС сохраняет за собой контроль над планировщиком, и в зависимости от сложности аппаратной составляющей конкретного ускорителя управление очередью может быть как чисто программным или аппаратным, так и смесью программного и аппаратного управления. В случае полностью аппаратного управления, CPU может ставить задачи на исполнение ускорителю без обращения к ОС.

Обеспечение малых задержек при диспетчеризации задач — важный элемент архитектуры HSA. Данная особенность позволяет рассматривать дополнительные ускорители как сопроцессоры и работать с ними соответствующим образом — разработчики приложений могут не опасаться снижения производительности из-за перемещения данных на внешнее или удаленное вычислительное устройство, а также за задания малого размера.

Очереди представляют собой области физической памяти, куда производитель помещает запросы, а потребитель их считывает. Потребителем в данном случае является исполняющее вычислительное устройство. Системные программные компоненты HSA отвечают за выделение и освобождение памяти и создание очередей — как только очередь создана, программист может посылать в нее задания без участия ОС, причем работать с очередями можно при помощи вызовов библиотечных функций или напрямую. При непосредственной работе с очередями необходимо учитывать их ограниченный размер и прочие нюансы. Библиотеки, в свою очередь, могут предоставлять возможность автоматической балансировки нагрузки и другой полезный функционал.

Очереди могут быть не только у ускорителей, но и у CPU, что позволяет любому вычислительному устройству посылать задачи на исполнение на любое другое вычислительное устройство. Возможны три варианта:

- Процессор посылает задачу на ускоритель. Данный вариант взаимодействия типичен для GPU-вычислений (OpenCL, CUDA).
- Ускоритель посылает задачу другому ускорителю (в том числе себе). Этот вариант позволяет ставить на исполнение дополнительные задачи без участия CPU. В противном случае необходимость выполнения кода на CPU и ожидания диспетчеризации соответствующего процесса ОС вносила бы очень большие задержки.
- Ускоритель посылает задачу на CPU. Таким образом ускоритель может запросить выполнение системных функций, например выделение памяти или осуществление ввода-вывода.

Важным требованием HSA является вытесняемость задач (preemption) — неотъемлемое свойство многопроцессорных многопользовательских систем. Без реализации вытесняемости одна задача может занять все устройство на произвольный промежуток времени, блокируя остальные задачи, стоящие в очереди. Ошибка в коде может привести к необходимости перезагрузки устройства. HSA не ограничивает разработчиков ускорителей в способе реализации вытесняемости.

Рассмотрим, как происходит запуск программы на графическом процессоре:

1. Хост выделяет необходимое количество памяти на устройстве.
2. Хост копирует данные из своей памяти в память устройства.
3. Хост запускает ядро на устройстве.
4. Устройство исполняет это ядро.
5. Хост копирует результаты из памяти устройства в свою память.

Книга по CUDE (более менее есть про описание архитектуры) http://elar.urfu.ru/bitstream/10995/40616/1/978-5-7996-1722-6_2016.pdf

Продукция на базе архитектуры Intel® Many Integrated Core (Intel® MIC) обеспечивает разработчикам ключевое преимущество: для ее работы используются уже имеющиеся стандартные инструменты и методы программирования.

Архитектура Intel® MIC объединяет несколько процессорных ядер Intel® в одной микросхеме. Разработчики, заинтересованные в программировании этих ядер, могут использовать стандартный исходный код

на языках C, C++ и FORTRAN. Тот же исходный код программы, написанный для продукции Intel® Many Integrated Core (Intel® MIC), можно компилировать и выполнять на стандартных процессорах Intel® Xeon®. Знакомые модели программирования позволяют разработчикам обойтись без дополнительного обучения и сосредоточиться на проблемах, не имеющих отношения к проектированию программного обеспечения.

24. Межсистемные коммуникации, ч. 1. Интерфейсы HTX, Ангара.

Hyper Transport Bus *(системная шина)* – высокоскоростная, двунаправленная системная шина по принципу точка-точка, разработанная для соединения низко скоростных системных шин, компонентов компьютеров, серверов, сетевых центров и телекоммуникационного оборудования, предоставляя до 48x прирост скорости.

Помогает сократить количество шин в системе и используется чаще всего в ПК, для соединения с контроллёром и оперативной памятью, позволяя им работать быстрее в одной среде и с меньшими задержками ввода-вывода. Очень часто шина используется и для соединения ядер процессора между собой.

Шина является последовательной. Скорость передачи зависит от двух параметров – ширины шины и частоты её функционирования. Шина, кроме передачи самих данных, может использоваться для передачи прерывания, служебных, системных и конфигурационных сообщений.

Шина может работать в двух режимах: Posted и Non—Posted. Первый обычно используется в настольных потребительских системах (для DMA-передачи к примеру) и обеспечивает максимальную скорость передачи данных. Posted операция записи просто посылает пакет с данными на определённый адрес, данные записываются и на этом всё. Non—Posted подразумевает передачу данных на определённый адрес, а после успешной записи в обратном направлении отправляется пакет с подтверждением успешной записи. Данный тип записи работает значительно медленней, но исключает возникновение ошибок передачи. Потому он используется преимущественно в серверных, научных, высокоточных машинах.

ВИКИ. Шина HyperTransport основана на передаче пакетов. Каждый пакет состоит из 32-разрядных слов, вне зависимости от физической ширины шины (количества информационных линий). Первое слово в пакете — всегда

управляющее слово. Если пакет содержит адрес, то последние 8 бит управляющего слова сцеплены со следующим 32-битным словом, в результате образуя 40-битный адрес. Шина поддерживает 64-разрядную адресацию — в этом случае пакет начинается со специального 32-разрядного управляющего слова, указывающего на 64-разрядную адресацию, и содержащего разряды адреса с 40 по 63 (разряды адреса нумеруются начиная с 0). Остальные 32-битные слова пакета содержат непосредственно передаваемые данные. Данные всегда передаются 32-битными словами, вне зависимости от их реальной длины (например, в ответ на запрос на чтение одного байта по шине будет передан пакет, содержащий 32 бита данных и флагом-признаком того, что значимыми из этих 32 бит являются только 8).

Пакеты HyperTransport передаются по шине последовательно. Увеличение пропускной способности влечёт за собой увеличение ширины шины. HyperTransport может использоваться для передачи служебных сообщений системы, для передачи прерываний, для конфигурирования устройств, подключённых к шине, и для передачи данных.

Операция записи на шине бывает двух видов — *posted* и *non-posted*. Posted-операция записи заключается в передаче единственного пакета, содержащего адрес, по которому необходимо произвести запись, и данные. Эта операция обычно используется для обмена данными с высокоскоростными устройствами, например, для DMA-передачи. Non-posted операция записи состоит из посылки двух пакетов: устройство, инициирующее операцию записи, посылает устройству-адресату пакет, содержащий адрес и данные. Устройство-адресат, получив такой пакет, проводит операцию записи и отсылает устройству-инициатору пакет, содержащий информацию о том, успешно ли произведена запись. Таким образом, posted-запись позволяет получить максимальную скорость передачи данных (нет затрат на пересылку пакета-подтверждения), а non-posted-запись позволяет обеспечить надёжную передачу данных (приход пакета-подтверждения гарантирует, что данные дошли до адресата).

Шина HyperTransport нашла широкое применение, в основном, в качестве замены шины процессора. Для примера, к процессору Pentium нельзя напрямую подключать устройства с шиной PCI, так как этот процессор использует свою специализированную шину (которая может быть различной у разных поколений процессоров). Для подключения дополнительных устройств (например, с шиной PCI) в таких системах необходимы дополнительные устройства для сопряжения шины процессора с

шиной периферийных устройств (мосты). Данные адаптеры обычно включают в специализированные наборы системной логики, называемые северный мост и южный мост.

Процессоры разных производителей могут использовать разные шины, а значит, для них нужны разные мосты для соединения шины процессора с периферийными шинами. Компьютеры, использующие шину HyperTransport, более универсальны и просты, а также более производительны. Однажды разработанный мост PCI-HyperTransport позволяет взаимодействовать любому процессору, поддерживающему шину HyperTransport, и любому устройству шины PCI. Например, чипсет NVIDIA nForceиспользует шину HyperTransport для соединения между северным и южным мостами.

Другое применение HyperTransport — шина NUMA многопроцессорных компьютеров. AMD использует HyperTransport как часть проприетарной архитектуры Direct Connect Architecture в своей линейке процессоров Opteron, Athlon 64 и Phenom. Технология шинного соединения Horus компании Newisys расширяет концепцию до уровня кластерных систем.

Сеть Ангара — отечественная сеть, сопоставимая по своей функциональности, производительности и надёжности с современными разработками мировых лидеров в данной области (Cray, IBM, Mellanox).

Сетевой адаптер представляет собой плату расширения PCI Express (аналогично сетевым адаптерам InfiniBand), к которой подключается до 6 (или до 8, с платой расширения) кабелей для соединения с соседними узлами. Поддерживаются топологии сети без коммутаторов: кольцо, 2D, 3D и 4D-тор (либо решётка); в общем случае передача данных между узлами может осуществляться через промежуточные узлы.

Основной режим программирования — совместное использование MPI, OpenMP и OpenSHMEM; также поддерживаются другие модели программирования, в том числе Charm++, GASNet, ARMCI, UPC. Для поддержки OpenSHMEM и PGAS-языков на каждом узле выделяется регион памяти, доступный для удалённых обращений (чтения, записи, атомарных операций) от других сетевых узлов. Сеть «Ангара» совместима с коммерчески доступными процессорами и материнскими платами.

СБИС сетевого адаптера (ЕС8430) была разработана в АО «НИЦЭВТ» и

изготовлена на фабрике TSMC по технологии 65 нм. Плата сетевого адаптера изготавливается на собственном производстве в АО «НИЦЭВТ».

Статья https://servernews.ru/931123

25. Межсистемные коммуникации, ч. 2. Интерфейсы InfiniBand, 10/40/100 Gbit Ethernet.

Infiniband (иногда сокр. **IB**) — высокоскоростная коммутируемая компьютерная сеть, используемая в высокопроизводительных вычислениях, имеющая очень большую пропускную способность и низкую задержку. Также используется для внутренних соединений в некоторых вычислительных комплексах. По состоянию на 2014 год Infiniband являлся наиболее популярной сетью для суперкомпьютеров. Контроллеры Infiniband (*host bus adapter*) и сетевые коммутаторы производятся компаниями Mellanox и Intel. При создании Infiniband в него закладывалась масштабируемость, сеть использует сетевую топологию на основе коммутаторов (*Switched fabric*).

В качестве коммуникационной сети кластеров Infiniband конкурирует с группой стандартов Ethernet и проприетарными технологиями[1], например, компаний Cray и IBM. При построении компьютерных сетей IB конкурирует с Gigabit Ethernet, 10 Gigabit Ethernet, и 40/100 Gigabit Ethernet. Также IB используется для подключения накопителей информации DAS.[2] Развитием и стандартизацией технологий Infiniband занимается InfiniBand Trade Association.

Подобно многим современным шинам, например, PCI Express, SATA, USB 3.0, в Infiniband используются дифференциальные пары для передачи последовательных сигналов. Две пары вместе составляют одну базовую двунаправленную последовательную шину (англ. *lane*),

обозначаемую 1x. Базовая скорость — 2,5 Гбит/с в каждом направлении. Порты Infiniband состоят из одной шины или агрегированных групп 4x или 12x базовых двунаправленных шин. Основное назначение Infiniband — межсерверные соединения, в том числе и для организации RDMA (Remote Direct Memory Access).

InfiniBand использует коммутируемую среду с соединениями точка-точка, в отличие от ранних вариантов сетей Ethernet, которые использовали общую среду и, изначально, шинное соединение. Все передачи начинаются и заканчиваются на адаптере канала. Каждый вычислительный узел содержит *HCA*-адаптер (host channel adapter), подключаемый к процессору по интерфейсу PCI Express (ранее через PCI-X). Между адаптерами пересылаются данные и управляющая информация, в том числе необходимая для реализации QoS.

Для периферийных устройств предполагалось использование TCA-адаптеров (target channel adapter), но они не получили распространения, а такие периферийные устройства создаются на базе стандартных материнских плат[11].

HCA-адаптеры обычно имеют один или два порта 4x, которые могут подключаться либо к таким же портам HCA и TCA, либо к коммутаторам (свитчам). Коммутаторы могут быть организованы в сети с топологиями типа утолщенное дерево (Fat Tree), Сеть Клоза, реже — многомерный тор, двойная звезда, и в различных гибридных комбинациях

Сори стало впадлу, поэтому
https://ru.wikipedia.org/wiki/10-гигабитный _Ethernet
https://ru.wikipedia.org/wiki/100-гигабитный _Ethernet
http://www.expressit.ru/?p=185 и https://ru.wikipedia.org/wiki/Ethernet –про Ethernet в общем