С. С Андреев, А. А. Давыдов, С. А. Дбар,
А. Б. Карагичев, А. О. Лацис, Е. А. Плоткина

Институт прикладной математики им. М. В. Келдыша РАН,
Москва

Опыт создания макета гибридного NUMA - кластера.

Работа выполнена при финансовой поддержке РФФИ,

проект 08-07-00086-а.

Почему мы задумались о машине с новой архитектурой?

- **Архитектура суперкомпьютера, построенного по кластерной технологии, определяется свойствами исходного материала, то есть компонентов крупносерийного выпуска.**
- **Качественные изменения в нашем исходном материале происходят примерно раз в 10 лет.**
- **Примерно 10 лет назад мы начали строить кластеры в их нынешнем виде.**
- **Что идет им на смену и почему?**

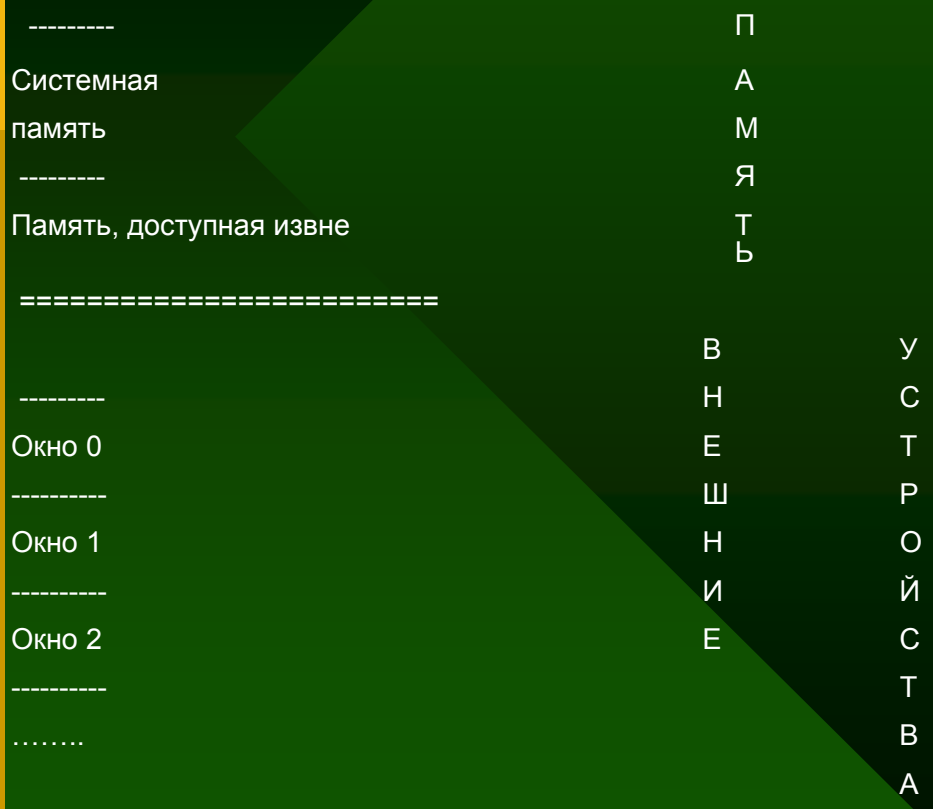
2007 год отмечен реальным появлением многочисленных новых архитектурных решений.

- **Изменения коснулись как средств интеграции процессоров в систему (коммуникационной сети), так и устройства самого вычислительного узла.**
- **Работа на коммуникационном направлении оказалась гораздо более понятной и обещала полезные результаты за гораздо меньшее время, поэтому в этом году мы сосредоточились в основном на ней.**
- **Эту работу удалось довести до создания макета, пригодного к испытаниям на реальных задачах.**

В чем новизна предлагаемой коммуникационной сети?

- **До недавнего времени мы не знали доступного способа более тесной интеграции процессоров в единую систему, чем объединение их локальной сетью.**
- **Точнее, попытки более тесной интеграции автоматически означали удорожание системы в десятки раз, и на это шли только такие разработчики, как Cray.**
- **С появлением PCI Express положение изменилось. Достижимый еще недавно только в классе особо дорогих систем уровень тесноты связи процессоров в системе стал теперь доступен для кластеров.**
- **Сегодня совершенно реально строить кластеры с общим полем прямо адресуемой памяти и общим полем внешних устройств.**

Адресное пространство узла NUMA – системы с частично общей памятью.



Каждое окно аппаратно отображается на доступную извне память соответствующего узла.

Как выглядит программа?

- **С момента запуска до момента выяснения собственного номера ветви и числа ветвей – все как в случае MPI.**
- **Затем программа получает у системы указатели на окна и область внешнего доступа.**
- **Далее, вместо обращений к функциям «послать» и «принять» - прямой доступ к «чужой» памяти и барьерная синхронизация.**
- **Стиль программирования языка Co-array Fortran, но на C.**

Как это сделано?

- **Общепринятая сегодня технология организации внутри-машинных коммуникаций PCI Express является полноценной сетью, и не нуждается в дополнительных сетях для выхода за пределы системного блока.**
- **Для объединения системных блоков между собой не хватает только коммутатора каналов этой сети.**
- **Мы такой коммутатор построили.**
- **Перебрав несколько вариантов, мы остановились на использовании сетевых процессоров 8000-й серии производства фирмы PLX. Это недорогие микросхемы массового выпуска.**

О размерах системы.

- **Коммутатор каналов PCI Express – модульный и масштабируемый.**
- **Построенный сегодня макет имеет в своем составе 16 процессорных ядер (4 узла по 2 процессора Opteron, по 2 ядра в каждом) и коммутатор на 5 каналов (один не используется).**
- **Сегодняшний уровень проработанности технологии позволяет уверенно говорить о системах с ориентировочным размером в 100 процессорных ядер.**

Основные параметры производительности.

- **Скорость записи в смежные адреса памяти порциями не менее 64 байт: 650-700 МБ/с**
- **Время выполнения операции «запись» при использовании не смежных адресов, в интенсивном потоке: 65нс**
- **Латентность записи одного слова: 1мкс**
- **Время синхронизации двух процессоров: 1.2мкс**
- **Время чтения слова: 2мкс**
- **Использовались каналы x4.**

А зачем это все нужно?

- **Что конкретно дает такая коммуникационная система:**
- **А). Программе**
- **Б). Программисту**

Машина строится не для латентности и не для задержек, а для решения задач. Как и насколько улучшится решение задач?

Размер зерна параллелизма

- **Размер зерна параллелизма – единственная интегральная характеристика качества коммуникационной системы.**
- **Для его измерения предлагается использовать известный «тест Якоби».**
- **Каков минимальный размер квадратного поля температуры, еще допускающий распараллеливание на 16 процессорных ядер?**

Сравнение с кластером rsc4, работающим сегодня в ИПМ.

- **rsc4: 500 на 500 ячеек.**
- **МВС-экспресс: 64 на 64 ячейки, или около 1000 полезных арифметических операций между актами синхронизации.**
- **МВС-экспресс на неструктурной сетке: тоже 64 на 64 ячейки.**
- **Зерно параллелизма сокращено примерно в 50 раз, неструктурная сетка в этом смысле не хуже индексной.**

Решение СЛАУ методом Гаусса – знаменитый «Тест Linpack».

- **Канонический вариант содержит примерно 1М исходного текста. Сложные манипуляции с рассылками данных, описываемые на десятках страниц документации. Реализован лишь частичный выбор главного элемента.**
- **МВС-экспресс: вариант решателя для «канонического» алгоритма содержит менее 100 строк, вариант с полным выбором главного элемента – 200 строк.**
- **Проблемы с синхронизацией. Барьеров не достаточно.**

Основные выводы.

- Внятной, общепринятой модели программирования для NUMA с высокой степенью неоднородности доступа к данным не существует.
- При ее разработке проблемы будут не столько с моделью доступа к памяти, сколько с синхронизацией.

Несколько слов о новых архитектурах вычислителя.

- Рассмотренные выше изменения в коммуникационной системе – однозначный «плюс». Хуже и сложнее жить от них точно никому не станет.
- С изменениями в архитектуре вычислителя все сложнее. Они являются вынужденными и усложняющими жизнь программиста, как когда-то – переход от последовательных машин к параллельным.

Что вынуждает нас задуматься о новых вычислительных архитектурах?

- **Известная проблема «паровозного кпд» современных процессоров.**
- **Проблема имеет системный характер.**
- **В программе универсального процессора слишком много «вспомогательных» (с точки зрения математика) команд.**
- **Ускорение «полезных» операций до уровня «вспомогательных» сделало сокращение доли «вспомогательных» команд главным источником повышения быстродействия.**
- **К сожалению, для такого сокращения надо перепроектировать процессор.**

Три магистральных направления:

- **Новые универсальные архитектуры (мультитрединг)**
- **Специализированные ускорители (обычно векторные) – Cell, GPGPU.**
- **Реконфигурируемые вычислители (процессоры одной задачи на программируемой логике).**
- **Мы работаем, в основном, по третьему направлению.**

Положение дел в этой области.

- **Эти работы получили мощный толчок в последнее время, когда технология PCI Express позволила значительно сократить зерно параллелизма во взаимодействии универсального процессора с ускорителем.**
- **Смысл работы – в максимально возможном переносе «вспомогательных» действий в структуру схемы. Должен получиться «процессор одной задачи», гарантированно достигающий своего пикового быстродействия.**
- **Главная проблема – отсутствие языков разработки прикладных схем, понятных математику.**
- **Проблема эта системная. Языков нет, потому что не выстроена система понятий.**

Все упирается в человеческий фактор.

- **Поэтому так важны даже скромные результаты, если они получены бригадой, не имеющей многолетнего опыта разработки специализированных вычислителей**
- **Удастся ли вообще заметно ускорить типовой для традиционных вычислительных приложений фрагмент, если реализовать его схемно?**
- **На «тесте Якоби» нам это удалось.**

Предварительные результаты для «теста Якоби», 256 на 128.

- **Технический уровень использованного кристалла соответствует процессору Pentium-3, пиковое быстродействие которого – 900 Мфлопс.**
- **На «тесте Якоби» реально достигается примерно 60Мфлопс.**
- **При схемной реализации получаем 900 Мфлопс (примерно в 15 раз быстрее)**
- **Это заметно быстрее даже по сравнению с процессорным ядром современных процессоров, если брать реально достигаемую ими производительность на этой задаче.**

Предварительные результаты использования GPGPU

- **Использование такого оборудования обещает более быстрый эффект, чем использование программируемой логики.**
- **А. А. Давыдов проводил измерения быстродействия на:**

- **Решении задачи Римана о распаде произвольного разрыва (много мелких независимых задач), и на**
- **2D уравнениях Эйлера по схеме Годунова первого порядка (одна задача)**

В первом случае ускорение составило примерно 30 раз, во втором – 8-10.

Этим оборудованием мы готовы оснастить наш макет уже сегодня.

- Спасибо за внимание.
- Вопросы?