Двумерная виртуализация процессора: новый подход к повышению реальной эффективности суперкомпьютеров

Андрей Ефимов (eai@kbsp.by, eai_andr_kb@mail.ru)

"Конструкторское бюро системного программирования"
(г.Гомель, Беларусь).

Реальная эффективность и скорость: вызовы высокопроизводительных параллельных вычислений (ВПВ)

60-е годы – наиболее актуальны ВПВ научных расчетов (матфизика)

Вызов Сеймур Крэя: "Каждый : может построить быстрый процессор. Фокус в том, чтобы построить быструю систему"

Его ответ - иновационный гетерогенный суперкомпьютер CDC 6600 (1960е) :

- векторный центральный процессор (ЦП);

- мультитредингoвый сервисный процессор: 10 тредов в сервисном процессоре обеспечили толерантность доступа быстрой логики в медленную ферритовую память.

Новое время и актуальности - ВПВ общего назначения (СУБД, облака, реальное время [1-4])

Вызовы Бартона Смита:

– переизобрести компьютер и вывести ИТ индустрию из спирали специализации, при которой "высокопроизводительными считаются вычисления, хорошо выполняемые существующими высокопроизводительными системами";

- универсальные микропроцессоры (killer micros) также плохи для параллелизма общего назначения, как и динозавры, которые на них охотятся. …Что не в порядке с микропроцессорами? …Ответ очевиден – отсутствие толерантности к тонко-гранулированной латентности всех источников" (далее – обобщенной латентности).

Его ответ:

- 30-летнее развитие экстремального мультитредингa (XMT) в системах HEP Denecollor, HORIZON и Tera MTA, Cray MTA/XMT [4] (до 128 тредов) обеспечило толерантность для NUMA и неограниченное масштабирование NUMA-кластеров.

Обобщенная латентность как основа метрик эффективности и реальной скорости

“Четыре всадника Апокалипсиса” (Томас Стерлинг) вызывают состояния обобщенной латентности (ОЛ), снижающей эффективность и скорость выполнения реальной работы (КПД и Vr):

накладные расходы -- дополнительные работы аппаратуры на управление параллельностью;

латентность - простои при доступе к памяти или другим частям системы;

конкуренция - бесполезная активность при соперничестве за разделяемые ресурсы;

голодание - простои из-за слабого параллелизма и дисбаланса загрузки (в частности, из-за стены ILP в суперскалярных и VLIW процессорах).

На основе ОЛ в дополнении к экономическому показателю эффективность/стоимость, определим **чисто технический показатель**

КПД = 1 - Сол / С, где С – стоимость аппаратуры при выполнении работы, а Сол - стоимость ее части, находившей в состоянии обобщенной латентности.

Определим реальную скорость Vr (пропорциональную КПД) как

Vr = Vb * VM, где Vb (байт в секунду) – скорость обработки операндов в приведенном к байту формате, а VM (байт) – объем памяти системы, на которой обеспечивается данная скорость Vb.

Повышение эффективности и реальной скорости на основе 2х-мерной виртуализации

Рост скорости ВПВ исторически **обеспечивался аппаратной виртуализацией процессоров в 2х ортогональных направлениях - памяти и тредов,** как схематически показано на следующем слайде.

По горизонтали – типы памяти, их размеры и оценки времен доступ в процессорных циклах (clk).

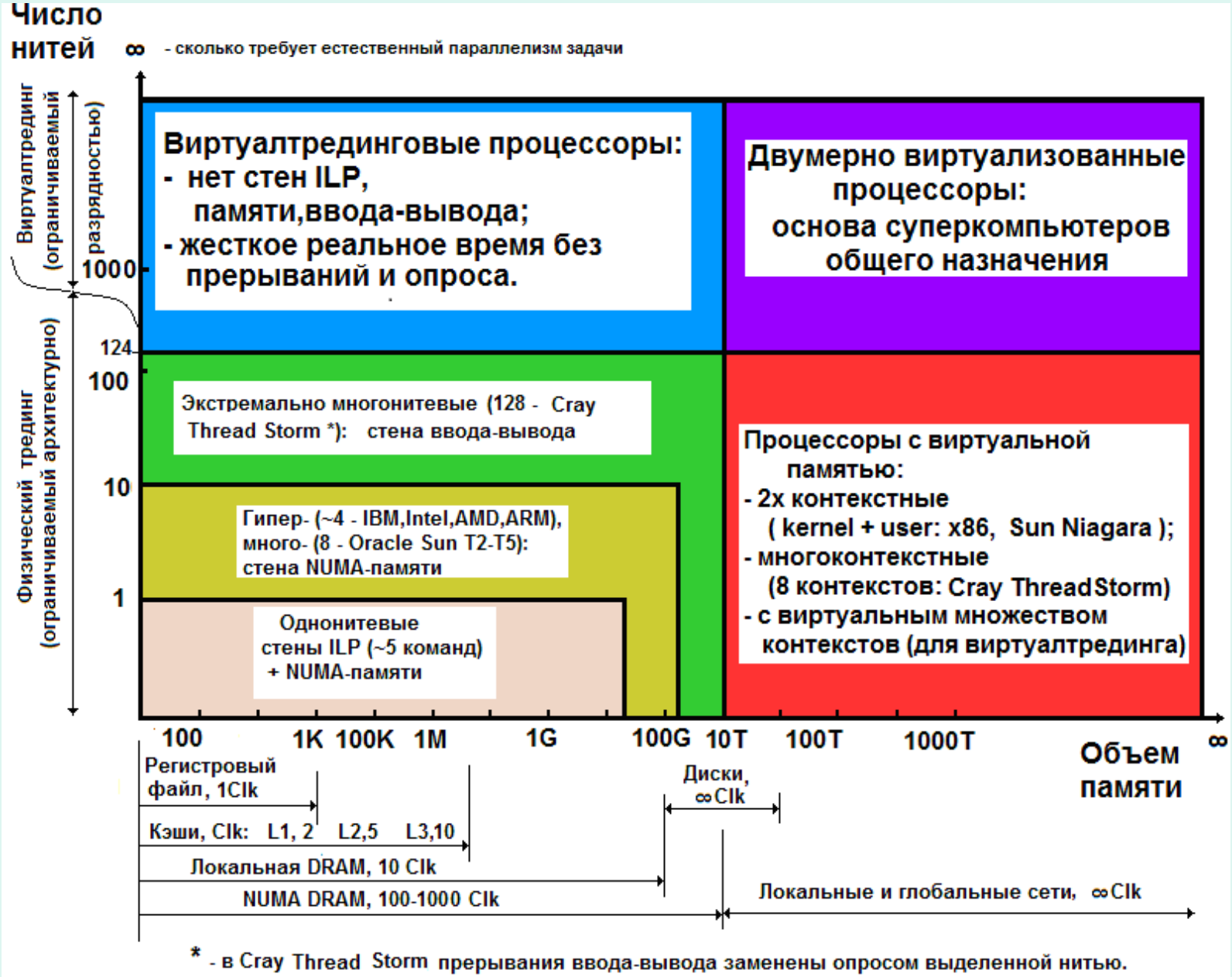
По вертикали – типы систем и количество тредов в них.

Выделенные цветом прямоугольники связывают типы систем и типы памяти, к которой эти системы обеспечивают толерантный доступ:

- простые 1-тредовые - к данным в регистровом файле;
- 1-тредовые с ILP (суперскалярный и VLIW) – к кэшам L1, L2 и частично к локальной памяти (совмещения обращения к памяти с вычислениями, но ILP wall позволяет запускать в такте около 5 команд);
- гипертрединговые, используя ILP и TLP – к локальной памяти;
- XMT-процессоры (Cray Thread Storm, [4]) (ILP + экстремальный TLP), исполняя в общем конвейере потоки 128 тредов, толерантны к NUMA.

Но XMT-процессоры не обеспечивают толерантность доступа к виртуальной памяти, отображаемой на внешние устройства (локальные или сетевые диски) – они **сталкиваются** со стеной, **которую естественно назвать стеной ввода-вывода (IO wall).**

Двумерная виртуализация процессоров



* - в Cray Thread Storm прерывания ввода-вывода заменены опросом выделенной нитью.

От к ХМТ к виртуалтредингу

Увеличение числа тредов в ХМТ до некоторого предела снижает стоимость оборудования, повышает КПД и реальную скорость Vr. Но это увеличение приводит к пропорциональному росту размера микроархитектурного регистрового файла (МРФ) и квадратичному росту его связей с конвейером, что приводит к снижению КПД из-за роста объема латентных элементов МРФ и связей.

Способ уменьшения размера МРФ за счет тонко-гранулированной виртуализации предложен в архитектуре виртуального контекста (VCA) [6], которая наиболее близка к предлагаемой виртуалтрединговой. В VCA впервые тонко-гранулированным элементом свопинга стал отдельный архитектурных регистр в отличие от крупного гранулированного свопинга полного набора регистров треда в ХМТ. Однако VCA оптимизирует только регистровый файл рабочего множества нитей, а поддержка полного множества оставляется операционной системе.

Предлагаемая в докладе виртуалтредингова архитектура (ВТА) является развитием архитектур ХМТ и VCA и представляется их логическим завершением.

Основные концепции виртуалтредингговой системы

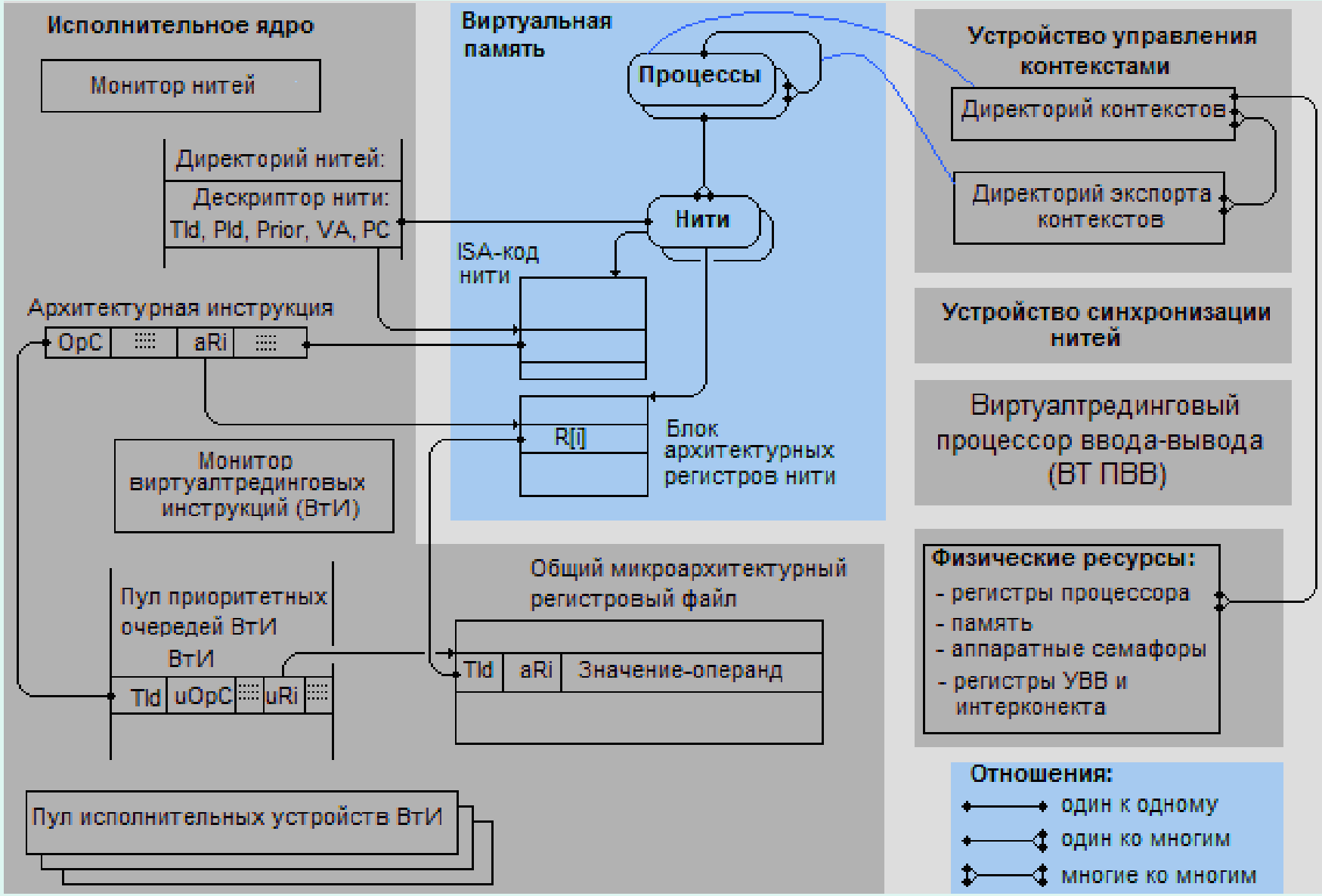
1. **Устранена концептуальная причина низкой эффективности существующих реализаций ВПВ - Силиконовый Занавес в виде фиксированного множества тредов и аппаратных контекстов** (схема на следующем слайде). Виртуалтредингговая ISA и транзакционная микроархитектура поддерживают, в развитие известной VCA, виртуальные множества программных нитей и процессов (контекстов).

2. **Транзакционная микроархитектура реализует прямую тонко-гранулированную приоритетную виртуализацию активностей полного множества нитей** за счет замены тредов как аппаратной единицей активности их своих тонко-гранулированными элементами – виртуалтредингговыми инструкциями (ВтИ), которые аппаратура генерирует динамически для полного множества нитей. При этом каждая ВтИ наследует все атрибуты, представляющие порождающую нить как объект управления в ОС, в том числе принадлежность к процессу-хозяину и приоритет.

3.**Устройство выборки команд и исполнительный конвейер в виртуалтредингговой архитектуре обеспечивают параллельное приоритетное исполнение полного множества нитей** (вычислительных и ввода-вывода) за счет исполнения ВтИ без участия программного обеспечения.

Различия мультитредингового и виртуалтредингового процессоров

Эталонная модель виртуалтредингговой СнК



Двухуровневая организация исполнения инструкций

Монитор нитей реализует базовую ISA для обеспечения полной преемственности унаследованного ПО и виртуалтредингговую ISA, в которой системные вызовы ОС, реализующие управление процессами, разделяемой памятью, нитями, их синхронизацией и вводом-выводом заменены виртуалтредингговыми инструкциями (ВтИ).

Исполнение реализуется по двухуровневой схеме:

уровень 1- монитор нитей потактово анализирует готовности к исполнению и приоритет полного множества нитей и формирует на этой основе из первичных ISA-инструкций пул приоритетных очередей ВтИ .

уровень 2 - монитор ВтИ ведет пул приоритетных очередей ВтИ в микроархитектурной памяти, через который он взаимодействует с монитором нитей и пулом исполнительных устройств.

Взаимодействуя с монитором нитей, монитор ВтИ принимает ISA инструкции, переводит их в формат ВтИ и помещает в пул ВтИ. После отработки ВтИ пулом исполнительных устройств монитор ВтИ выдает результаты в монитор нитей.

Организация управление доступом к разделяемым данным, синхронизация и ввод-вывод

Устройство управления контекстами совмещает функции устройств управления памятью и памятью ввода/вывода и **обеспечивает аппаратную поддержку полного множества контекстов** - ВтИ любой нити и контекста могут обращаться к данным других контекстов за счет прямой аппаратной поддержки экспорта разделяемых регионов с использованием директория экспорта контекстов.

Устройство синхронизации нитей реализует работу в критических секциях на основе аппаратных семафоров.

Виртуалтредингговый процессор ввода-вывода позволяет организовать обмены с внешними устройствами без прерываний.

Логику совместной работы этих устройств иллюстрируют следующие слайды.

Транзакционная микроархитектура

Монитор ВтИ – ключевой элемент транзакционной микроархитектуры, поддерживающий ВтИ на разных стадиях исполнения в конвейерах исполнительных устройств, конкурируя за ее ресурсы – обрабатывающую логику, связи и микроархитектурную память. Размещение всех операндов завершает этап подготовки операндов ВтИ как транзакции, которая далее может быть отработана в нескольких фазах исполнение и ожидания. Важнейшим свойством виртуалтрединга является поддержка перехода как ВтИ, так и породившей нити, в состояние ожидания произвольной длительности без участия ОС.

Транзакционная микроархитектура дает ответ практически на все приведенные в [3] требования к "переизобретенному" универсальному суперкомпьютеру:

- **приоритетный пул ВтИ можно рассматривать как пул транзакций, для которого клиентами являются монитор и исполнительные устройства ВтИ;**

- в терминах требований [3], серверами для ВтИ, являются:

- транзакционная память, используемая в существующих системах;

- устройство синхронизации нитей средствами аппаратных семафоров;

- виртуалтреридинговый процессор ввода-вывода, обеспечивающий организацию обменов без прерываний.

Синхронизация без инверсии приоритетов

Прямая аппаратная поддержка монитором ВтИ смены состояний “ожидание-активность” ВтИ (и, как следствие, их нитей-хозяев) позволяет заменить в системах на основе виртуалтрединга программные функции ОС аппаратными инструкциями синхронизации.

ВтИ синхронизации отрабатывает устройство синхронизации нитей как транзакции, на основе аппаратных семафоров -- активных объектов, реализующих работу таймера и ведение очередей ВтИ. Это устройство автономно, локально для каждого семафора, реализует в расширенном виде объединенную функциональность имеющихся в современных ОС мутексов (от MUTual EXlusion) и условных переменных (condition variables) (следующий слайд).

Аппаратное таймировании и повышение приоритета нити при работе в критической секции очень просто и эффективно решают проблему инверсии приоритетов, особенно актуальную при работе в жестком реальном времени.

Кроме того, замена системных вызов аппаратными инструкциями синхронизации, делает ненужными фьютексы (Fast User space muTEXes).

Инструкции синхронизации

Виртуалтредингговые средства синхронизации представлены следующими инструкциями, реализующими наиболее важные функции синхронизации стандарта POSIX:

- **SGet с операндами tw –(тайм-аут ожидания входа нити в критическую секцию – (КС)) и prs (приоритет при работе в критической секции) объединяет функций pthread_mutex_init и pthread_cond_init, выделяет программе аппаратный семафор (из пула свободных) и возвращает его указатель ps;**

- SLock(ps, ts) заменяет функцию pthread_mutex_lock и обеспечивает либо вход нити в КС с тайм-аутом пребывания в ней ts, приоритетом prs (если эта секция свободна), либо ожидание входа в нее в течении ts до выдачи команды SPas(ps);

- SWait(ps, ts) (аналог pthread_cond_timedwait) реализует неактивное ожидание входа в КС до выдачи команды SPas(ps), позволяя устранить ресурсоемкий опрос разделяемых переменных в критической секции;

- SPas(ps) объединяя действия функций pthread_mutex_unlock и pthread_cond_broadcast, выводит нить из критической секций, охраняемой семафором ps и освобождает ее для других нитей.

Ввод-вывод: Interruptions considered harmful

Ввод-вывод в виртуалтредингoвой СнК использует однородное управление вычислительными нитями и нитями ввода-вывода, причем вычислительные нити управляют нитями ввода-вывода как ведомыми. Виртуалтредингoвый процессор ввода-вывода (ПВВ) организован аналогично исполнительному ядру и поддерживает полное множество нитей ввода-вывода для каждого канала – независимой активности ввода-вывода.

Выделение пары нитей каждому каналу реализует прерывания как сообщения, отображающие завершение обращений в удаленную память с разницей, что при обращениях в память агентами являются регистр и память, а при вводе-выводе – оперативная память и память устройства.

Поддерживая виртуальное множество контекстов и нитей обоих типов, виртуалтрединг устраняет понятия “аппаратное ядро” как средство поддержки треда и ядра как средств реализации физического параллелизма в связанном контексте. Это свойство виртуалтредингa позволяет утверждать, что прерывания в современных СнК еще более вредны, чем оператор goto в языках программирования. Отсутствие прерываний существенно упрощает как организацию фон-Неймановских машин, так и их программирование.

Вместо заключения: Время собирать в камни

Мир ИТ становится экстремально параллельным– запустив диспетчер задач Widows на своем ПК, каждый может увидеть стократное превышение числа программных нитей над числом тредов.

Виртуалтредингoвая архитектура (ВТА) позволяет устранить Силиконовый Занавес аппаратных тредов и “собрать в камни” прямое тонко-гранулированное приоритетное мультипрограммирование (= виртуализацию активностей) и на этой основе объединить программистов и инженеров-схемотехников для переноса Силиконового Занавеса в самое подходящее место – перед всадниками Апокалипсиса. Это позволит объединившимся комплексным специалистам, как и пользователям их разработок, избежать судьбы динозавров и обеспечить длительную эволюции в таком параллельном мире.

Введенные КПД и метрика реальной скорости могут стать основой чисто технической системы оценки суперкомпьютеров.

При конвергенции усилий сообщества российских специалистов может быть достаточно быстро создана виртуалтредингoвая СнК как фундамент для российских суперкомпьютеров общего назначения.

Одним из первых камней в этом фундаменте (сделанным, возможно силами студентов и аспирантов технических университетов) должна стать СнК с виртуалтредингoвой реинкарнацией выдающейся русскоязычной системы Эльбрус-2. Это будет хорошей подстраховкой переводу ряда важнейших систем на совершенно новые аппаратные и программные платформы.

Список цитируемых работ

1. Smith, J.E. et all, Future general purpose supercomputer architectures. Proceedings of the 1990 ACM/IEEE conference on Supercomputing.

2. Burton Smith, The Quest for General Purpose Parallel Computing 1994, Developing a computer science agenda for high-performance computing, ACM New York, NY, 1994

3. Burton Smith, Reinventing Computing Microsoft Research Faculty Summit 2007, http://www.cct.lsu.edu/~estrabd/LACSI2006/Smith.pdf

4. Burton Smith, The end of architecture, ACM SIGARCH Computer Architecture News Homepage archive, Vol. 18, № 4, Dec. 1990.

5. Meeting the Demands of Unstructured Data with an extreme multithreaded machine. Introducing the Cray XMT Supercomputer, http://www.cray.com/Assets/PDF/products/xmt/CrayXMTOverviewWhitepaper.pdf

6. How to Fake 1000 Registers, Proceedings of the 38th annual IEEE/ACM International Symposium on Microarchitecture, http://aggregate.ee.engr.uky.edu/LAR/micro05.pdf

Спасибо за внимание

Андрей Ефимов
eai@kbsp.by
eai_andr_kb@mail.ru

"Конструкторское бюро системного программирования"
(г.Гомель, Беларусь).