IX Международная молодежная научно-практическая школа
«Высокопроизводительные вычисления на грид-системах»,
г. Архангельск, Россия, 6 февраля 2018 года

Опыт разворачивания и сопровождения международных проектов добровольных распределенных вычислений

Курочкин Илья Ильич

Центр распределенных вычислений
Института проблем передачи информации РАН

- Основные термины и понятия
- Основные архитектуры МВС
- Видеокарты и Xeon Phi
- Грид-системы
- Грид-системы из персональных компьютеров

Единицы измерения и термины

МВС — многопроцессорные вычислительные системы

HPC (High-performance computing) — высокопроизводительные вычисления

Многие современные процессоры управляются **тактовым генератором**. Процессор внутри состоит из логических элементов и ячеек памяти — триггеров. Когда приходит сигнал от тактового генератора, триггеры приобретают своё новое значение, и логическим элементам требуется некоторое время для декодирования новых значений. Затем приходит следующий сигнал от тактового генератора…

Такт процессора или такт ядра процессора — промежуток между двумя импульсами тактового генератора, который синхронизирует выполнение всех операций процессора.

Выполнение различных элементарных операций (команд или **инструкций**) может занимать от одного до многих тактов в зависимости от команды.

Единицы измерения вычислительной мощности компьютера:

MIPS (Million Instructions Per Second) — количество миллионов инструкций процессора в секунду.

FLOPS (FLoating-point Operations Per Second) —количество операций с плавающей точкой в секунду.

Наборы инструкций процессора

RISC (***R**estricted (reduced) **I**nstruction **S**et **C**omputer*) — архитектура процессора, в которой быстродействие увеличивается за счёт упрощения инструкций, чтобы их декодирование было более простым, а время выполнения — короче.

Упрощение инструкций облегчает повышение тактовой частоты.

CISC (**C***omplex **I**nstruction **S**et **C**omputing*) — архитектура процессора, в которой реализуется расширенный набор инструкций.

Данная концепция проектирования процессоров характеризуется следующим набором свойств:

- нефиксированное значение длины команды;
- арифметические действия кодируются в одной команде;
- небольшое число регистров, каждый из которых выполняет строго определённую функцию.

Неоднозначность понятия «суперкомпьютер»

Несколько попыток определения понятия суперкомпьютер (экономическое, физическое, философское)

Производительность системы может сильно зависеть от типа выполняемой задачи

Оценка производительности осуществляется с помощью специализированных тестов

Оценки производительности

Основные тесты:

- LINPACK(решение СЛАУ)
- HPL (реализация Linpack для top500 и не только)
- NAMD (решение задач молекулярной динамики)
- HPCC (HPC Challenge Benchmark)
- NAS Parallel Benchmarks (версия 3.3 состоит из 11 тестов)
- 14(24) ливерморских цикла (ориентированы на векторные компьютеры)

Различают **пиковую производительность** (теоретическая величина = число процессоров * число процессоров в системе)

Реальную (фактическую) производительность – величина вычисляется с помощью ряда тестов

Рейтинги МВС:

- TOP500,
- Green500,
- Top50 СНГ.

Дата-центр или ЦОД

Дата-центр (от англ. **data center**), или **центр** (хранения и) обработки данных (**ЦОД**/ЦХОД) — это специализированное помещение/здание для размещения серверного и сетевого оборудования и его подключения к каналам глобальных сетей

Облако

Модель обеспечения сетевого доступа по требованию к общему пулу конфигурируемых ресурсов

Обязательные характеристики облака

- Универсальный доступ по сети
- Объединение ресурсов *(resource pooling)*
- Эластичность
- Самообслуживание по требованию *(self service on demand)*
- Учёт потребления ресурсов

Основные модели обслуживания

- **SaaS** *(Software-as-a-Service)*
- **PaaS** *(Platform-as-a-Service)*
- **IaaS** *(Infrastructure-as-a-Service)*

Классификации МВС

Классификации архитектур

- Флинна (основная)
- Дополнения Ванга и Бриггса к классификации Флинна
- Фенга (число бит в слове и число слов)
- Шора (типичные способы компоновки)
- Хендлера (по возможности конвейерной и параллельной обработки информации)
- Хокни (по топологии соединения)
- Скилликорна
- Шнайдера
- …

Классификация по Флинну

	Одиночный поток команд (Single Instruction)	Множество потоков команд (Multiple Instruction)
Одиночный поток данных (Single Data)	SISD (ОКОД)	MISD (МКОД)
Множество потоков данных (Multiple Data)	SIMD (ОКМД)	MIMD (МКМД)

Дополнения Ванга и Бриггса

Класс **SISD** разбивается на два подкласса:

- архитектуры с единственным функциональным устройством (PDP-11)
- архитектуры, имеющие в своем составе несколько функциональных устройств (CDC 6600, CRAY-1, FPS AP-120B, CDC Cyber 205, FACOM VP-200)

В классе **SIMD** также вводится два подкласса:

- архитектуры с пословно-последовательной обработкой информации (ILLIAC IV, PEPE, BSP)
- архитектуры с разрядно-последовательной обработкой (STARAN, ICL DAP)

В классе **MIMD**:

- вычислительные системы со слабой связью между процессорами, к которым они относят все системы с распределенной памятью (Cosmic Cube)
- вычислительные системы с сильной связью (системы с общей памятью), куда попадают такие МВС, как C.mmp, BBN Butterfly, CRAY Y-MP, Denelcor HEP.

Классификация Фенга

В 1972 году Фенг (T. Feng) предложил классифицировать вычислительные системы (ВС)
на основе двух простых характеристик.

Первая характеристика – **число n бит в машинном слове**, обрабатываемых параллельно при выполнении машинных инструкций.

Вторая характеристика – **число m слов**, обрабатываемых одновременно.

Произведение P = n x m определяет интегральную характеристику потенциала параллельности архитектуры, которую Фенг назвал максимальной степенью параллелизма ВС.

Иерархия памяти

Иерархия памяти

Различные виды памяти образуют иерархию, на различных уровнях которой расположены памяти с отличающимися

- временем доступа,
- сложностью,
- стоимостью,
- объемом.

Возможность построения иерархии памяти вызвана тем, что большинство алгоритмов обращаются в каждый промежуток времени к небольшому набору данных, который может быть помещен в более быструю, но дорогую и поэтому небольшую, память.

Уровни иерархии памяти

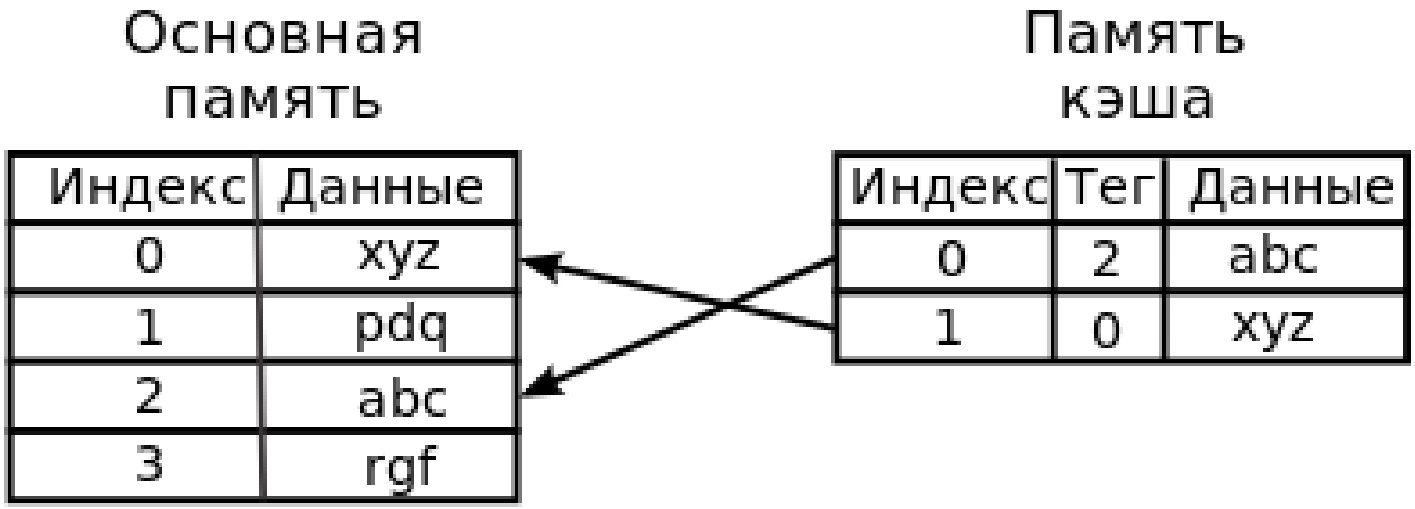
1. Внутренняя память процессора (регистры и кэш нескольких уровней)
2. Оперативная память (ОЗУ)
3. Вторичная память
   (жесткие диски, твердотельные накопители)
4. Третичная память
   (внешние носители, сетевые и распределенные хранилища)

Пирамида иерархии памяти

Виды кэшей

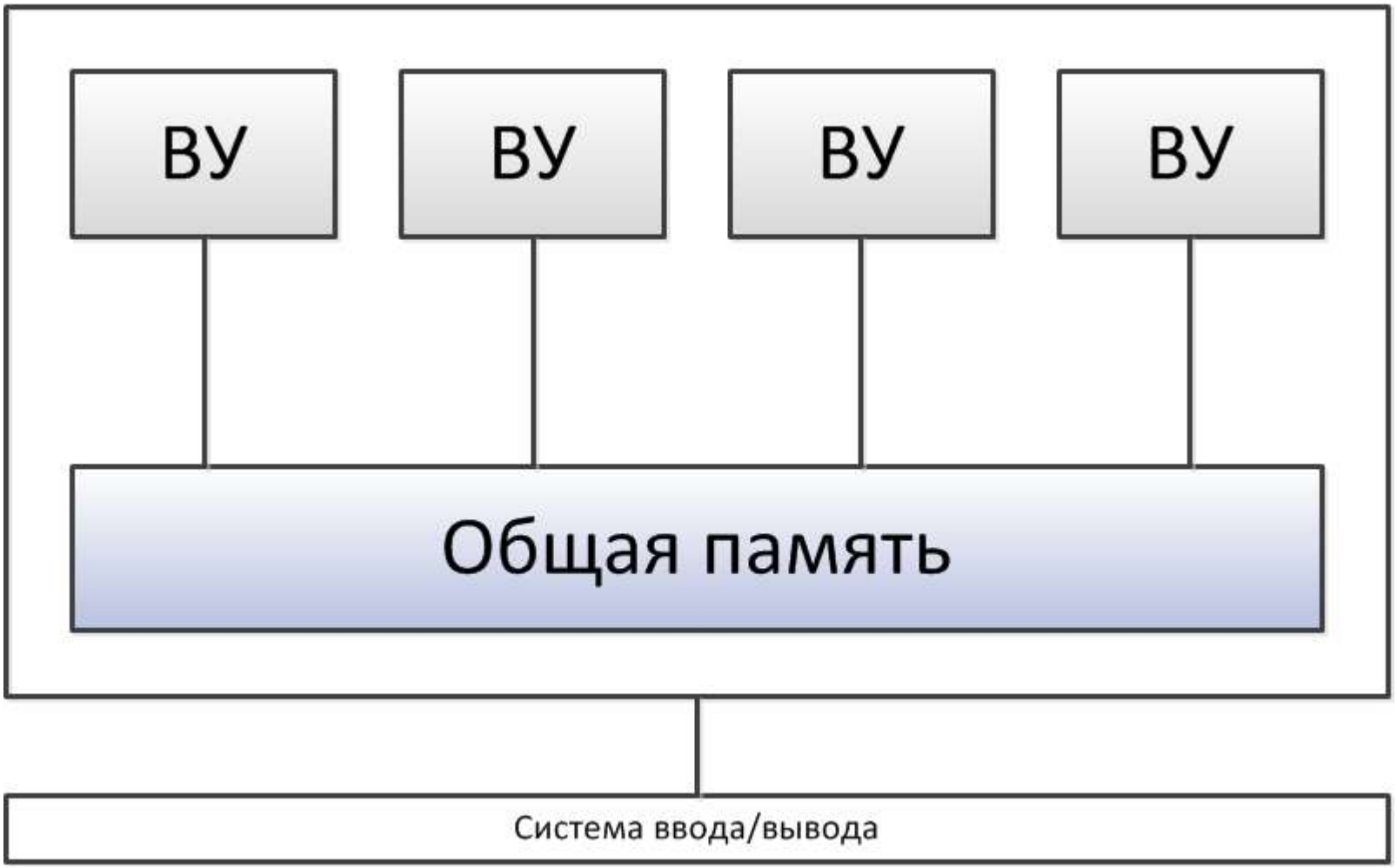
- Кэш данных (как правило 3 уровня)
- Кэш инструкций
- Буфер ассоциативной трансляции (TLB) для ускорения трансляции виртуальных (математических) адресов в физические, как для инструкций, так и для данных

Взаимодействие оперативной памяти и кэш-памяти



Архитектуры МВС

Symmetric multiprocessing (SMP)



Особенности SMP-архитектуры

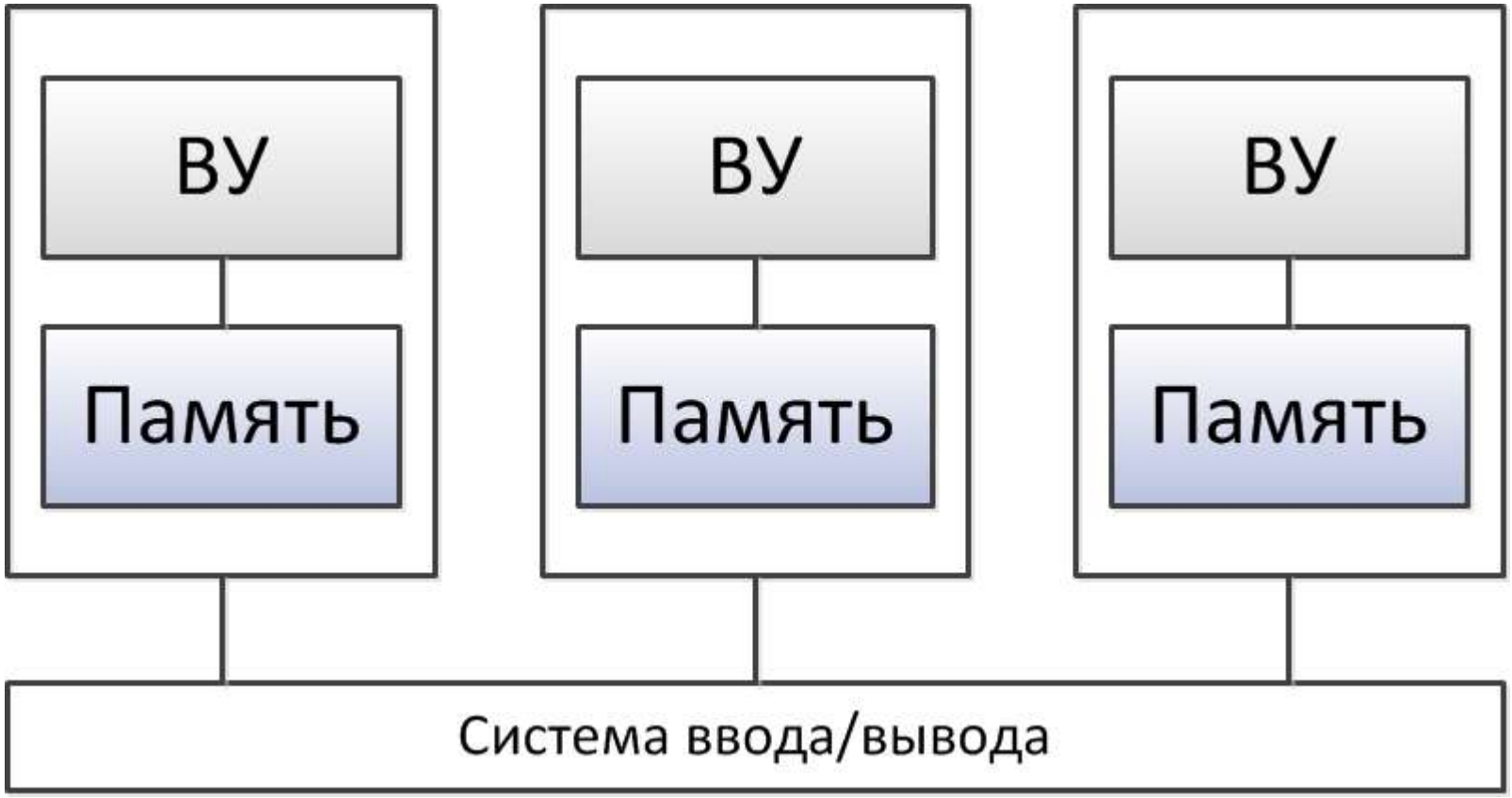
Преимущества

- Простота и универсальность программирования. Нет ограничений на модель программирования
- Простота эксплуатации и тех. обслуживания
- Невысокая цена комплектующих

Недостатки

- Плохая масштабируемость

Massive parallel processing (MPP)



Особенности MPP-архитектуры

Преимущества

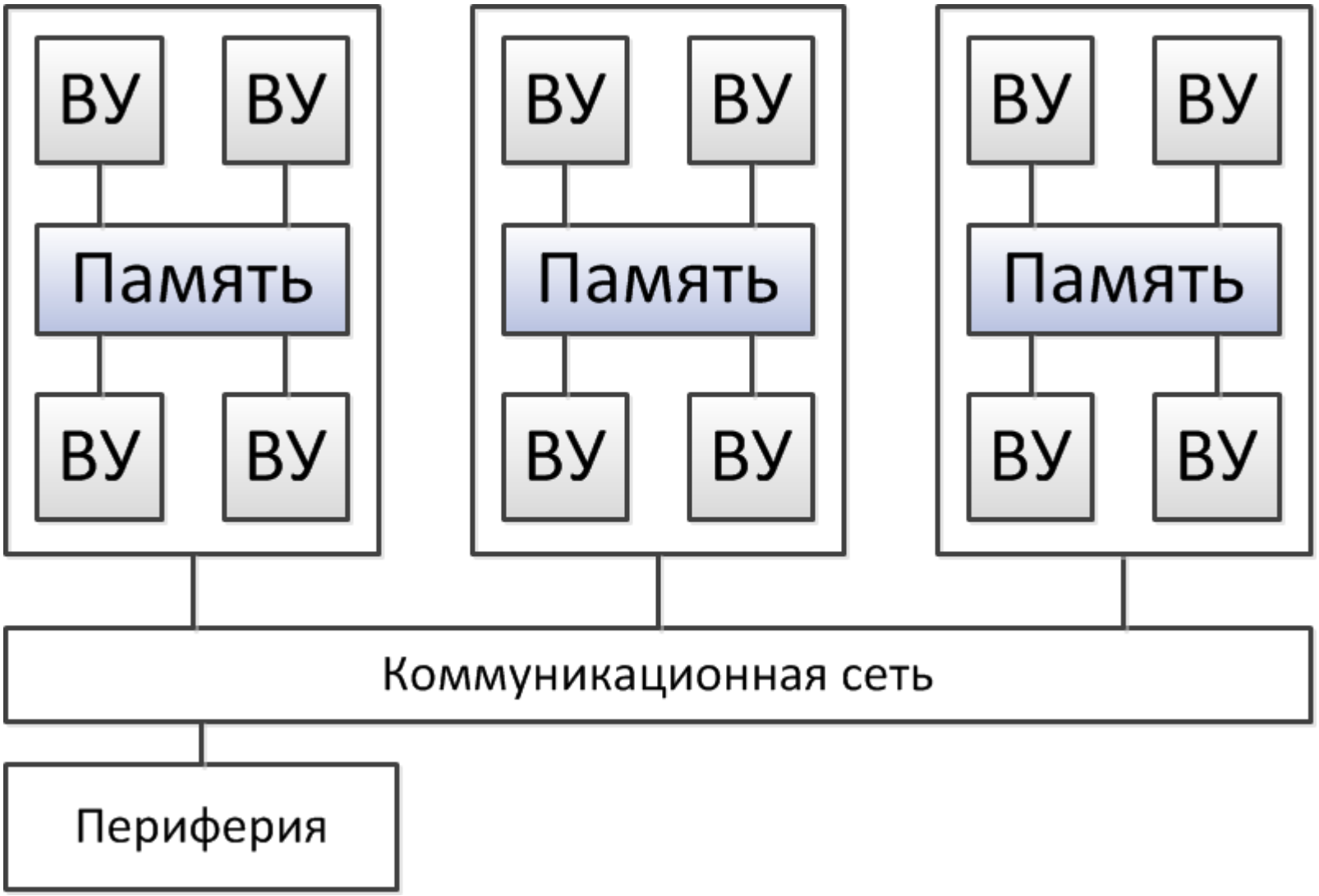
- Хорошая масштабируемость
- Большинство рекордов производительности устанавливаются на машинах MPP-архитектуры

Недостатки

- Отсутствие общей памяти снижает скорость межпроцессорного обмена
- Каждый процессор может использовать только ограниченный объем банка памяти
- Высокая цена программного обеспечения

Гибридная архитектура NUMA

(NonUniform Memory Access)



Когерентность памяти

Когерентность кэшей процессоров:
все процессоры получают одинаковые значения одних переменных в любой момент времени

Кластерные системы

Набор рабочих станций (или даже ПК) общего назначения, используется в качестве дешевого варианта MPP-архитектуры

Для связи узлов используется одна из стандартных сетевых технологий на базе шинной архитектуры или коммутатора (к примеру Ethernet)

При объединении в кластер компьютеров разной мощности или разной архитектуры, говорят о **гетерогенных** (неоднородных) кластерах

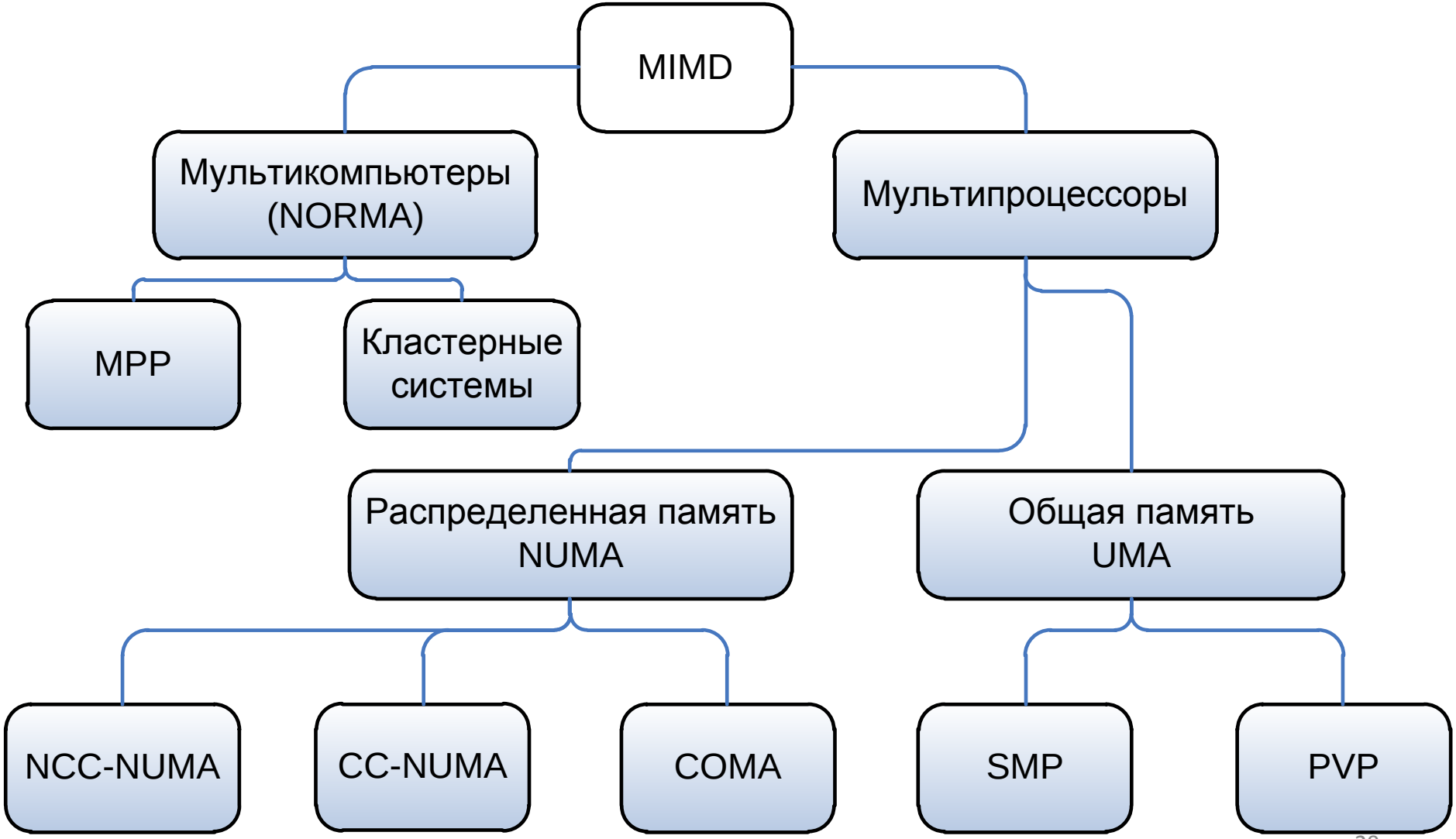
Узлы кластера могут одновременно использоваться в качестве пользовательских рабочих станций. (несколько функций у узлов кластера)

Программирование, как правило, в рамках модели передачи сообщений (MPI), технологии PVM (Parallel Virtual Machine)

Дешевизна подобных систем оборачивается большими накладными расходами на взаимодействие параллельных процессов между собой, что сильно сужает потенциальный класс решаемых задач

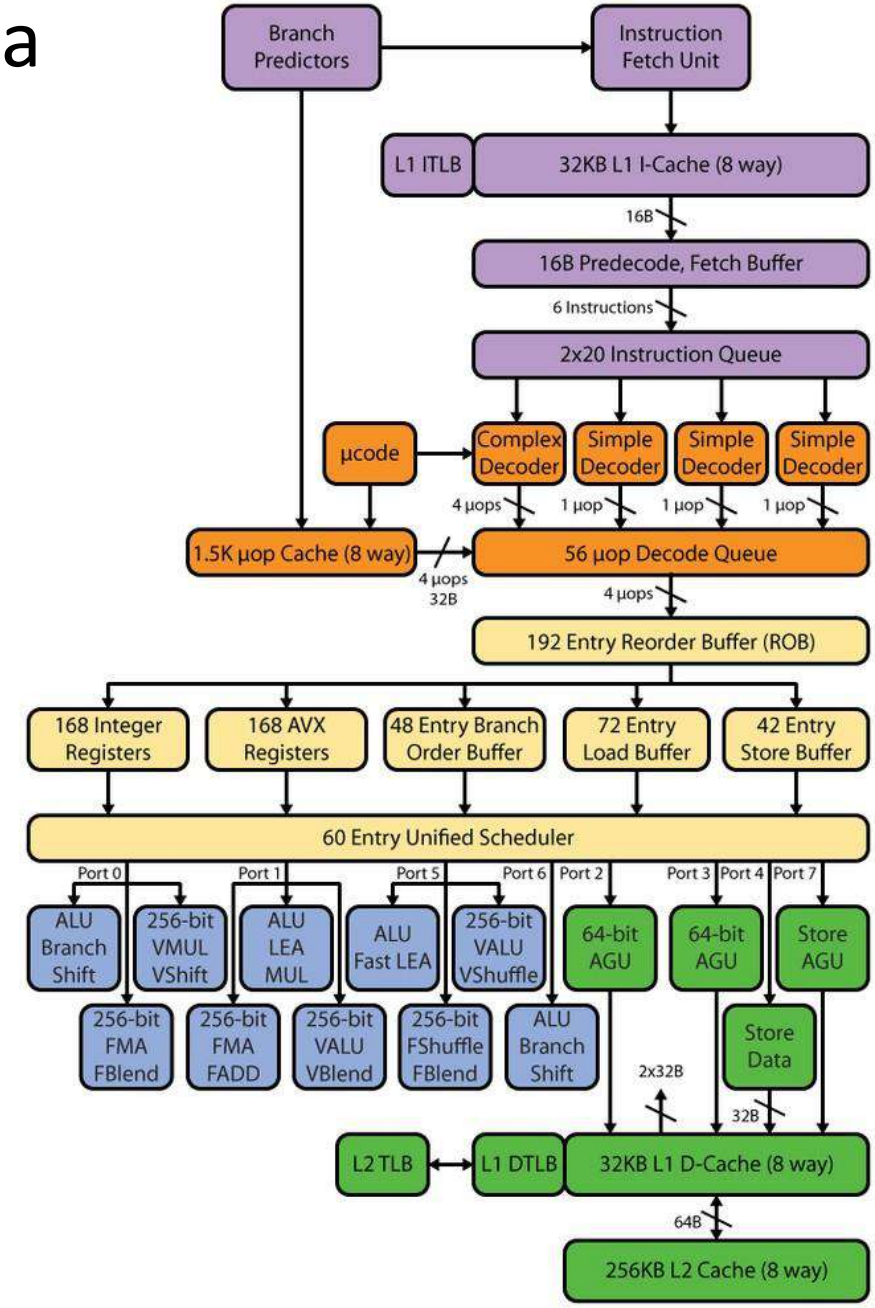
Кластер. Тип Beowulf

Разделение класса MIMD по типу доступа к памяти



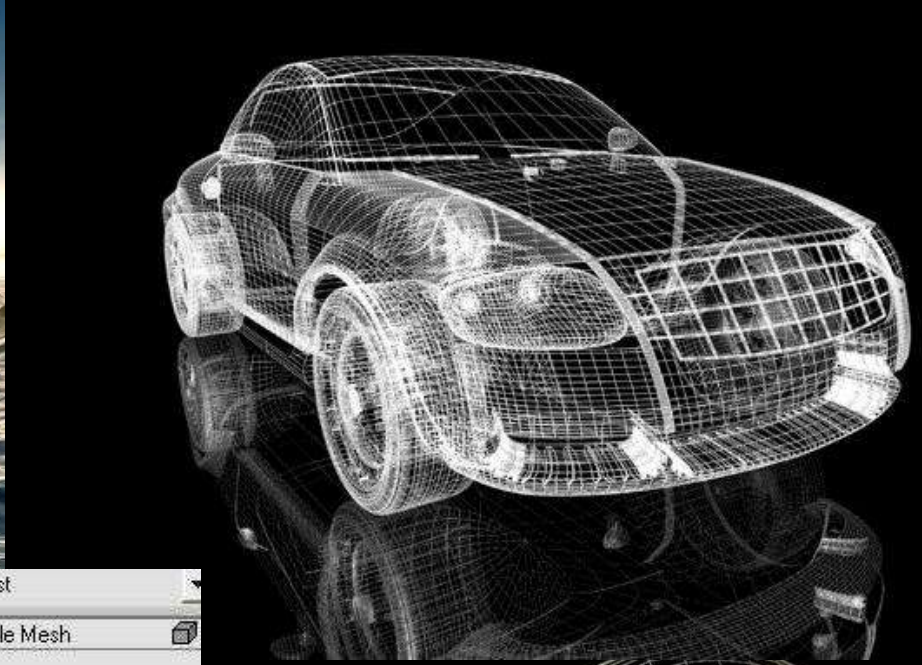
Архитектура процессора Intel Haswell

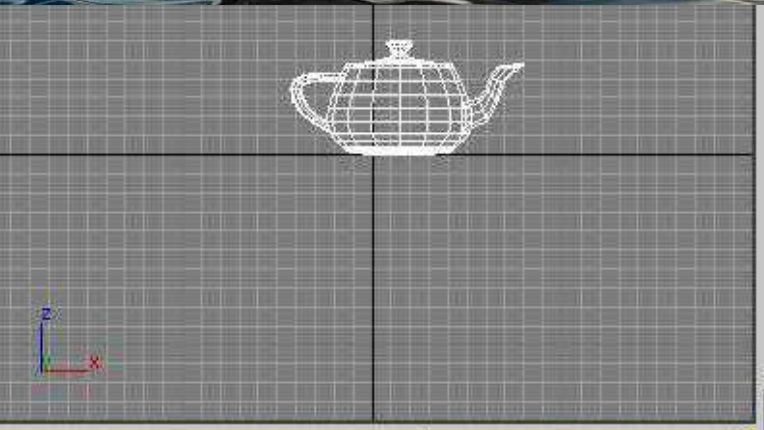
ВУ

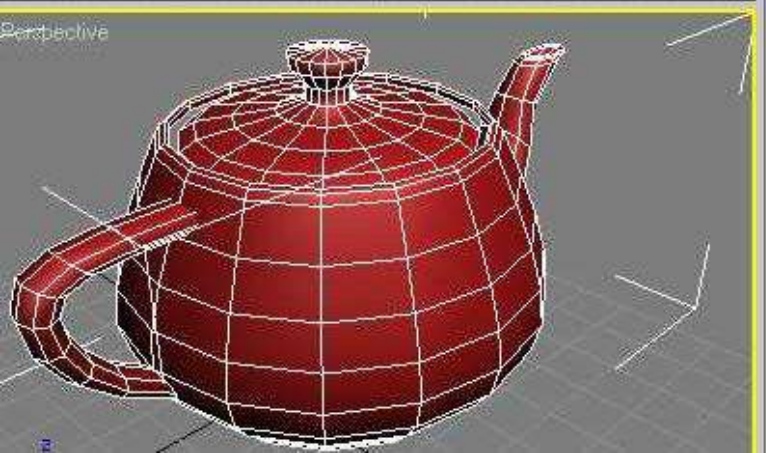


Процессор

Видеокарта







Modifier List

Editable Mesh

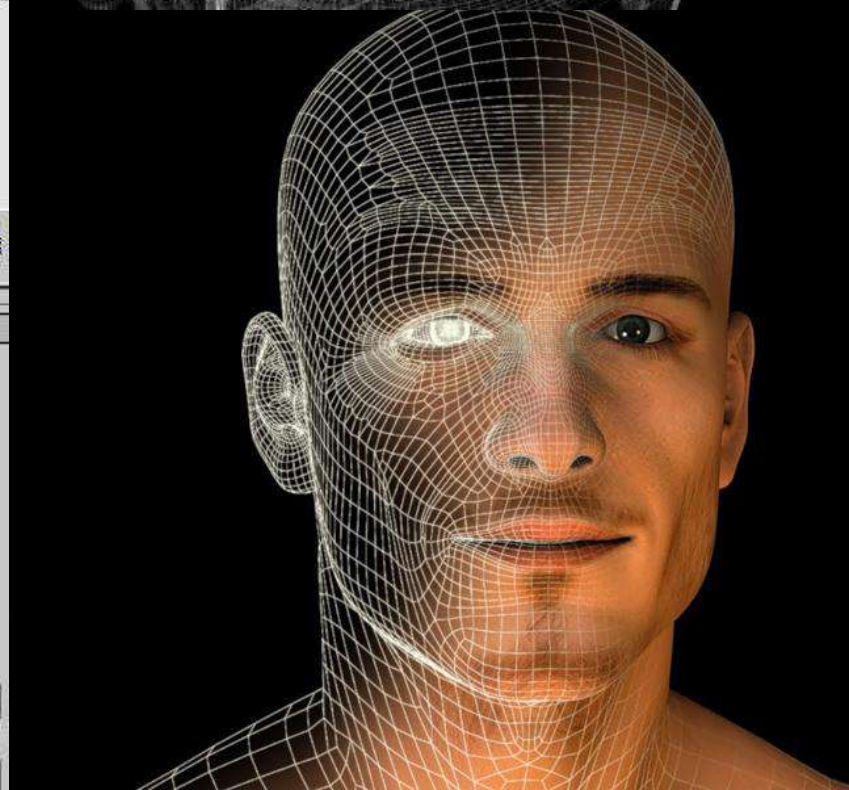
Selection

By Vertex
Ignore Backfacing
Ignore Visible Edges
Planar Thresh: 45,0
Show Normals
Scale: 20,0
Delete Isolated Vertices

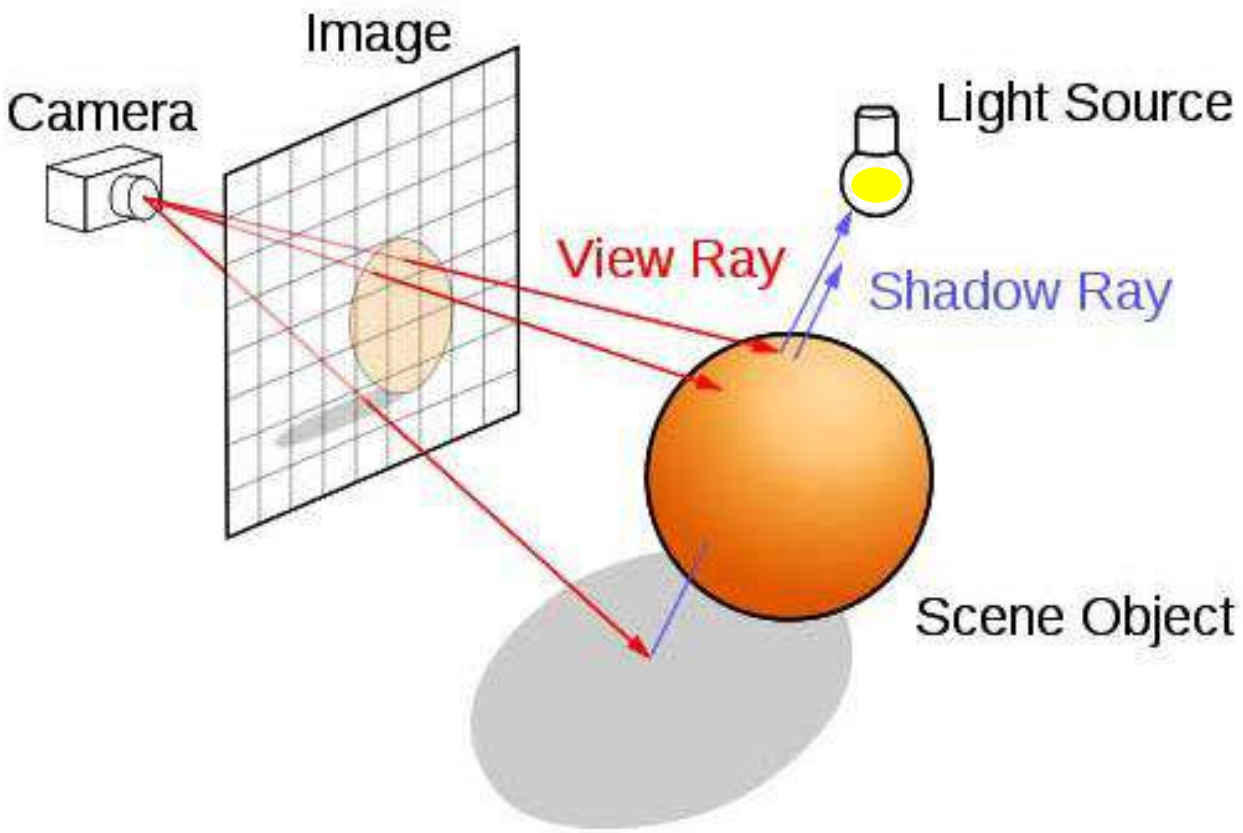
Hide | Unhide All

Named Selections:

Copy | Paste



Трассировка лучей



Виды видеокарт

- **Графика с разделяемой памятью** (*Shared graphics, Shared Memory Architecture*). Видеопамять в виде специализированных ячеек как таковая отсутствует; вместо этого под нужды видеоадаптера динамически выделяется область основной оперативной памяти компьютера. (интегрированные видеокарты, являющиеся частью северного моста).
  Преимущества данного решения — низкая цена и малое энергопотребление.
  Недостатки — невысокая производительность в 3D-графике и отрицательное влияние на пропускную способность памяти.
- **Дискретная графика** (*Dedicated graphics*). Есть своя видеопамять. Дискретная графика обеспечивает наивысшую производительность в трёхмерной графике. (выполнены в виде отдельных плат)
  Недостатки: более высокая цена и большее энергопотребление.
  Гибридная дискретная графика (*Hybrid graphics*). Комбинация вышеназванных способов, ставшая возможной с появлением шины PCIExpress. Небольшой объём физически распаянной на плате видеопамяти, который может расширяться за счёт использования основной оперативной памяти(ОЗУ).
- **Специализированные видеокарты**. Применяются для высокопроизводительных вычислений.
  Для вывода изображения на монитор не используется.

Характеристики видеокарты

- **ширина шины памяти**, измеряется в битах — количество бит информации, передаваемой за такт. Важный параметр в производительности карты.
- **объём видеопамяти**, измеряется в мегабайтах — объём собственной оперативной памяти видеокарты.
  частота ядра — измеряются в мегагерцах
- **частота памяти** — измеряются в мегагерцах
- **текстурная скорость заполнения** , измеряется в млн. пикселов в секунду, показывает количество выводимой информации в единицу времени
- **пиксельная скорость заполнения**, измеряется в млн. пикселов в секунду, показывает количество выводимой информации в единицу времени.

Технологический процесс изготовления, нм	14
Площадь кристалла, мм²	135
Количество транзисторов, млн	3300
Количество скалярных процессоров (ядер)	640
Количество кластеров обработки графики (GPC)	2
Количество блоков мультипроцессоров (SM)	5
Количество текстурных блоков (TMU)	40
Количество блоков растеризации (ROP)	32
Заполнение сцены, млрд пикс/с	43,3
Заполнение сцены, млрд текс/с	54,1
Разрядность шины видеопамяти, бит	128
Стандарт видеопамяти	GDDR5
Объём видеопамяти, МБ	2048
Пропускная способность шины памяти, ГБ/с	112,0
Интерфейс	PCI Express 3.0 x16
Энергопотребление, Вт	75
Частота ядра, МГц	1354
Частота в режиме Turbo Boost, МГц	1455
Реальная (Номинальная) частота видеопамяти, МГц	1750 (7000)
Производительность FP32, GFLOPS	1733,1
Производительность FP64, GFLOPS	54,1
Производительность FP16, GFLOPS	27,0
Поддержка версий API	Direct3D 12_1, OpenGL 4.5, Vulkan 1.0, OpenCL 2.0
Поддержка версии Shader Model	Shader Model 5.0

Типы шейдеров

- **Вершинные шейдеры (Vertex Shader)**
  Вершинный шейдер оперирует данными, сопоставленными с вершинами многогранников. К таким данным, в частности, относятся координаты вершины в пространстве, текстурные координаты, тангенс-вектор, вектор бинормали, вектор нормали. Вершинный шейдер может быть использован для видового и перспективного преобразования вершин, генерации текстурных координат, расчета освещения и т. д.
- **Геометрические шейдеры (Geometry Shader)**
  Геометрический шейдер, в отличие от вершинного, способен обработать не только одну вершину, но и целый примитив. Это может быть отрезок (две вершины) и треугольник (три вершины), а при наличии информации о смежных вершинах (adjacency) может быть обработано до шести вершин для треугольного примитива. Кроме того, геометрический шейдер способен генерировать примитивы «на лету», не задействуя при этом центральный процессор. Впервые начал использоваться на видеокартах Nvidia серии 8.
- **Пиксельные шейдеры (Pixel Shader)**
  Фрагментный шейдер работает с фрагментами изображения. Под фрагментом изображения в данном случае понимается пиксель, которому поставлен в соответствие некоторый набор атрибутов, таких как цвет, глубина, текстурные координаты. Фрагментный шейдер используется на последней стадии графического конвейера для формирования фрагмента изображения.

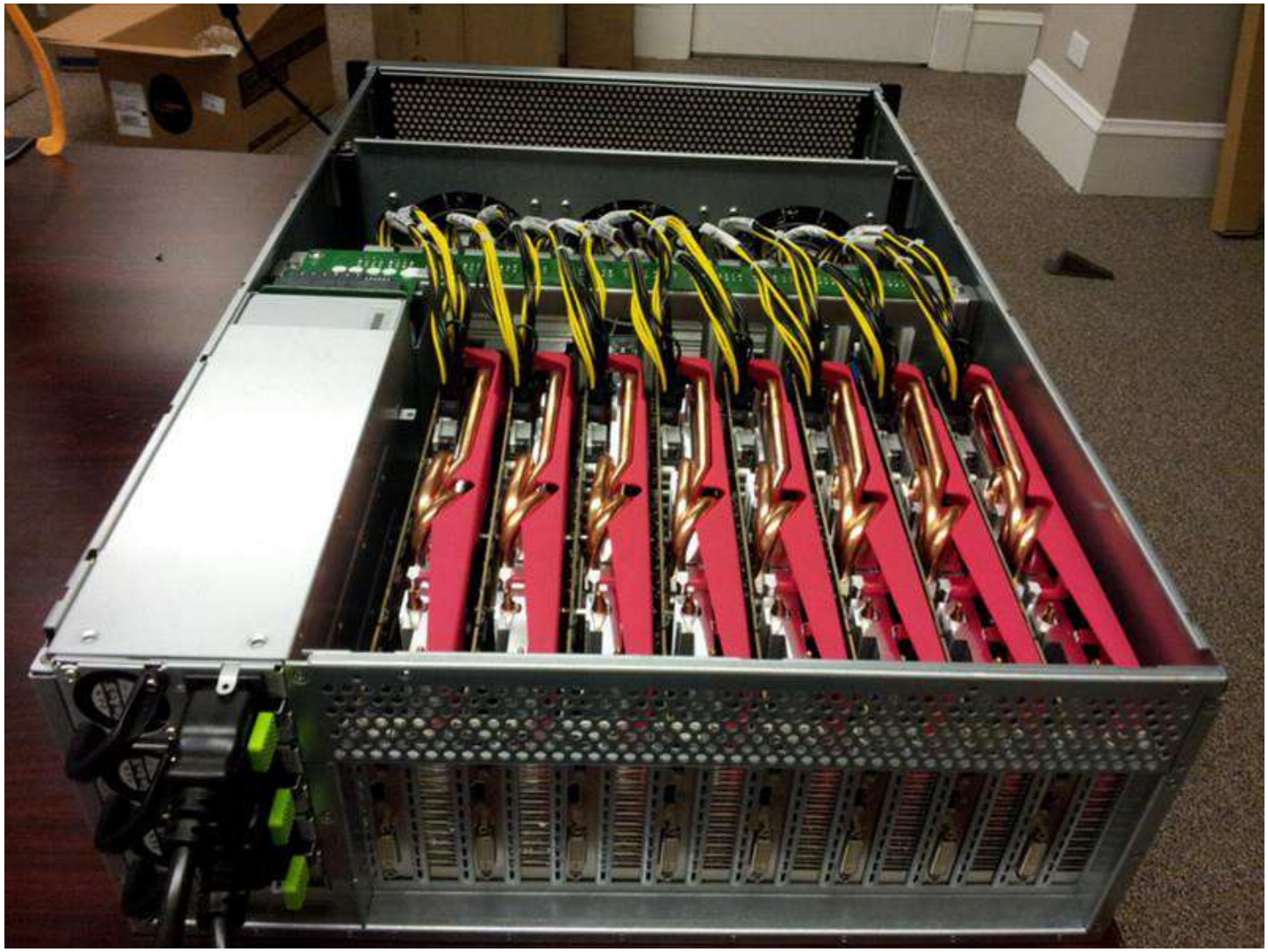
Особенности версий DirectX

DirectX 6.0 — мультитекстурирование
DirectX 7.0 — аппаратная поддержка преобразований, обрезания и освещения
DirectX 8.0 — шейдерная модель 1.1
DirectX 8.1 — пиксельные шейдеры 1.4 и вершинные шейдеры 1.1
DirectX 9.0 — шейдерная модель 2.0
DirectX 9.0b — пиксельные шейдеры 2.0b и вершинные шейдеры 2.0
DirectX 9.0c — шейдерная модель 3.0
DirectX 9.0L — версия DirectX 9.0 для Windows Vista
DirectX 10 — шейдерная модель 4.0 (только Windows Vista, Windows 7, Windows 8)
DirectX 10.1 — шейдерная модель 4.1 (только Windows Vista, Windows 7, Windows 8)
DirectX 11 — шейдерная модель 5.0 (только Windows Vista, Windows 7, Windows 8)
DirectX 11.3/DirectX 12 — шейдерная модель 5.1 (только Windows 10).

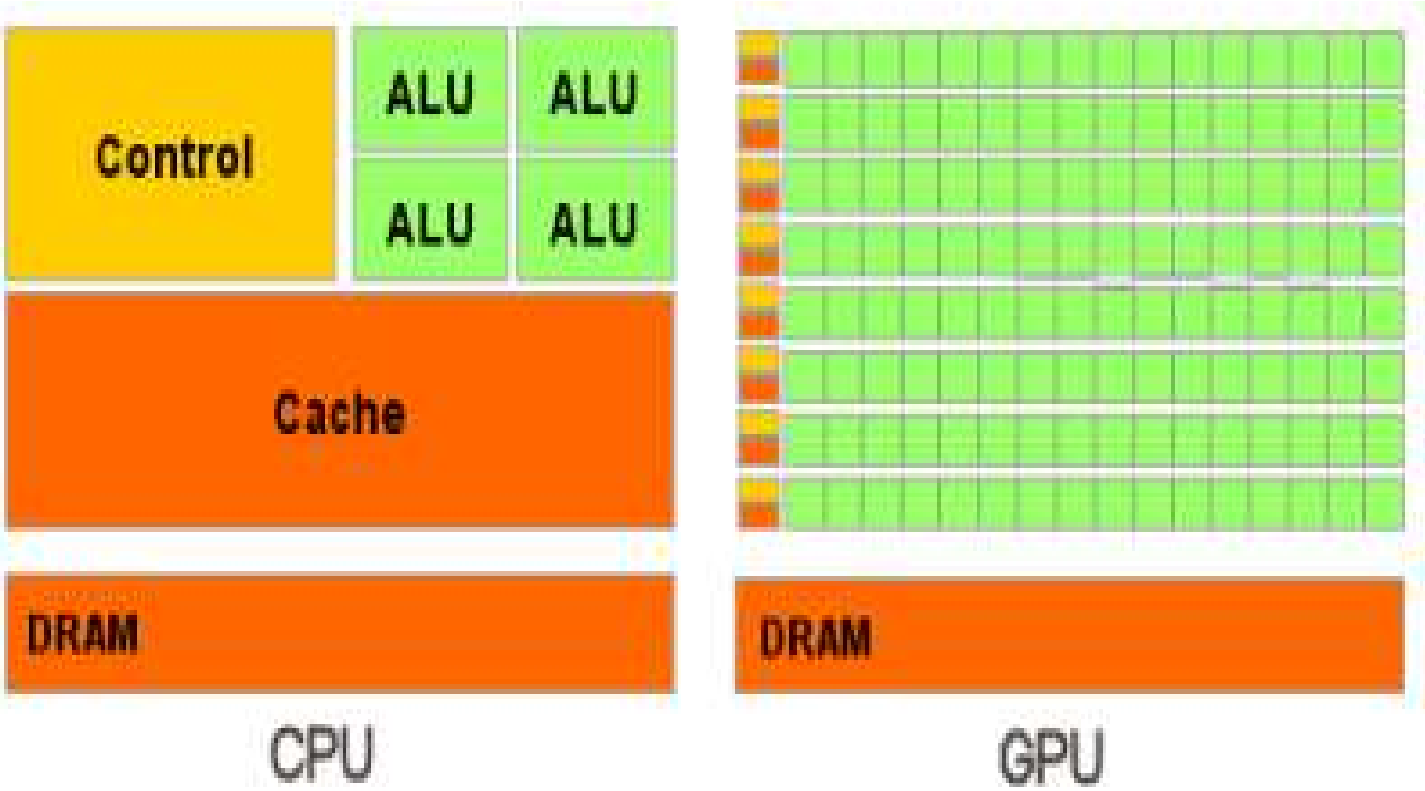
Вычисления
на графических картах

Видеокарты для вычислений

Пример серв. платформы с GPU



Причины использования GPU для вычислений

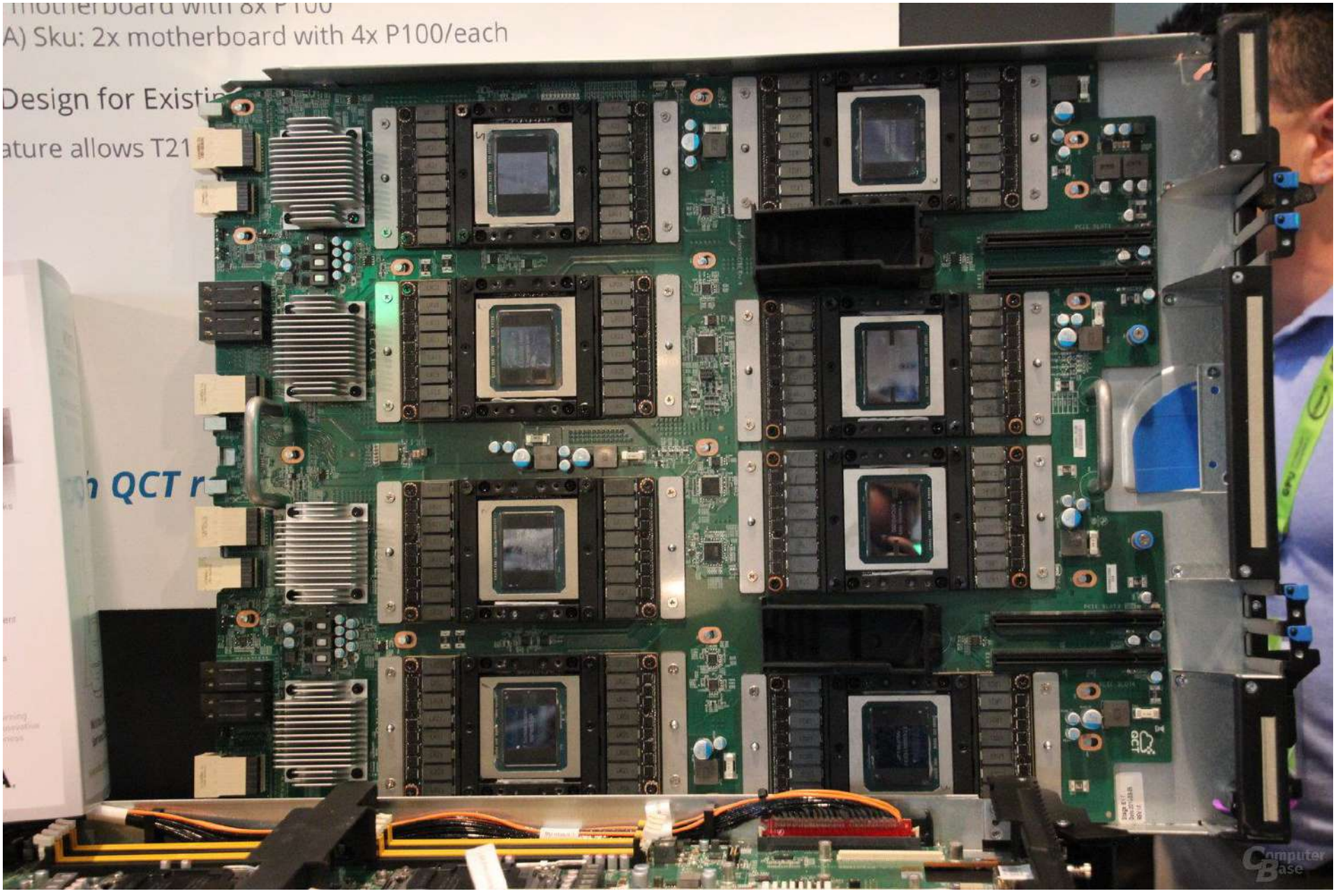


Инструменты для GPGPU

CUDA —фреймворк для видеокарт NVIDIA (аппаратно-зависимая платформа)
Firestream — фреймворк для GPGPU вычислений на видеокартах AMD. (аппаратно-зависимая платформа)
HLSL — шейдерный язык DirectX (аппаратно-зависимая платформа)
DirectCompute — прикладной язык программирования (DirectX)
GLSL — шейдерный язык OpenGL

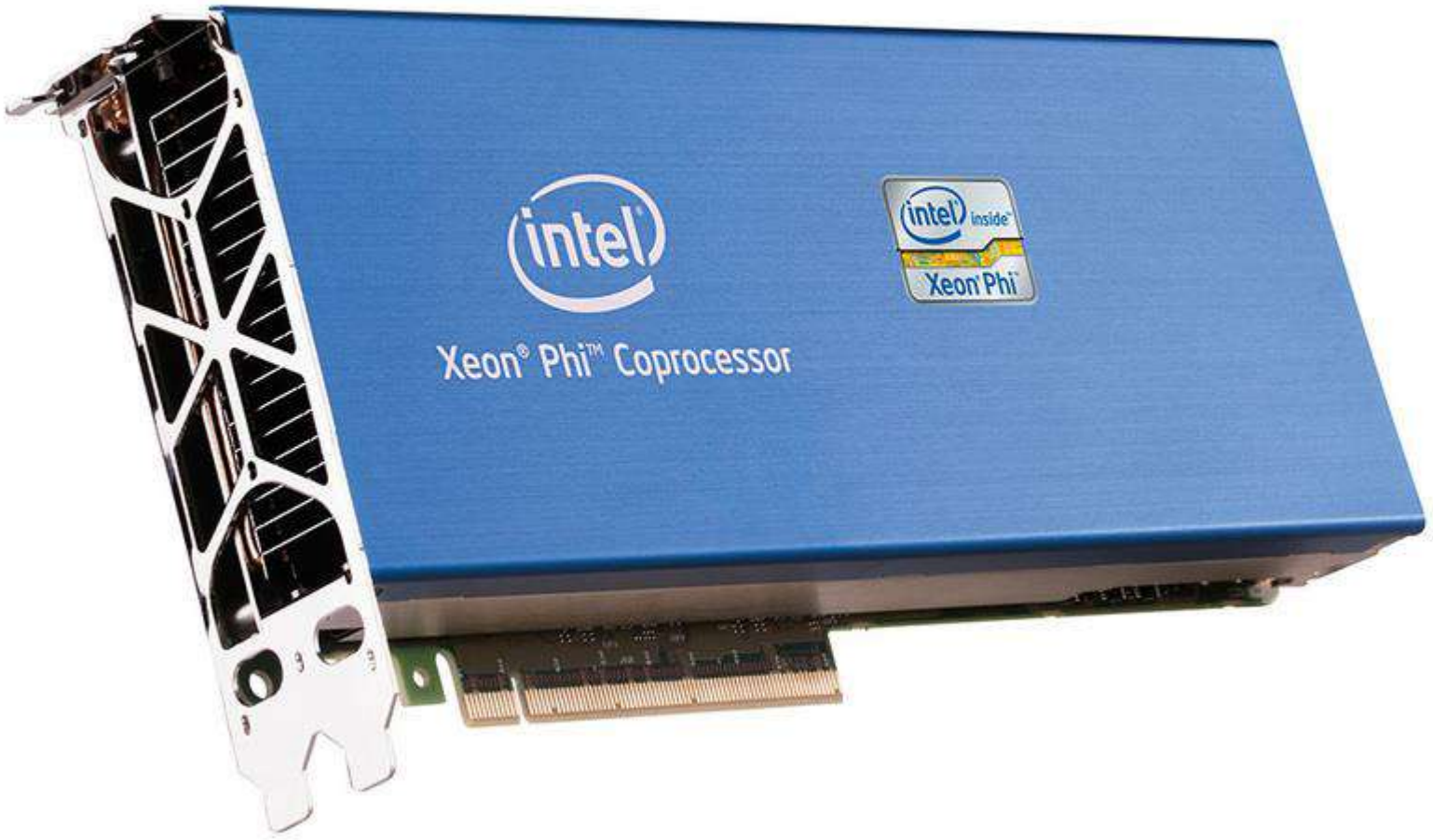
OpenCL — аппаратно и программно-независимая платформа для вычислений на CPU и GPU

Изменение форм-фактора


A) Sku: 2x motherboard with 4x P100/each
Design for Existi
ature allows T21
QCT r

Intel Xeon Phi

Сопроцессор Intel Xeon Phi



Архитектура MIC

- **Intel MIC** (*Many Integrated Core Architecture*) — архитектура многоядерной процессорной системы, разработанная Intel

В основе архитектуры MIC лежит классическая архитектура x86, на ускорителе исполняется ОС Linux.
Для программирования MIC предполагается использовать OpenMP, OpenCL, спец. компиляторы Intel Fortran, Intel C++.

От предыдущих разработок унаследованы:

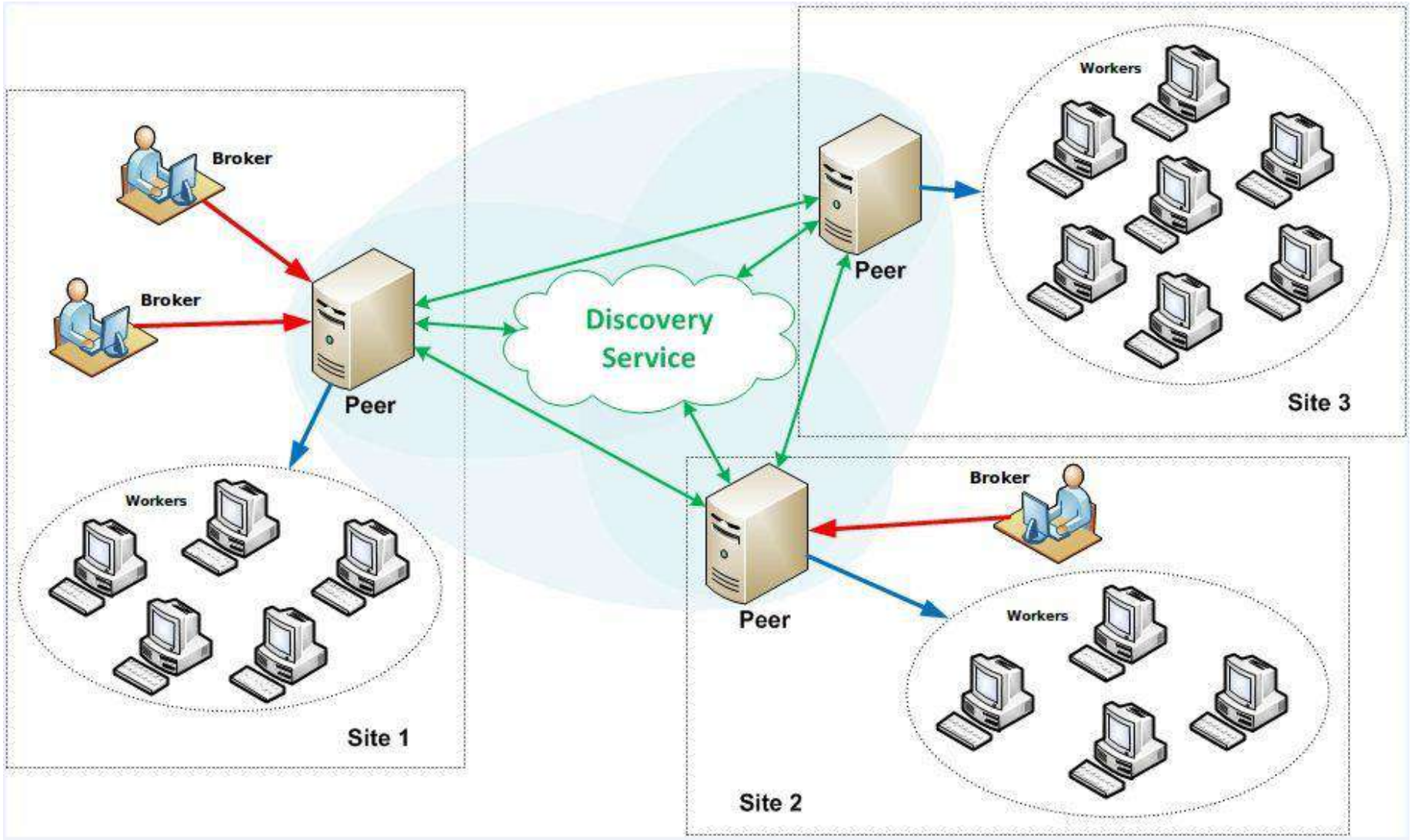
- набор команд x86;
- 512-битные векторные АЛУ;
- когерентный L2 кеш размером 512 КБ на ядро;
- кольцевая шина для связи ядер и контроллера памяти.

Поколения Intel Xeon Phi

1. Knights Ferry (2010) 45 нм
2. Knights Corner (2012) 22 нм
3. Knights Landing (2015) 14 нм
4. Knights Hill (?) 10 нм

Грид-системы

Грид-система OurGrid



Особенности грид-систем

- Гетерогенность узлов распределенной системы;
- Автономность расчетов на различных узлах и невозможность постоянной координации расчетов между узлами;
- Ненадежность связей и возможное отключение вычислительных узлов;
- Непостоянное время непрерывной работы узла и трудность расчета длительных заданий;
- Наличие ошибок и задержек при расчетах.

Bag of tasks

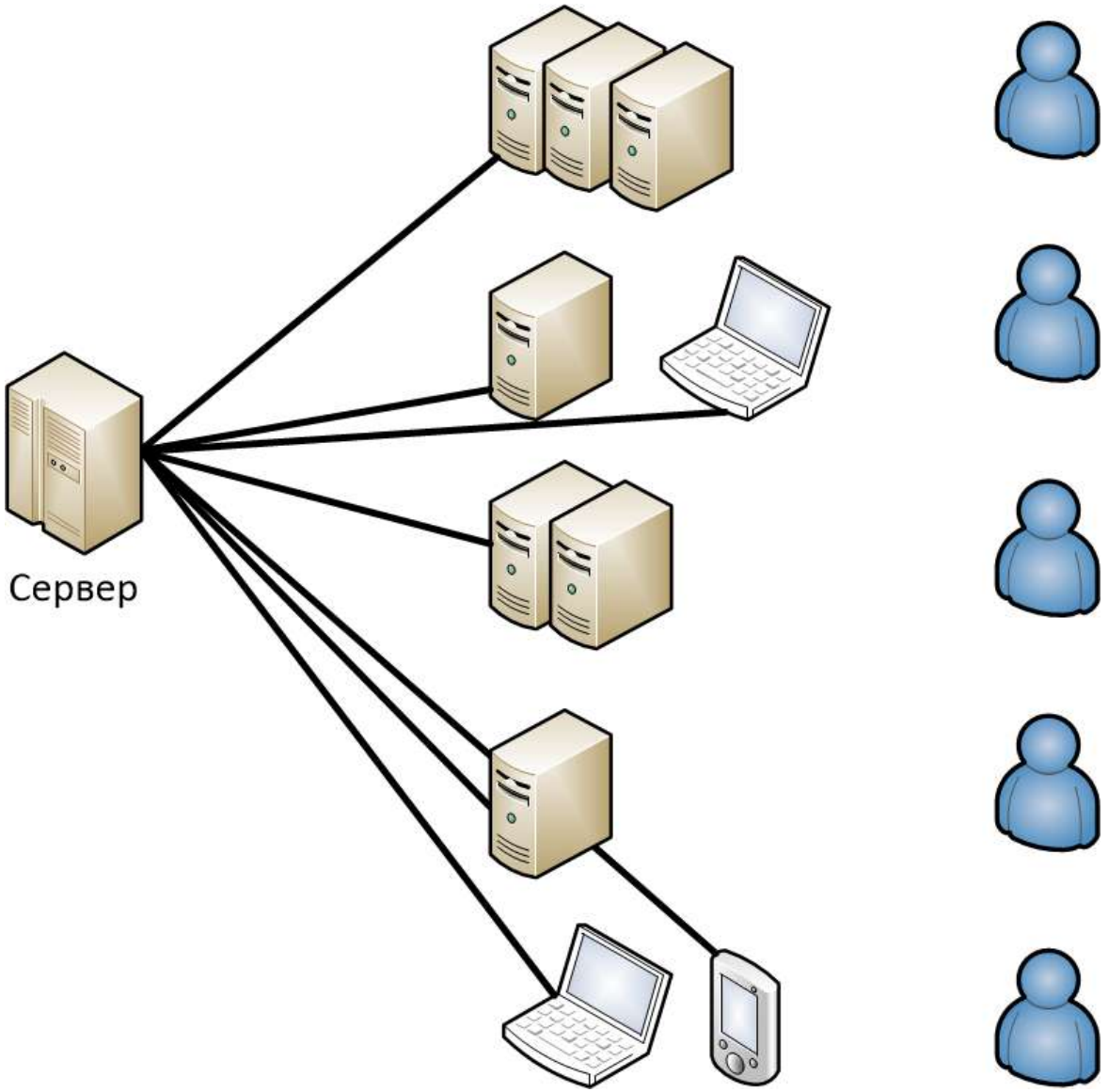
Задача может разбиваться на множество независимых подзадач. Каждая подзадача будет рассчитываться на отдельном вычислительном узле распределенной системы.

Для различных подзадач используются различные наборы входных данных и единый алгоритм их обработки. Такой тип задач в литературе называется «bag of tasks» или задача, разделяемая по данным.

В качестве примера таких задач можно привести задачи:

- Задачи комбинаторики и полного перебора;
- SAT-задачи;
- задачи машинного обучения,
- задачи имитационного математического моделирования
- и др.

Грид-система из персональных компьютеров



Платформы для организации распределенных вычислений

- HTCondor

- Globus

- BOINC

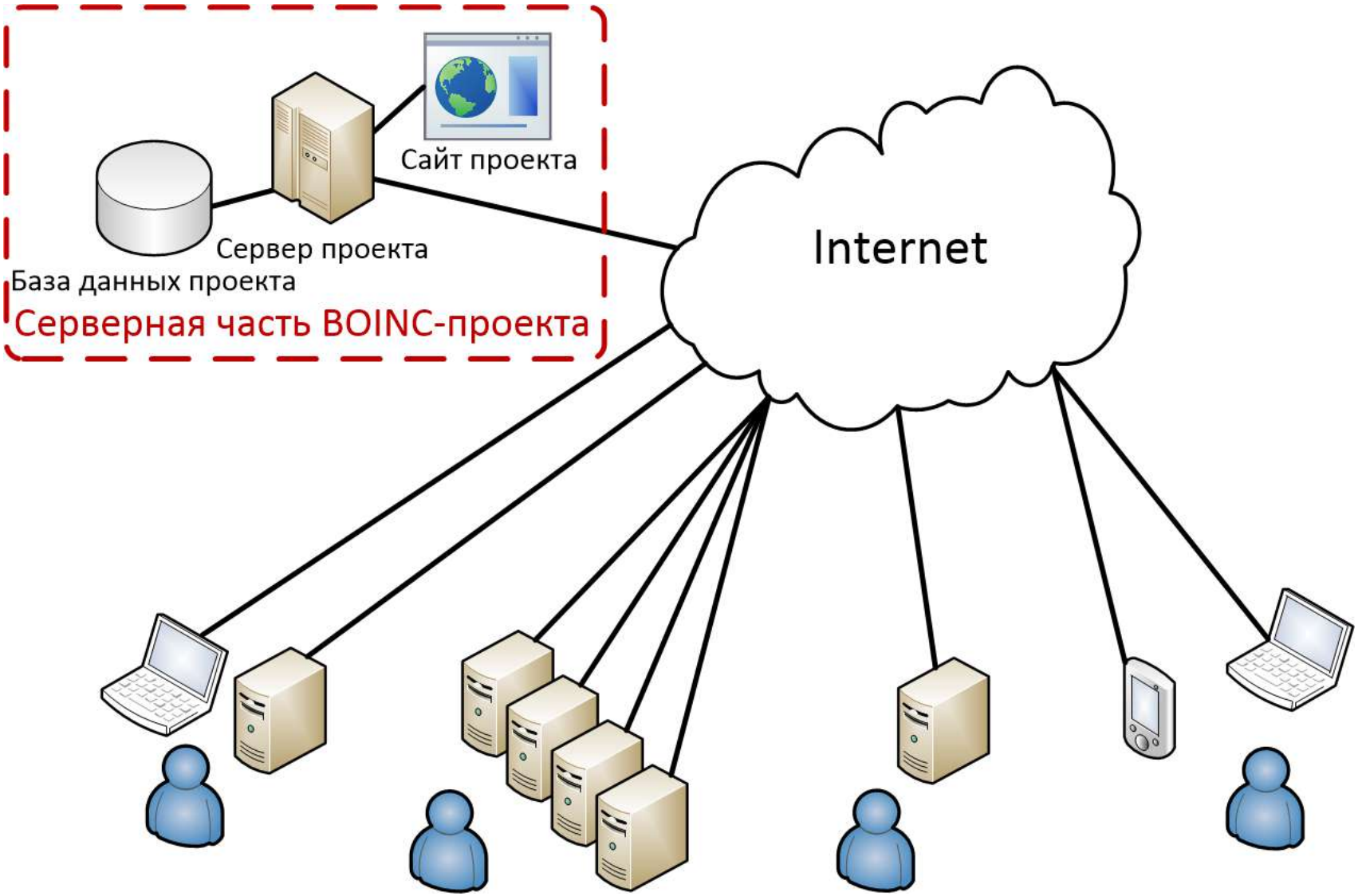
Платформа BOINC

BOINC – Berkeley Open Infrastructure for Network Computing

Платформа для организации ~~добровольных~~ распределенных вычислений:

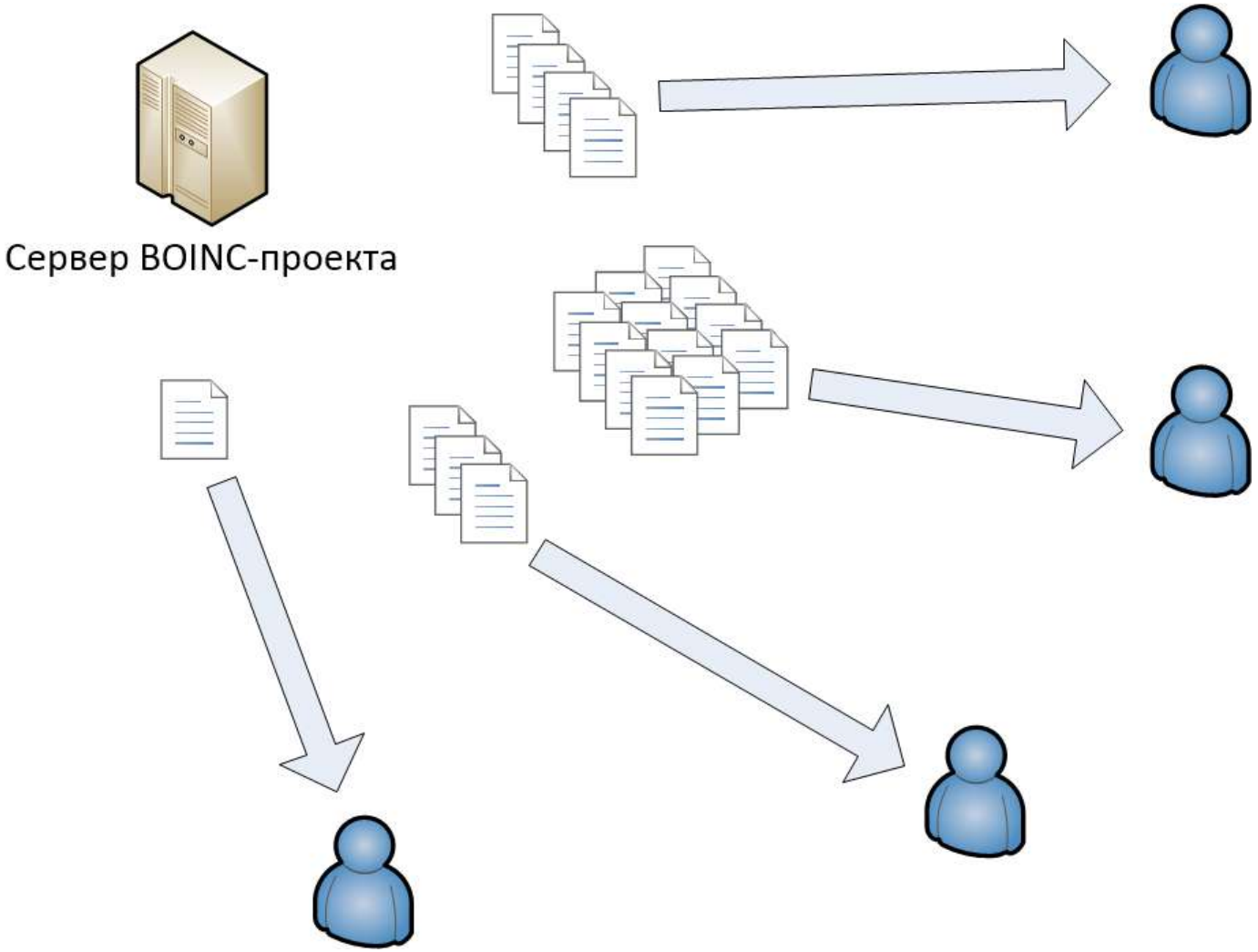
- состоит из серверной и клиентской части;
- дает возможность задействовать вычислительные мощности персональных компьютеров(ПК);
- Кросплатформенная клиентская часть;
- Гибкая настройка клиентской части для эффективного использования ресурсов ПК.

Схема подключения пользователей к проекту

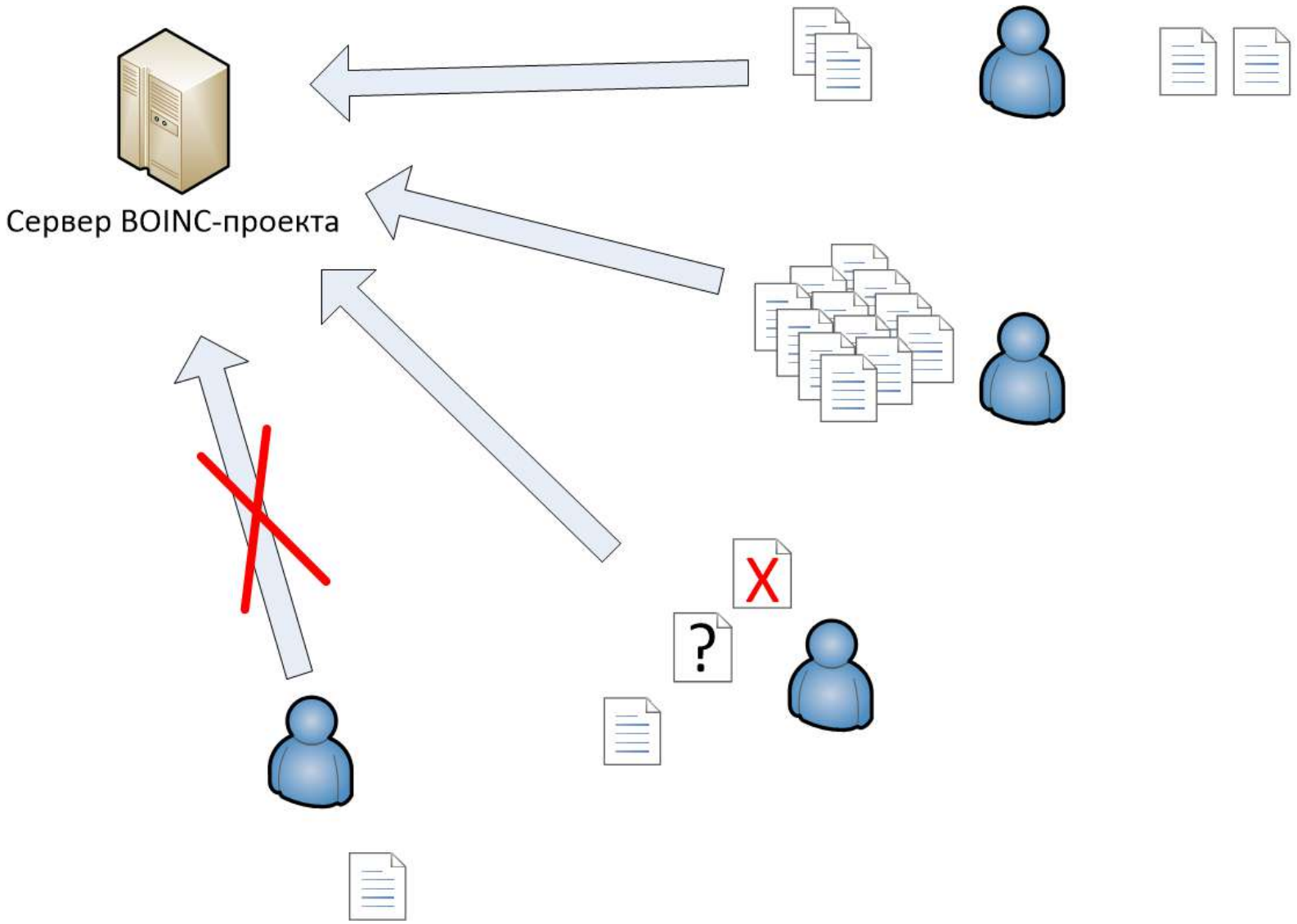


Обмен данными

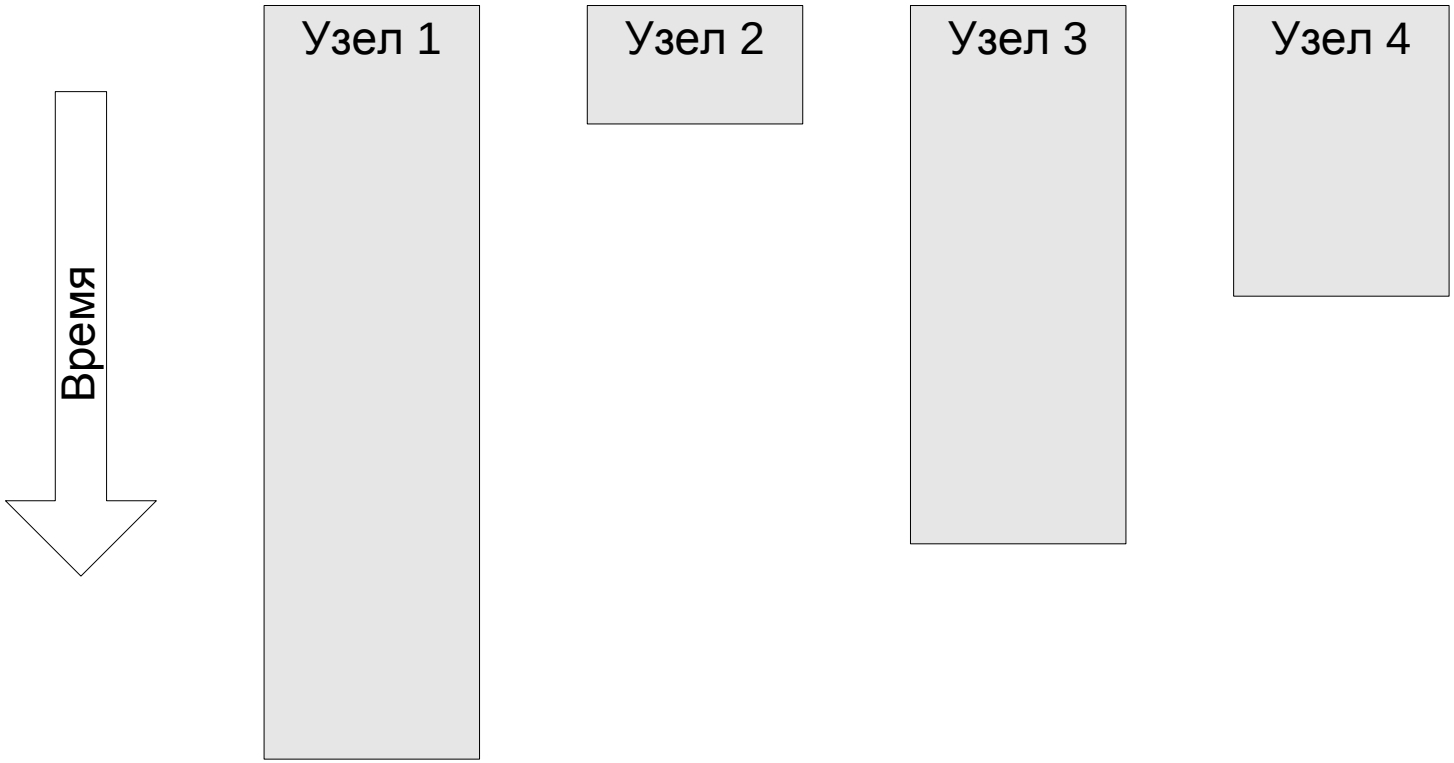
Выдача заданий участникам проекта



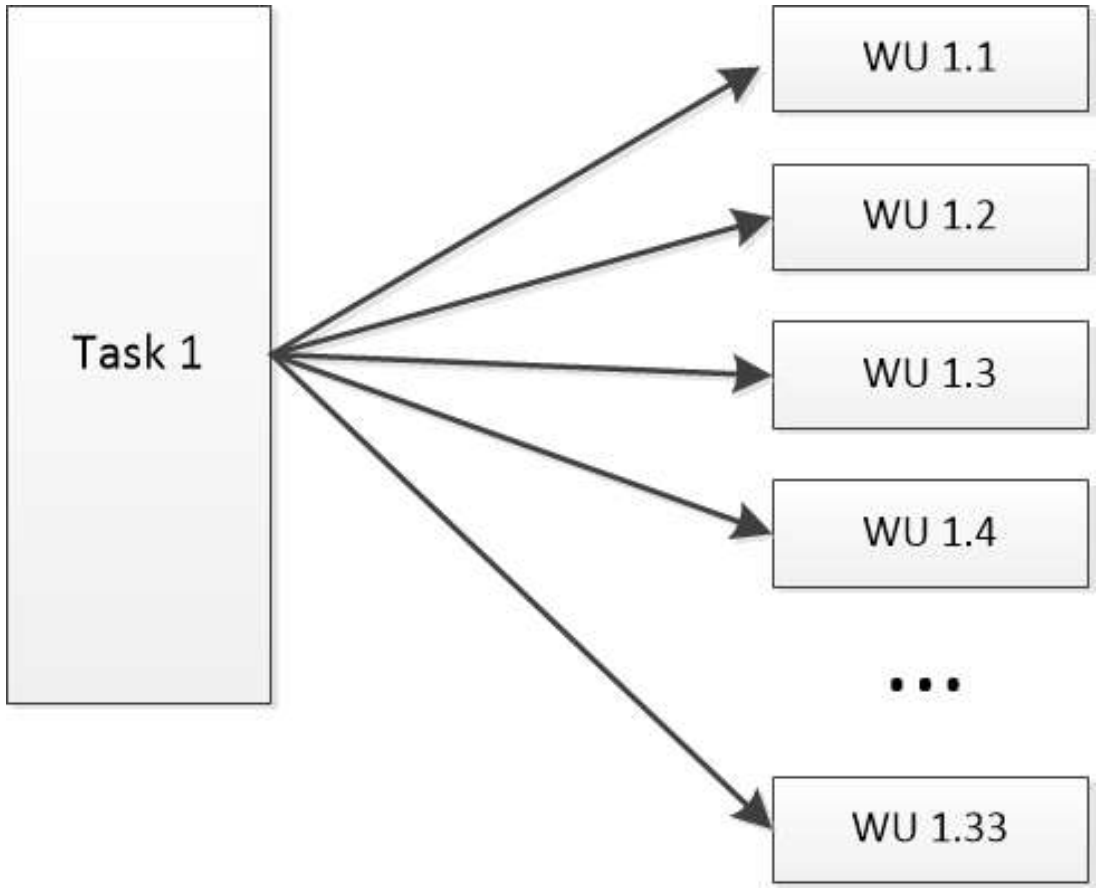
Получение результатов



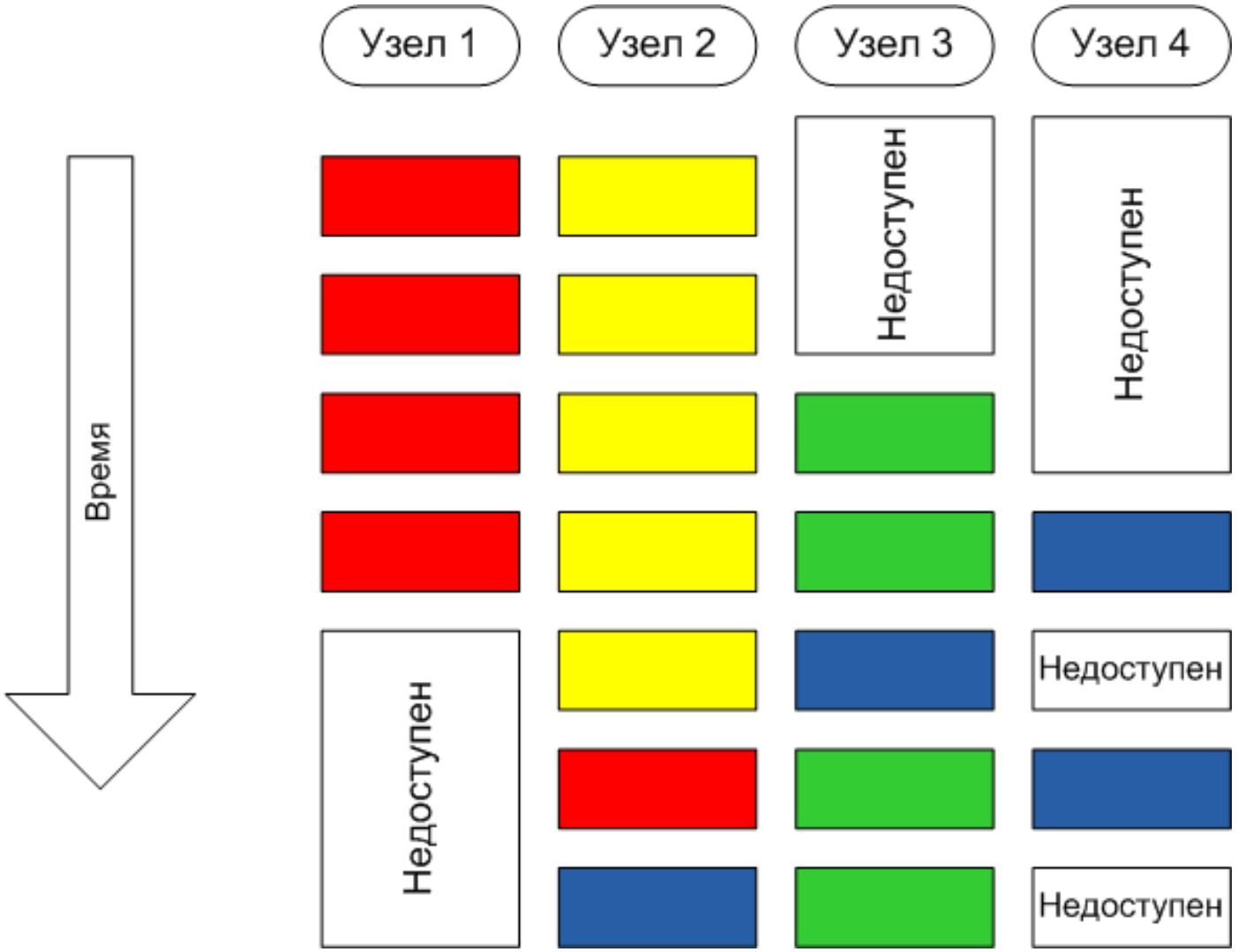
Пример причины потери эффективности при распределенной реализации



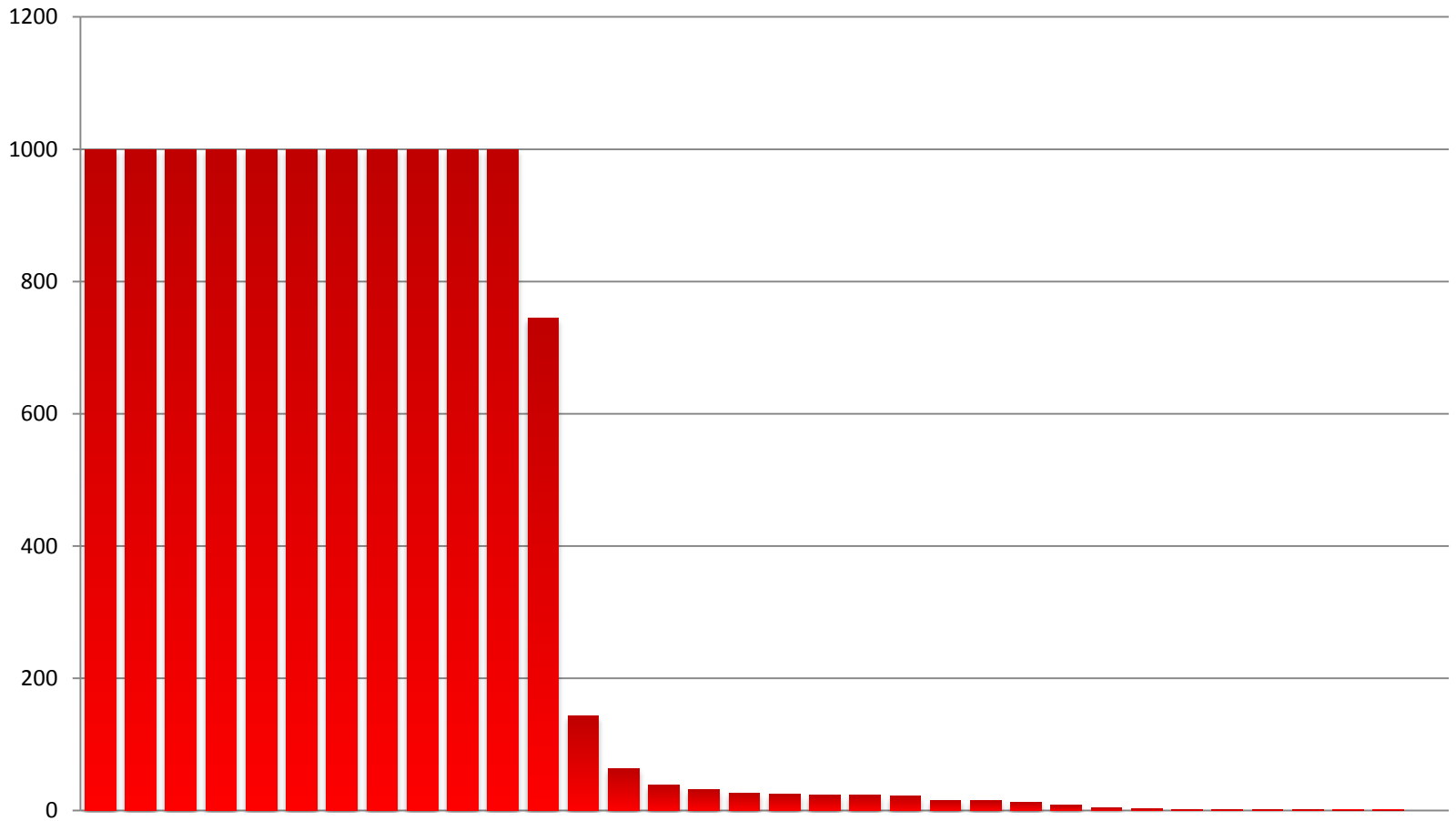
Большое задание разделяется на много небольших подзаданий



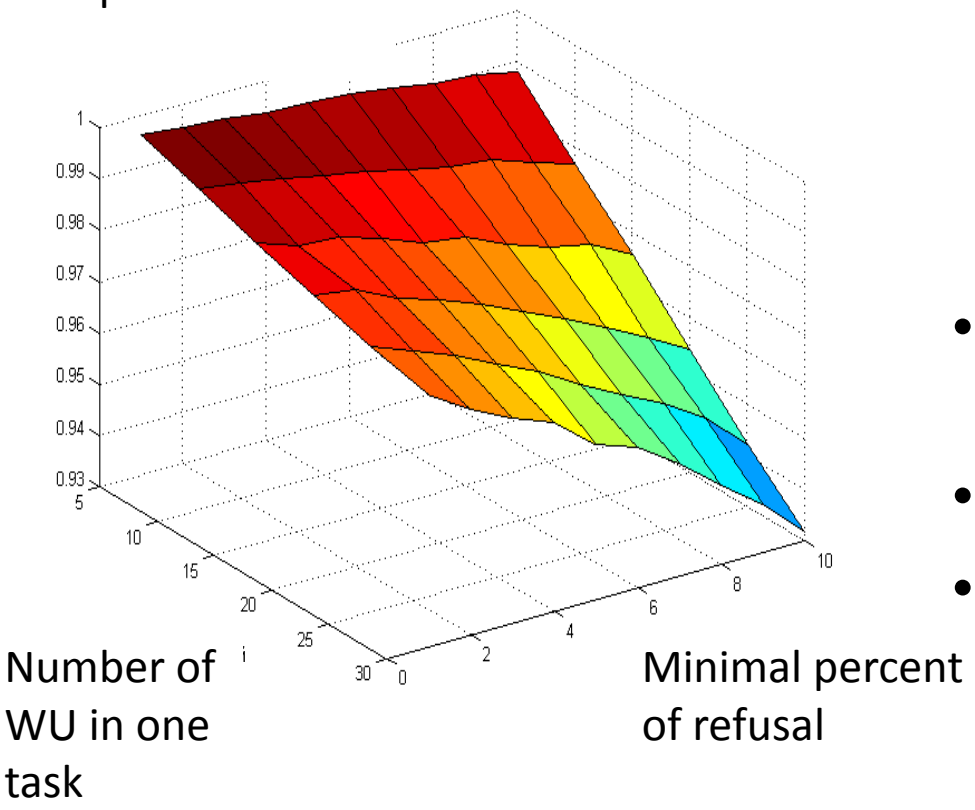
Разделение на подзадачи



Образование «хвостов»



Подбор параметров репликации

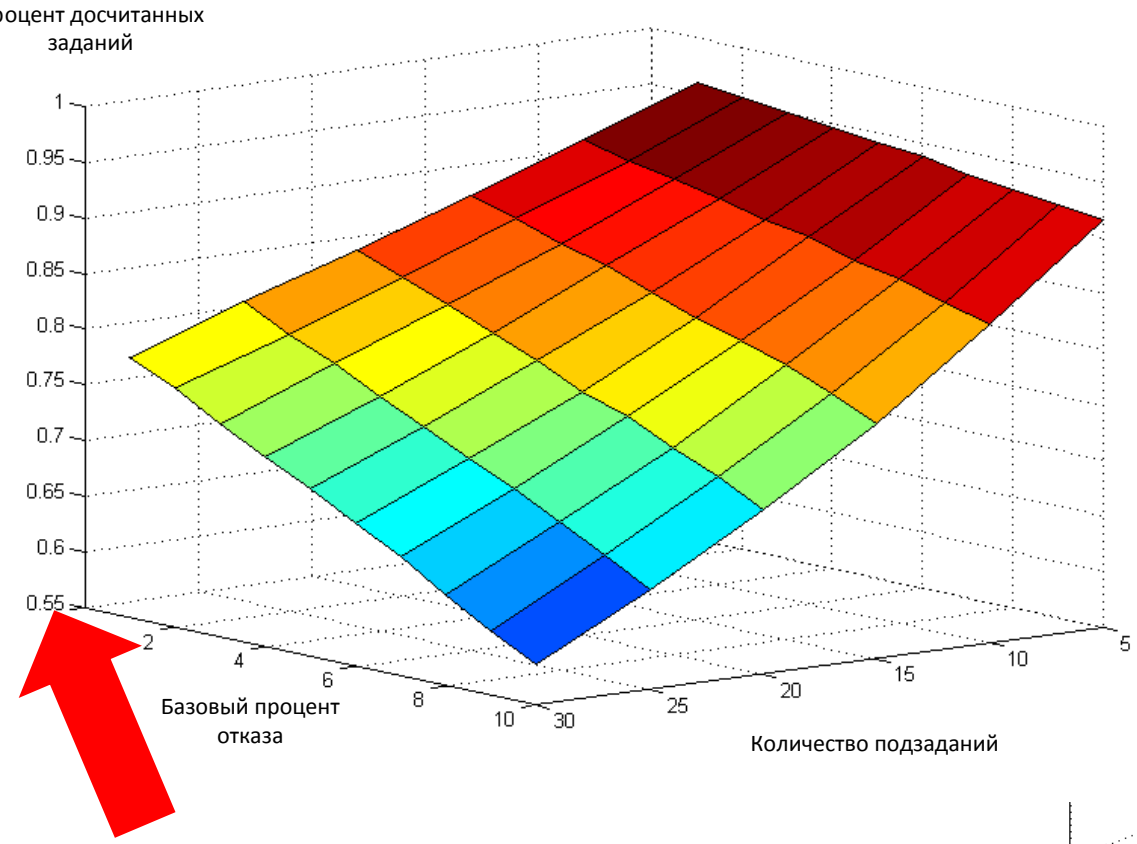


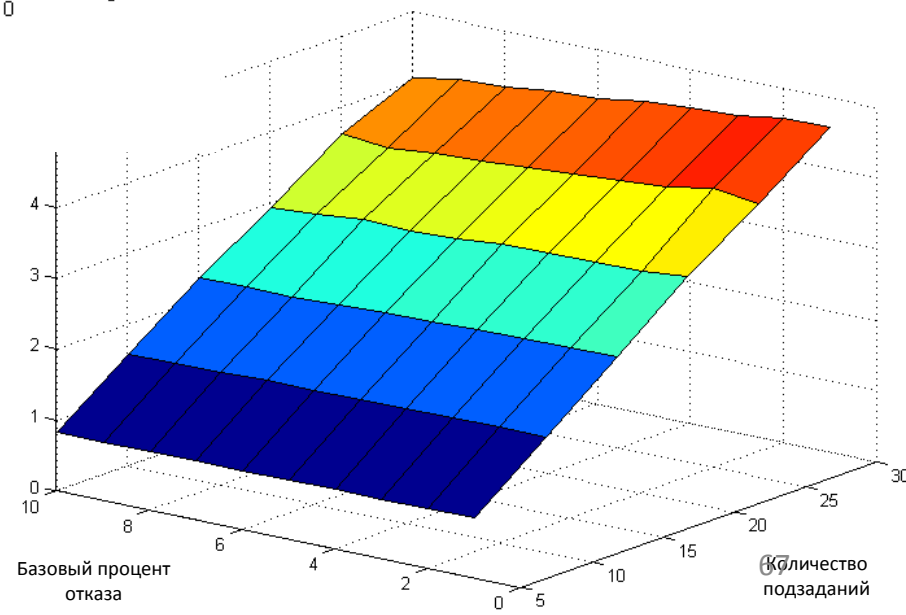
Определяется процент выполненных заданий

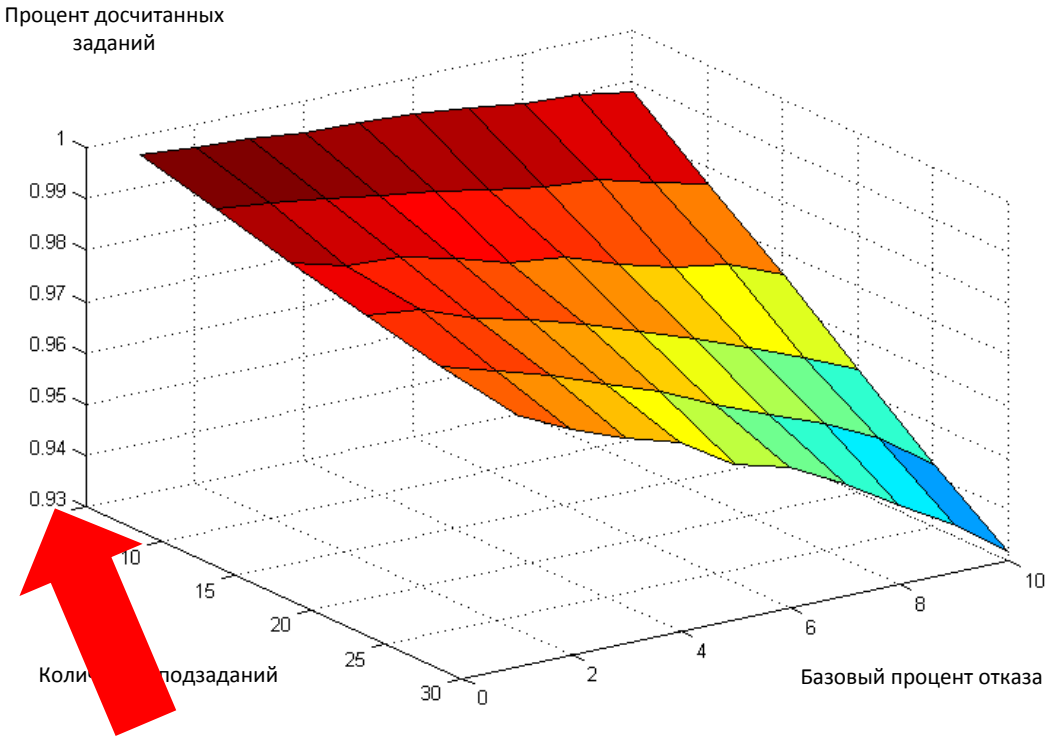
Параметры:

- Время вычисления задания
- Количество подзаданий
- Минимальный процент отказа

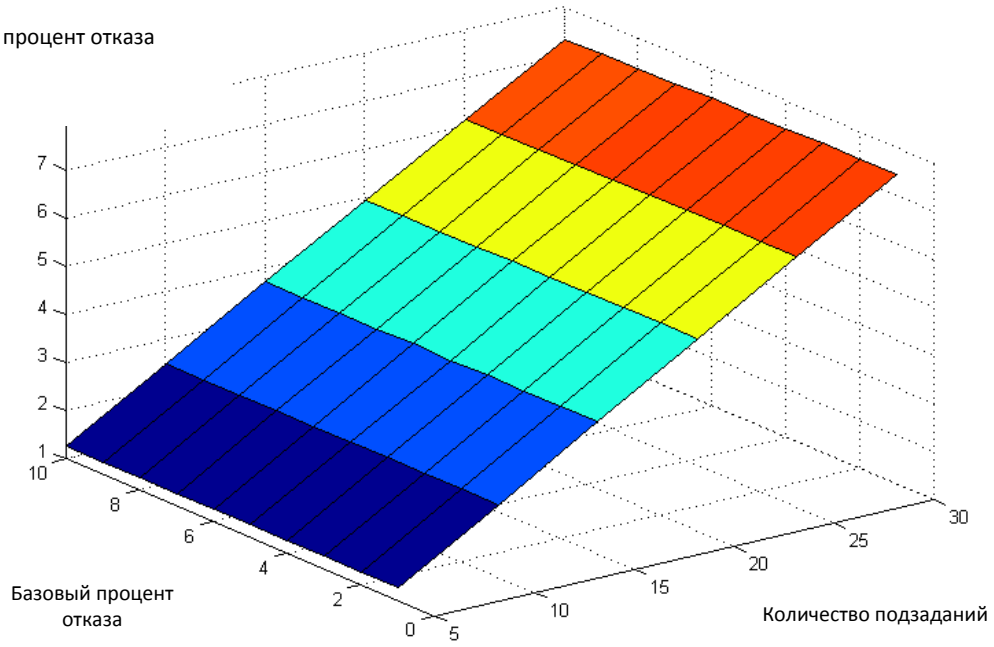
Репликация 2 копии только 56% выполненных заданий





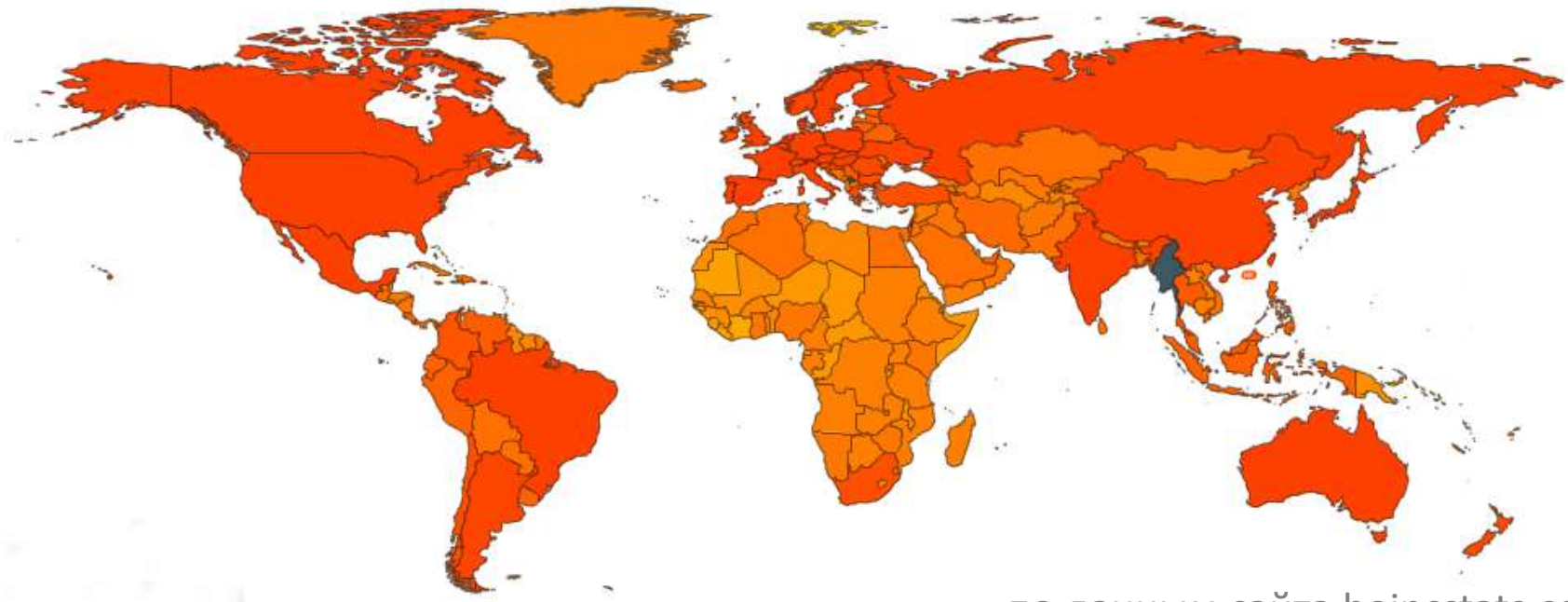


Репликация 3 копии
93%
выполненных заданий



Добровольные распределенные вычисления (ДРВ) на платформе BOINC

- 4.3 миллиона участников
- 16 миллионов компьютеров
- Около 100 международных проектов добровольных распределенных вычислений



по данным сайта boincstats.com

Кто такие добровольцы?

Доброволец (cruncher) предоставляет свои вычислительные ресурсы для расчета научных распределенных проектов

Почему?

- ✓Желание помочь науке
- ✓Причастность к научным открытиям (желание разобраться в получаемых результатах)
- ✓Спортивный интерес (кто больше наберет баллов)
- ✓Общение

Особенности организации экспериментов в проекте ДРВ

- Необходимость взаимодействия с добровольцами
- Общение на форуме
- Научно-популярное объяснение задач проекта
- Своевременное обновление информации на сайте проекта
- Обеспечение достаточного количества заданий для постоянной работы проекта
- Большой объем результатов и входных данных
- Необходимость обработки полной серии результатов
- Минимизация вероятности ошибочного расчета подзадачи
- Необходимость начальной репликации заданий в связи ненадежностью узлов

Взаимодействие с сообществом кранчеров (добровольцев)

- Социологические исследования
- Поддержка краудфандинговых проектов
- Размещение вычислительных мощностей
- Тестирование специализированных вычислителей
- Проведение круглых столов
- Совместные выступления на научных конференциях
- Совместная работа над задачами

Российские проекты ДРВ

- **SAT@home** (криптография, SAT-подход)
- **Optima@home** (решение задач конечномерной оптимизации) (зонтичный проект)
- **NetMax@home** (имитационное моделирование телекоммуникационных сетей)
- **USPEX@home** (поиск новых материалов)
- **Acoustics@home** (исследование дна Японского моря)
- **AndersonAttack@home** (криптография)
- Gerasim@home (комбинаторика, латинские квадраты)
- XANSONS for COD (материаловедение)
- ODLK@home (поиск канонических форм латинских квадратов)
- RakeSearch (поиск пар латинских квадратов)
- Amicable Numbers (поиск дружественных чисел)

Международная федерация грид-систем из персональных компьютеров

- Технологии грид-систем из персональных компьютеров (ГСПК) позволяют использовать простаивающие мощности персональных компьютеров, серверов для решения трудоемких вычислительных задач. ГСПК может использоваться в пределах одной организации или же объединять ресурсы добровольных участников из одного города, страны или даже всего мира.

- Развертывание гетерогенной инфраструктуры и разработка приложений для них являются непростыми задачами. Именно поэтому была создана Международная федерация грид-систем из персональных компьютеров.

- Международная федерация грид-систем из персональных компьютеров объединяет различные компании, университеты и исследовательские институты, а также людей заинтересованных в использовании простаивающей вычислительной мощности и желающих обменяться опытом друг с другом.

Образовательная деятельность

- Популяризация добровольных распределенных вычислений в СМИ
- Обучение студентов основам распределенных вычислений на платформе BOINC
- Работа студентов и аспирантов над действующими проектами

Студенты ведущих ВУЗов изучают распределенные вычислительные системы и добровольные распределенные вычисления:

- МГУ им.Ломоносова (г.Москва)
- МФТИ (г.Долгопрудный)
- МИСиС (г.Москва)
- ЮЗГУ (г.Курск)
- ИГУ (г.Иркутск)
- ПетрГУ (г.Петрозаводск)

Спасибо за внимание

Центр распределенных вычислений
Института проблем передачи информации РАН
(ИППИ РАН)

web: distributed-computing.ru

e-mail: kurochkin@iitp.ru