

# Основы параллельного программирования

Посыпкин Михаил Анатольевич

[mposypkin@gmail.com](mailto:mposypkin@gmail.com)

<http://parallelprog.blogspot.ru/>



# План лекции

- Понятие параллельного программирования
- Обзор основных типов параллельных и распределенных систем
- Производительность параллельных программ и основные препятствия к ее повышению



# Параллельное программирование

- Приложения требуют увеличения производительности компьютеров.
- Производительность процессора и памяти ограничена физическими характеристиками применяемых материалов.
- Многие задачи содержат независимые компоненты, которые могут решаться одновременно (т.е. параллельно).



# Параллельное программирование

Перечисленное приводит к естественному решению – **увеличивать число компонент оборудования**, участвующего в решении задач.

В частности, увеличивается число функциональных устройств одного процессора и общее число процессоров.

Параллельные вычисления – вычисления на системах, содержащих несколько параллельно работающих вычислителей.



# Спектр задач параллельного программирования

- Математическое моделирование:
  - Газовая и гидро-динамика.
  - Химическая физика.
  - Процессы в полупроводниках.
  - Имитационное моделирование в экономике.
  - Биология.
- Оптимизация:
  - Дискретное и линейное программирование.
  - Общая задача нахождения экстремума.
- Оптимальный поиск:
  - Дискретная оптимизация.
  - Распознавание образов.
  - Автоматическая верификация и доказательство теорем.





# Параллельные вычислительные системы

- **Системы с общей памятью** - ядра имеют доступ к общему адресному пространству
- **Системы с распределенной памятью** - каждое ядро обладает собственной памятью
- **GP GPU** (General Purpose Graphic Processing Units) – графические ускорители (карты), применяемые для задач общего назначения
- Гибридные системы



# Параллелизм внутри процессора (ядра)

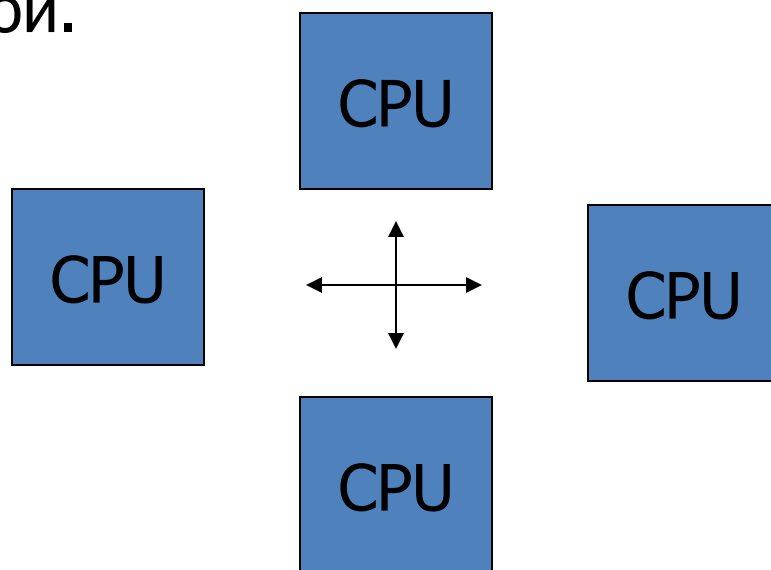
- Различные функциональные устройства.
- Конвейерная обработка.
- Векторные сопроцессоры.





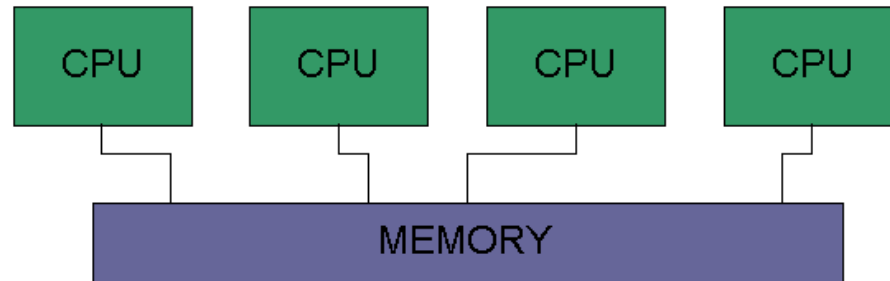
# Многопроцессорный параллелизм

В решении задачи принимает участие несколько (более одного) процессоров, взаимодействующих между собой.

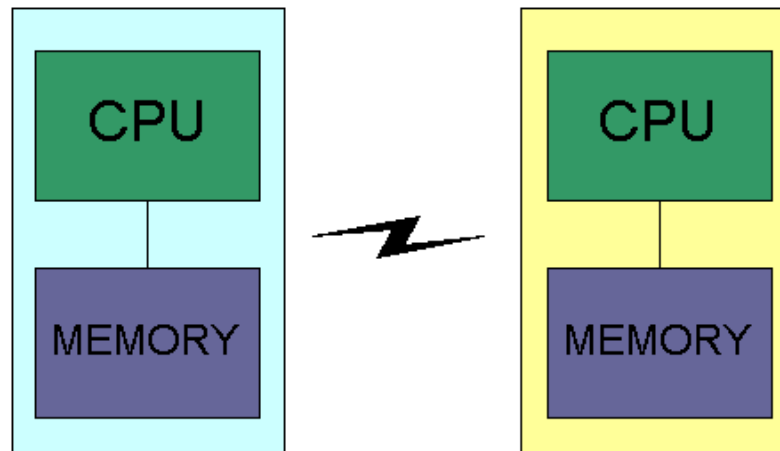


# Виды многопроцессорного параллелизма.

Общая память



Распределенная  
память



# Архитектура современных ЭВМ

Тип	Описание
Последовательные архитектуры	Один поток команд и данных. В настоящее время практически не встречается в «чистом» виде, но имеет важное значение как основная парадигма разработки программ.
Многоядерные процессоры	Несколько вычислительных ядер, реализованных на одном кристалле. Доминирующая архитектура в современных ПК и рабочих станциях.
Многопроцессорные системы с общей памятью	Несколько (многоядерных) процессоров, имеющих доступ к общему адресному пространству. Типична для мощных вычислительных серверов и рабочих станций.
Многопроцессорные системы с распределенной (гибридной) памятью	Совокупность вычислительных модулей, каждый из которых содержит собственный процессор(ы) и память. Между собой модули соединены высокопроизводительной сетью передачи данных. Классическая кластерная архитектура.



# Иерархия памяти

Регистры

Кэш 1-го уровня

Кэш 2(3)-го уровня

Оперативная память

Дисковая память



# Многоядерные процессоры Intel

**Multi-Core Leadership**

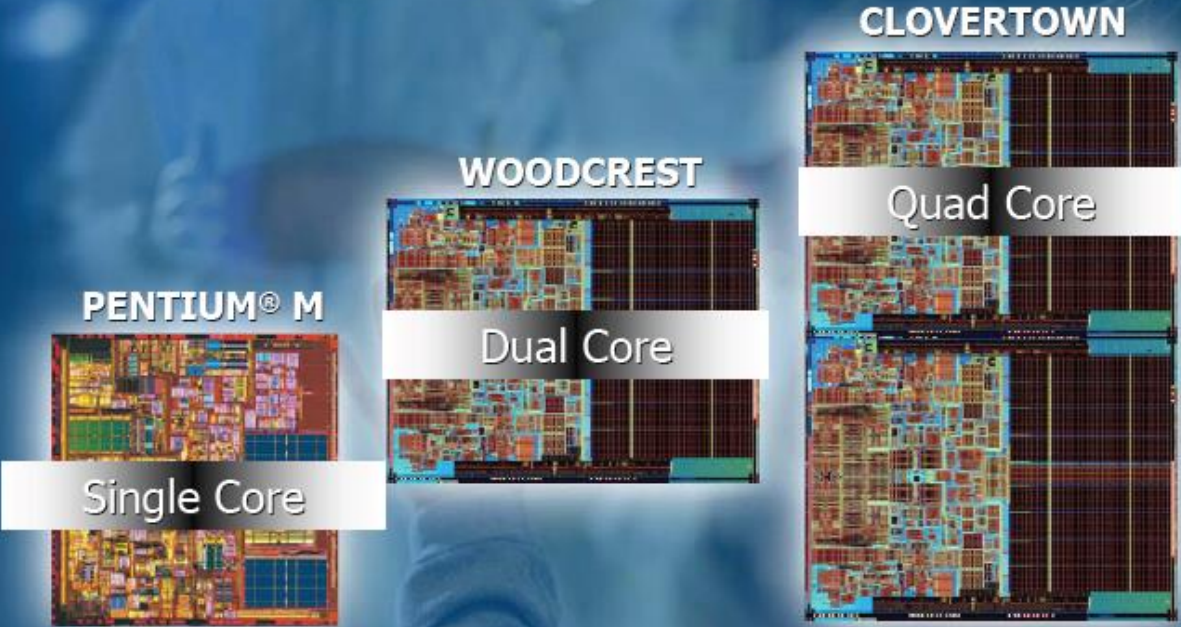
**PENTIUM® M**  
Single Core

**WOODCREST**  
Dual Core

**CLOVERTOWN**  
Quad Core

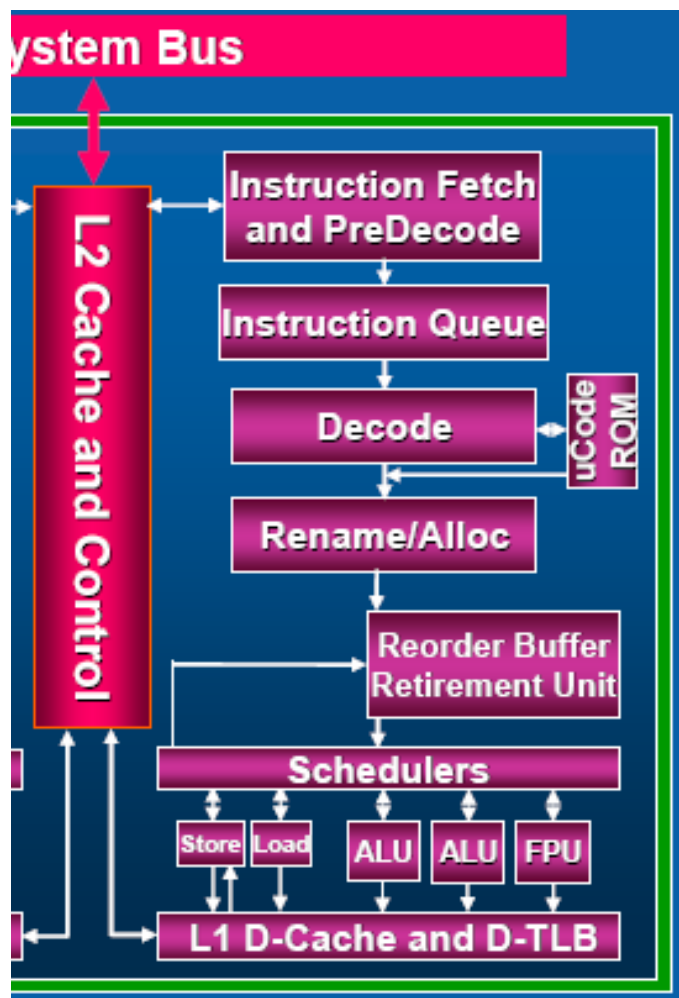
Intel Developer  
**FORUM**

intel



The diagram illustrates the evolution of Intel's multi-core processors. It features three processor models arranged in a staggered, ascending fashion from left to right. Each model is represented by a color-coded micrograph of the chip. The first model on the left is the Pentium M, labeled 'Single Core'. The middle model is the Woodcrest, labeled 'Dual Core'. The largest model on the right is the Clovertown, labeled 'Quad Core'. The background is a blue gradient with a faint image of a person's face. In the bottom left corner is the 'Intel Developer FORUM' logo, and in the bottom right corner is the 'intel' logo. A small speaker icon is located in the bottom right corner of the slide.

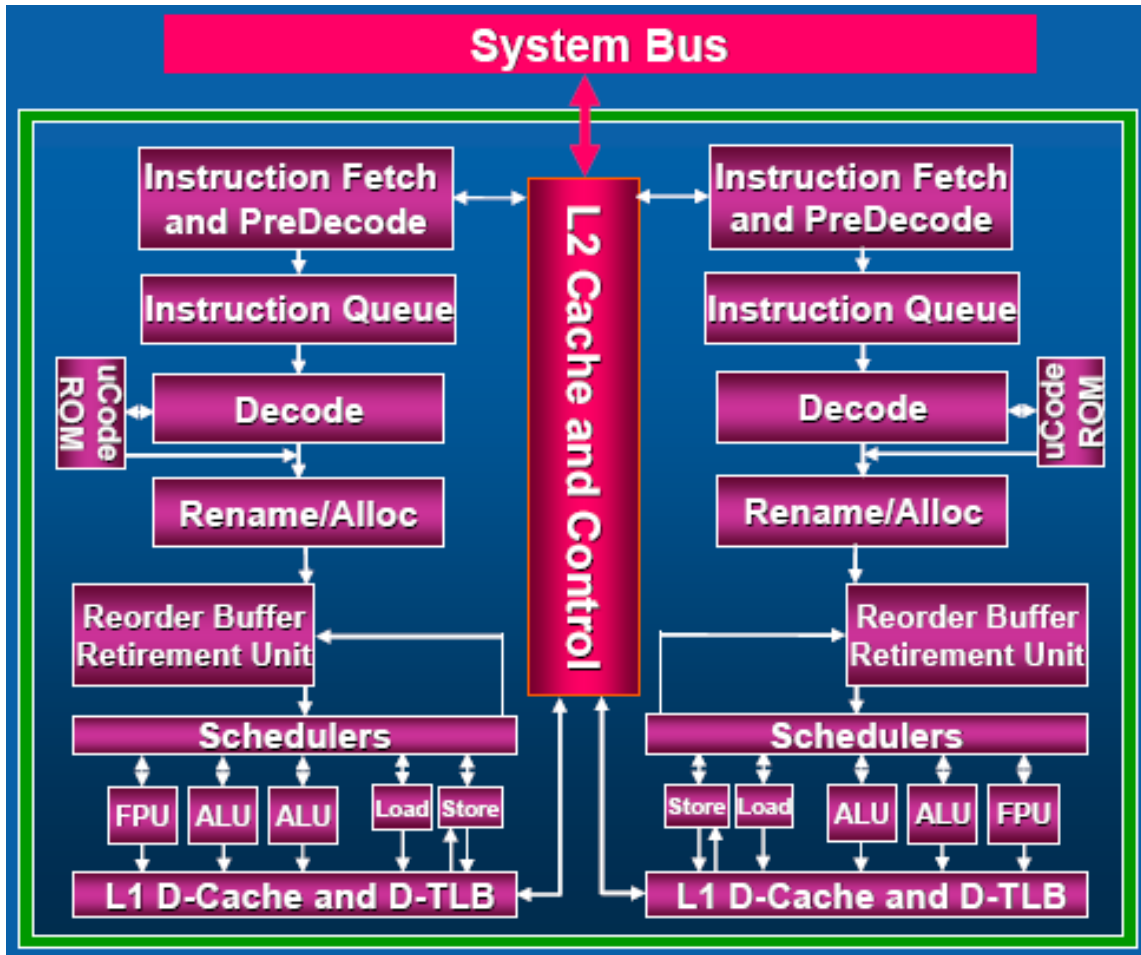
# Одно ядро (Pentium-M)



Есть возможность динамического переупорядочивания инструкций (**REORDER BUFFER**) с целью максимальной загрузки функциональных устройств



# Процессор Woodcrest



Общий L2-кэш,  
TLB и L1-кэш  
индивидуальн  
ый для  
каждого CPU



# Intel Clovertown (серия Xeon 5300)



Два  
двухядерных  
модуля  
Woodcrest

L1 и L2  
кэши разные





# Технология Hyper Threading

Общая идея гипертрейдинга состоит в том, чтобы за счет небольшого увеличения сложности и размера процессора обеспечить возможность выполнения двух потоков на ресурсах одного ядра

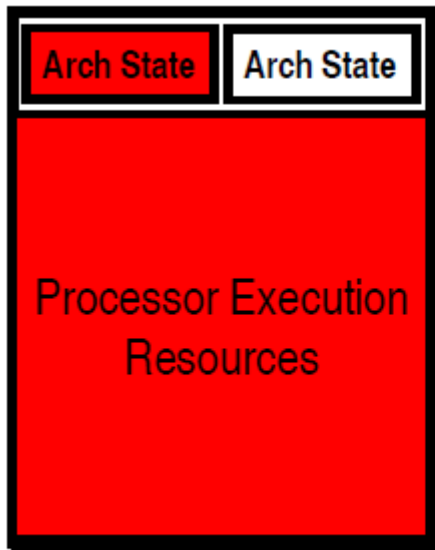


# Технология Hyper Threading

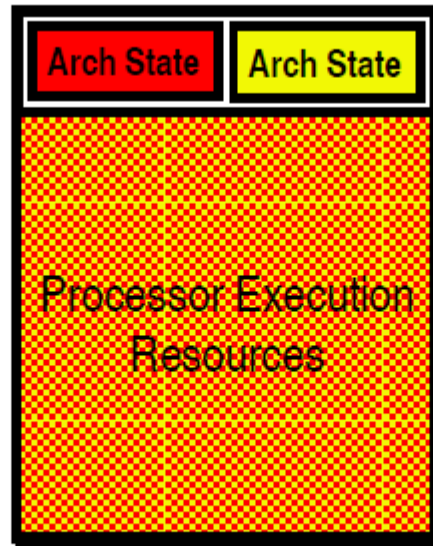
- Каждый физический процессор хранит состояние сразу двух потоков
  - Регистры
  - Контроллер прерываний APIC
  - Некоторые специальные таблицы (ITLB)
- Используются паузы из-за зависимостей по данным и обращений к памяти за счет общего планирования



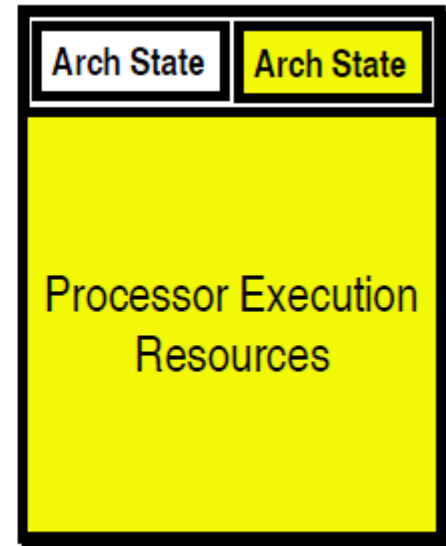
# Технология Hyper Threading



(a) ST0-Mode



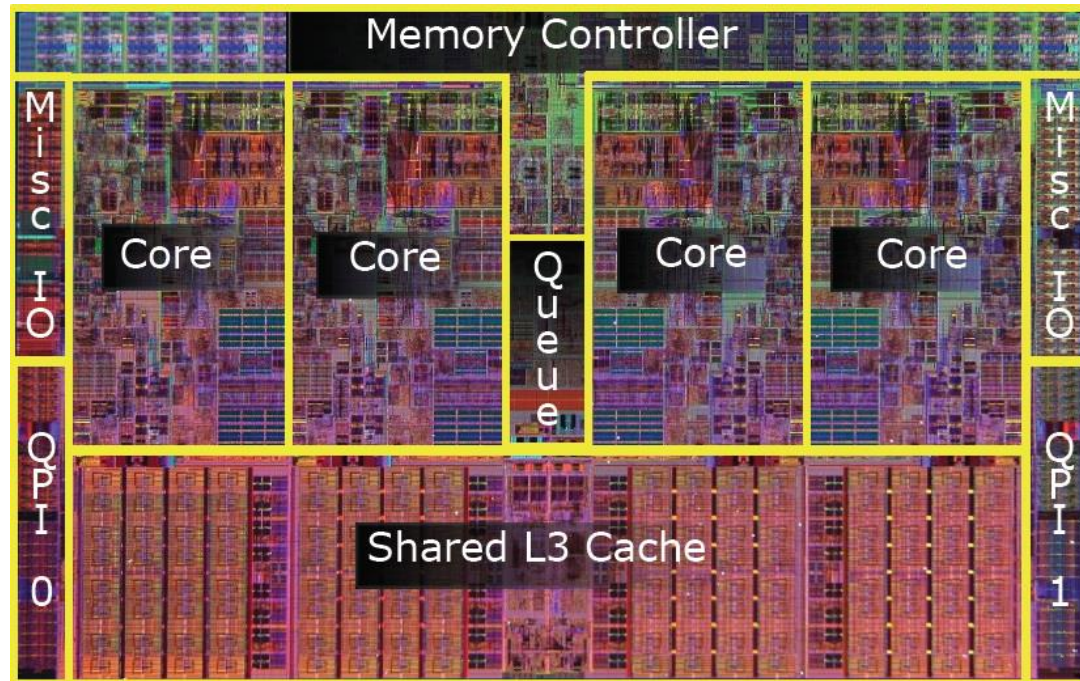
(b) MT-Mode



(c) ST1- Mode



# Intel Core i7 (Nehalem)



Каждое ядро имеет поддержку HT получается до 12 (в зависимости от модели CPU) виртуальных ядер



# Intel Xeon Phi

- Intel MIC (англ. Intel Many Integrated Core Architecture) — архитектура многоядерной процессорной системы
- Прототип процессоров архитектуры MIC (кодовое название Knights Ferry) был выпущен в 2010 году.
- В июне 2012 года Intel объявила о ребрендинге процессоров под названием **Xeon Phi**



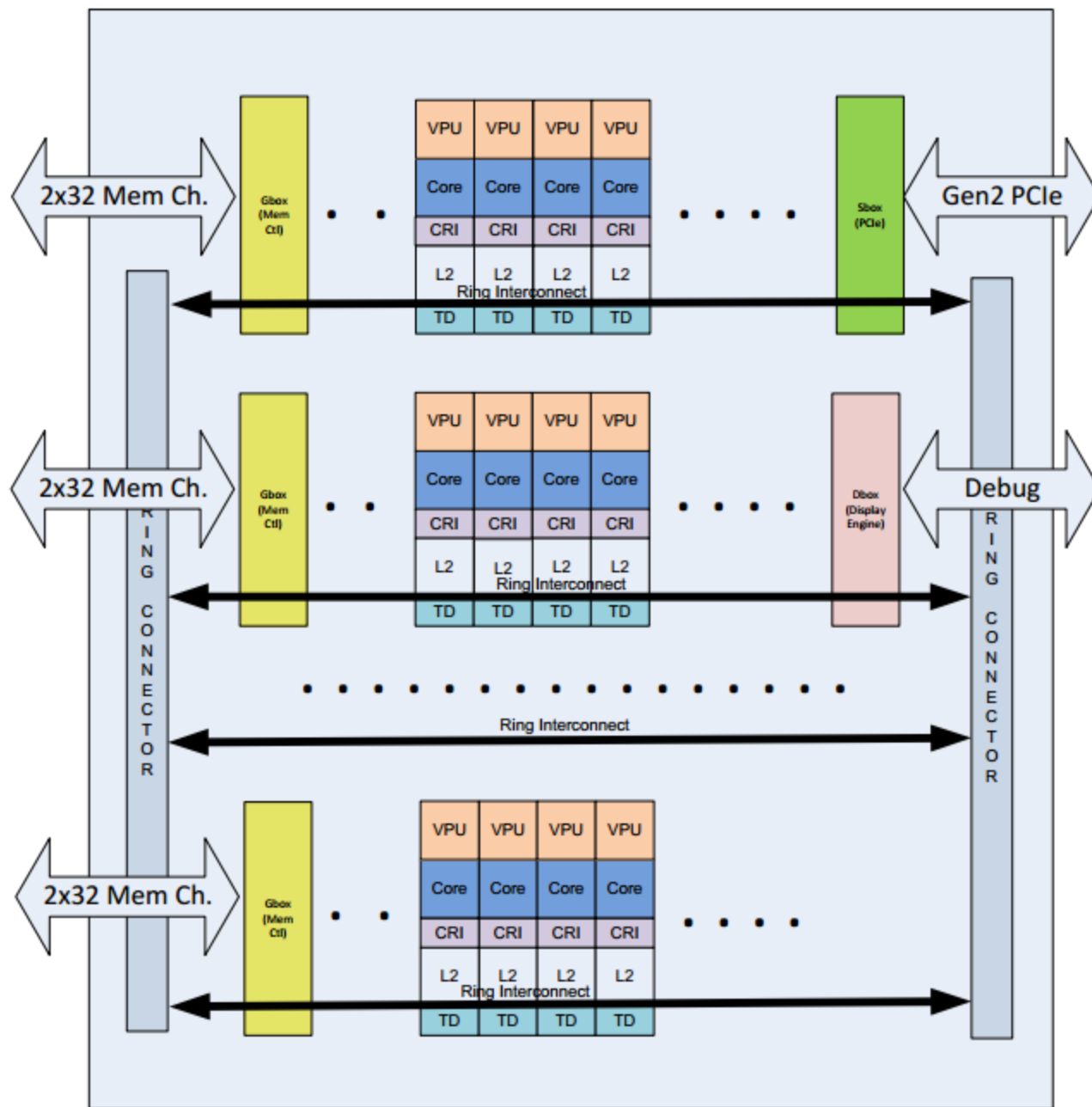


Figure 2-1: Basic building blocks of the Intel® Xeon Phi™ Coprocessor

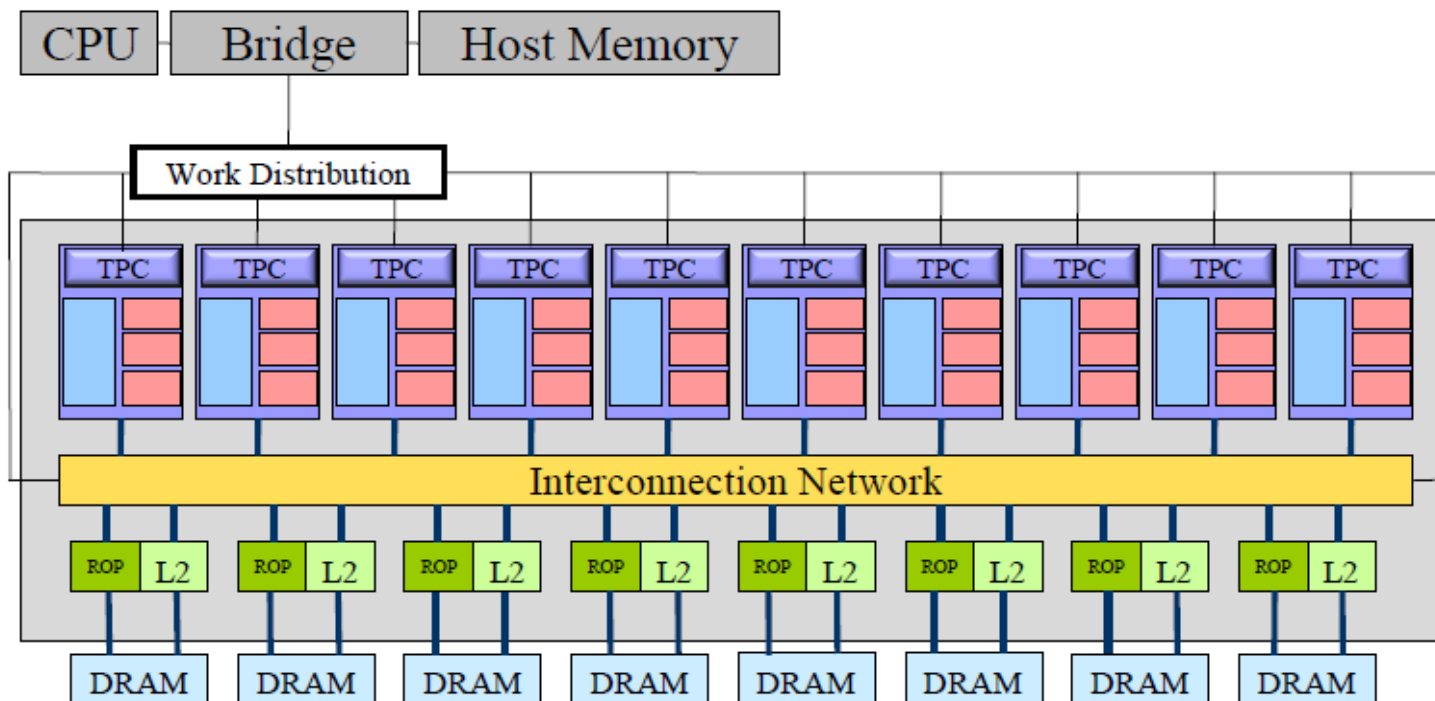


# Особенности Xeon Phi

- Векторное расширение (VPU) для работы с 512-битными векторами (по одному на ядро)
- До 61 вычислительного ядра
- Отдельные кэши 1-го и 2-го уровня на каждом ядре

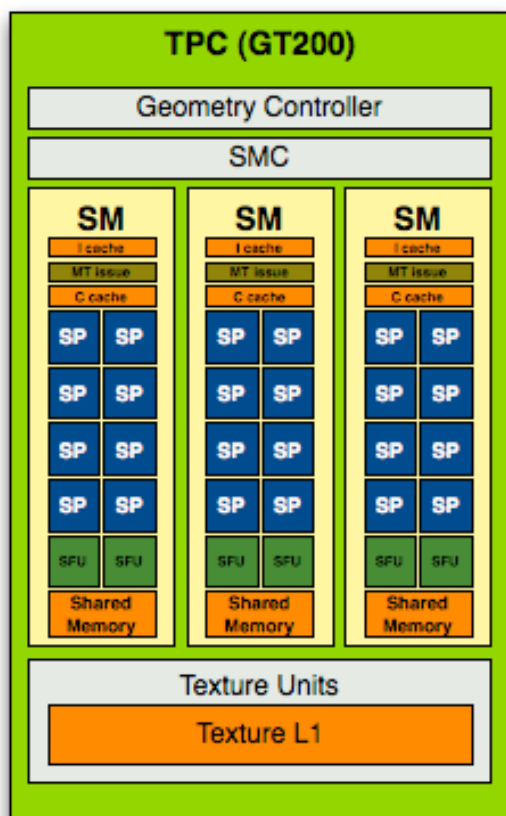
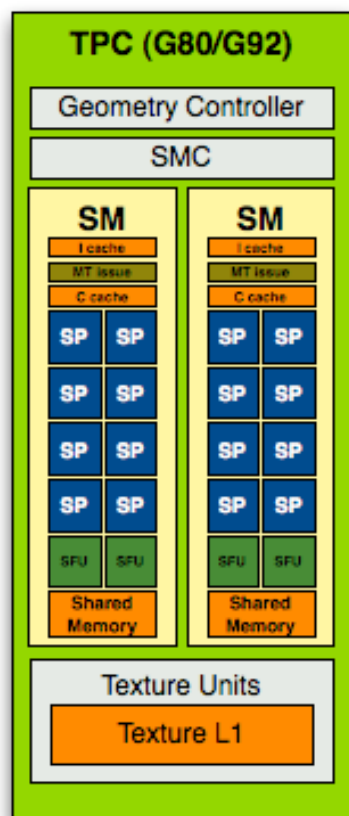


# Архитектура GP GPU





# Архитектура TPC (Texture Processing Cluster)

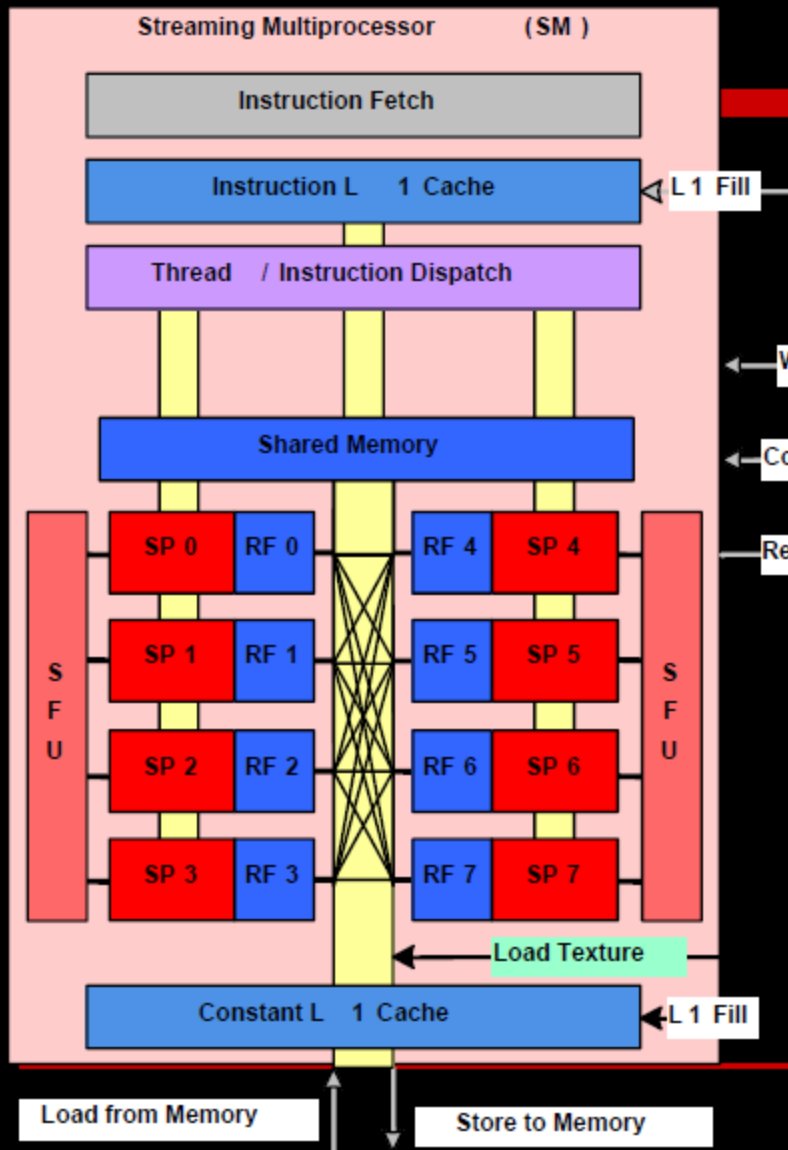


SM – Streaming Multiprocessor

SP – Streaming processor



# Streaming Multiprocessor (SM)



- 8 Streaming Processors (SP)

- 2 **Super Function Units** (SFU) – сложные функции (sin, cos, etc.)

- Много-поточная доставка инструкций

- 1 - 512 потоков активны в кажд. момент

- SIMD инструкции для воронов 16/32 тредов !!!! **ВЕТВЛЕНИЯ**

- Hot clock = 1.35 GHz

- 20+ GFLOPS для каждого SP

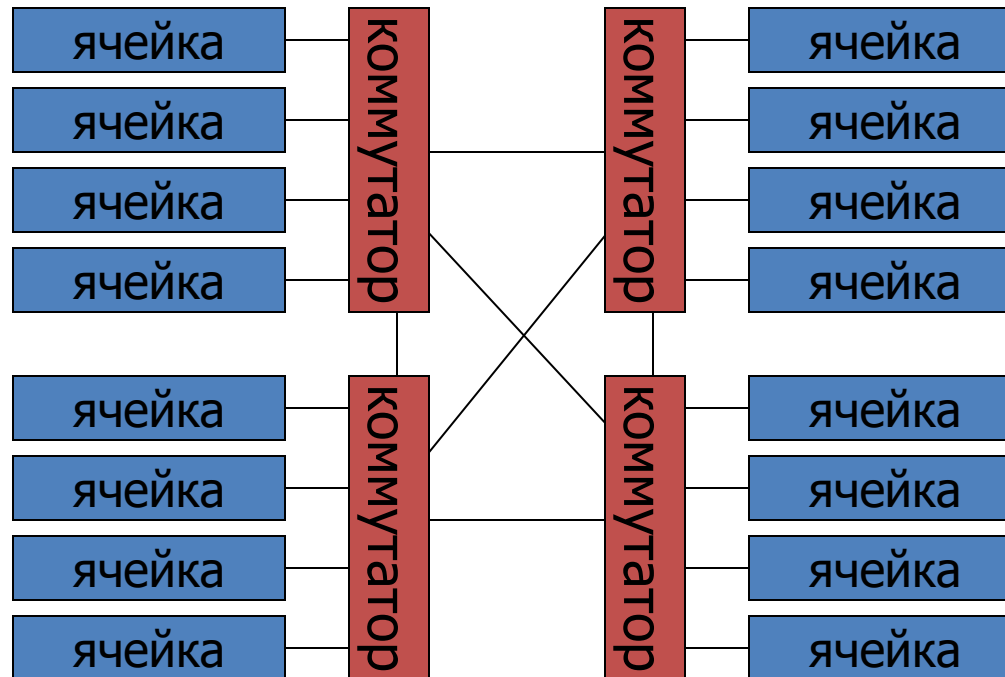
- Локальный регистровый файл (RFn)

- 16 KB разделяемой памяти

# Архитектура HP-Superdome

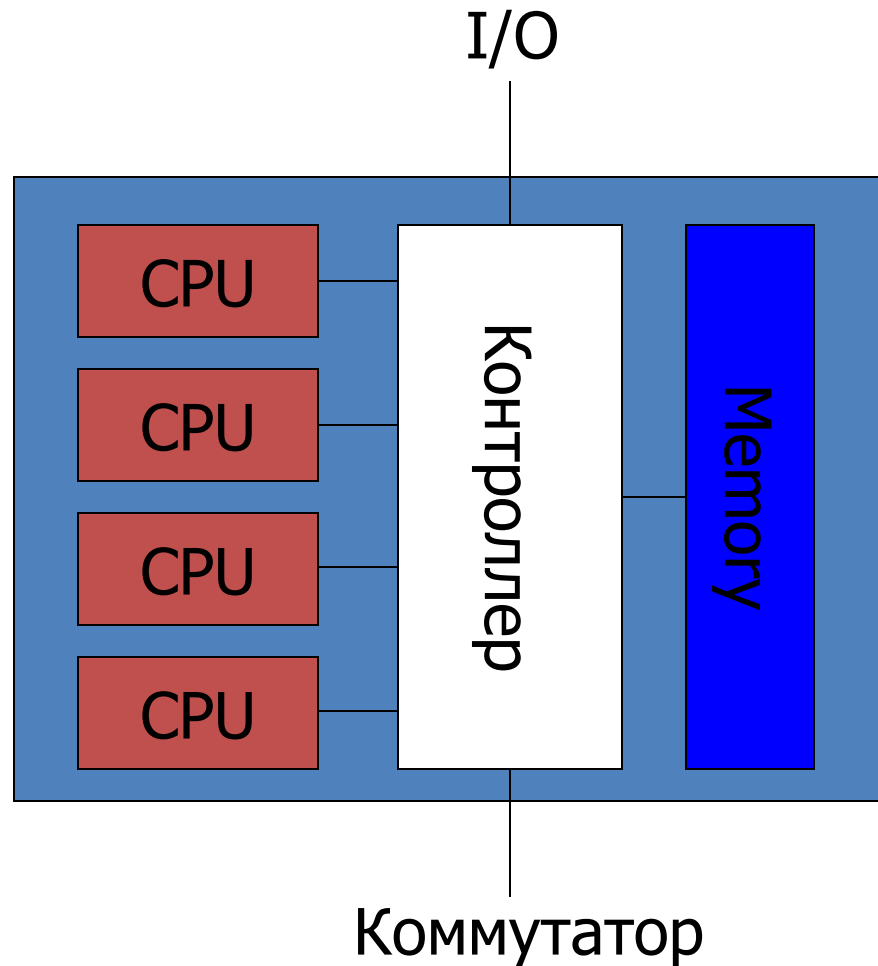


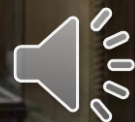
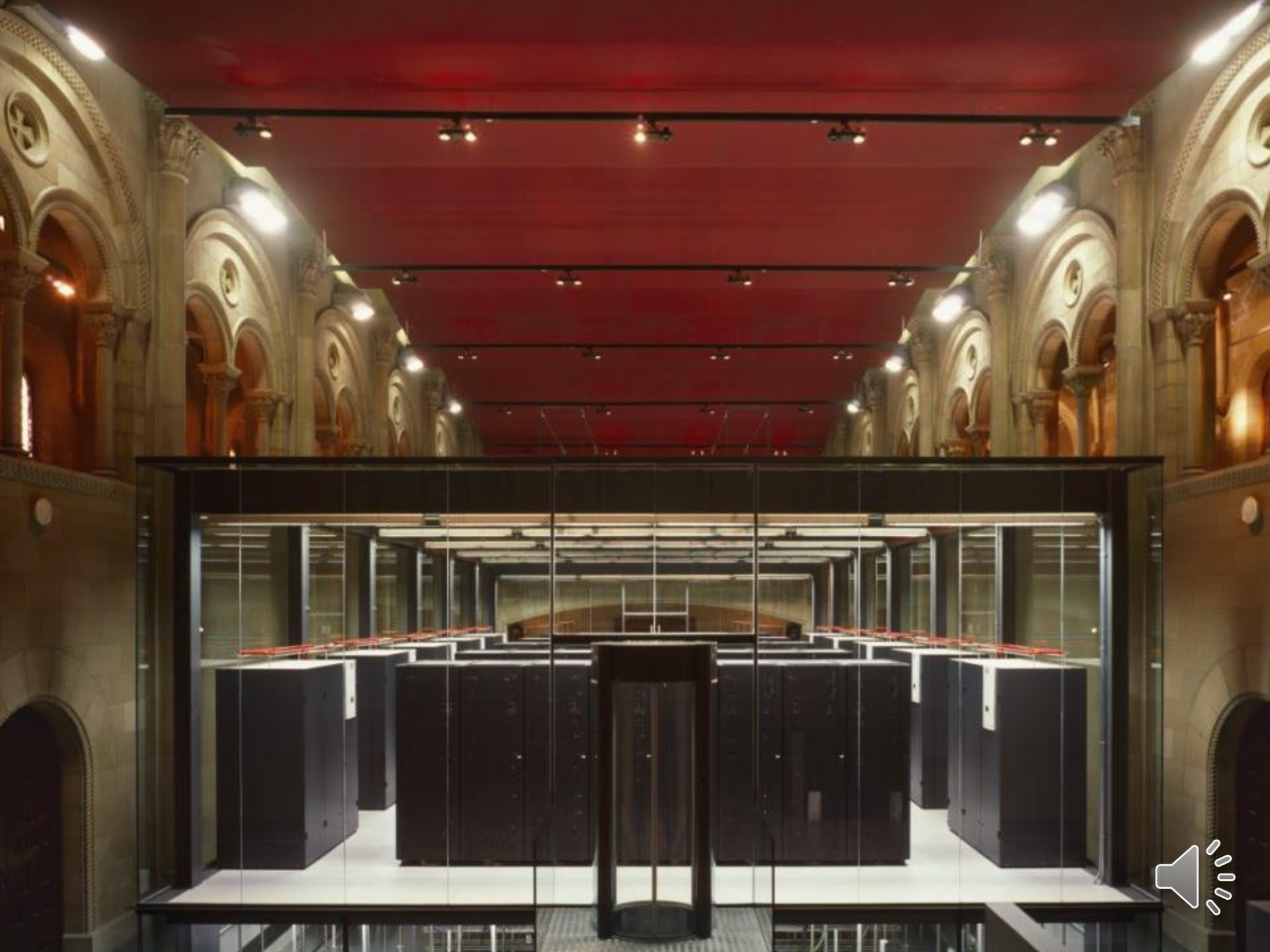
# Архитектура HP-Superdome: общая орагнизация: ccNuma



# Архитектура HP-Superdome:

## Ячейка – SMP система





# Характеристики MareNostrum

- Пиковая производительность: 94,21 TFlops
- 10240 PowerPC processors
- Оперативная память: 20 Tb
- Дисковая память: 480 Tb
- Коммуникации
  - Myrinet (вычисления)
  - Gigabit Ethernet (загрузка, управление)





# Архитектура

Blade Center



Blade Server JS320



29 x IBM Rack



# Суперкомпьютер «Ломоносов»



# Гибридная архитектура

В качестве основных узлов используются решения TB2-XN на базе четырехъядерных и шестиядерных процессоров Intel Xeon X5570 Nehalem и X5670 Westmere.

Суперкомпьютерный комплекс также содержит **гибридные узлы** TB2-TL на базе процессоров Intel Xeon и NVIDIA Tesla.



Пиковая производительность	1.7 Пфлопс
Число вычислительных узлов x86/GPU	5 104 / 1 065
Число процессоров x86	12 346
Число процессорных ядер x86/GPU	52 168 / 954 840
Число типов вычислительных узлов	8
Основной тип вычислительных узлов	TB2-XN
Процессор основного типа вычислительных узлов	Intel® Xeon X5570 / X5670



Оперативная память	83 ТБ
Занимаемая площадь (вычислитель)	252 м <sup>2</sup>
Энергопотребление вычислителя	2,6 МВт
Интерконнект	QDR InfiniBand
Система хранения данных	Трехуровневая с параллельной файловой системой хранения данных
Операционная система	ClustrX T-Platforms Edition



# Тест LINPACK (HPL)

Тест состоит в решении системы линейных уравнений с помощью LU-факторизации. Основное время затрачивается на векторные операции типа умножение и сложение. Производительность определяется как количество "полезных" вычислительных операций над числами с плавающей точкой в расчете на 1 секунду, и выражается в Мфлоп/сек (миллионах операций в секунду). Результаты теста используются при составлении рейтинга Top500.



# Тор 500 (43-я редация, июнь 2014)

Rank	Site	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	National Super Computer Center in Guangzhou China	Tianhe-2 (MilkyWay-2) - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P NUDT	3120000	33862.7	54902.4	17808
2	DOE/SC/Oak Ridge National Laboratory United States	Titan - Cray XK7 , Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x Cray Inc.	560640	17590.0	27112.5	8209
3	DOE/NNSA/LLNL United States	Sequoia - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom IBM	1572864	17173.2	20132.7	7890
4	RIKEN Advanced Institute for Computational Science (AICS) Japan	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect Fujitsu	705024	10510.0	11280.4	12660
5	DOE/SC/Argonne National Laboratory United States	Mira - BlueGene/Q, Power BQC 16C 1.60GHz, Custom IBM	786432	8586.6	10066.3	3945



# Грид-системы

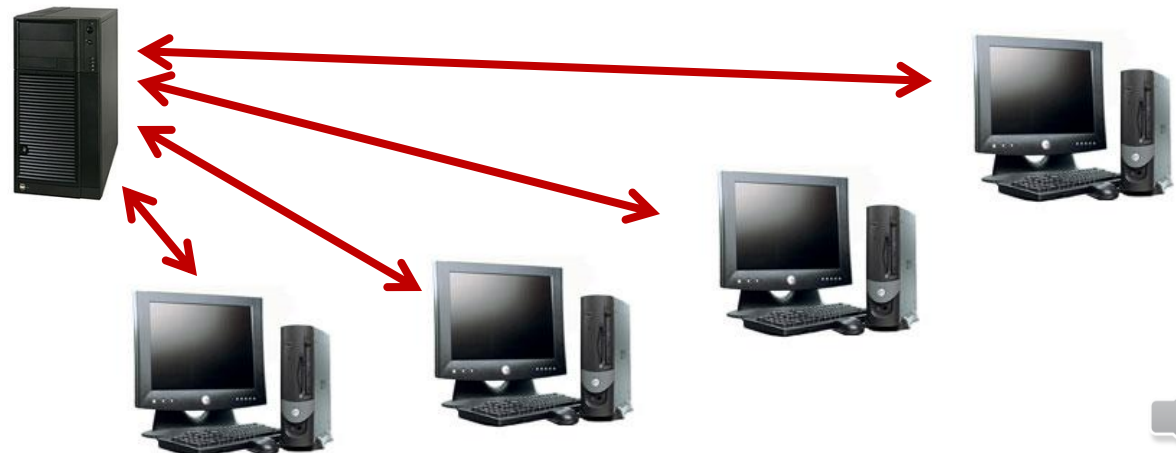
**Грид** (Grid) - согласованная, открытая и стандартизованная среда, которая обеспечивает гибкое, безопасное, скоординированное разделение ресурсов в рамках виртуальной организации. (Я. Фостер, К. Кессельман)

- Гриды рабочих станций – объединяют ресурсы простаивающих рабочих станций, домашних компьютеров, серверов (SETI@home ).
- Сервисные Гриды – совокупность вычислительных ресурсов, доступных в рамках единой политики доступа (EGEE, DEISA).



# Гриды рабочих станций

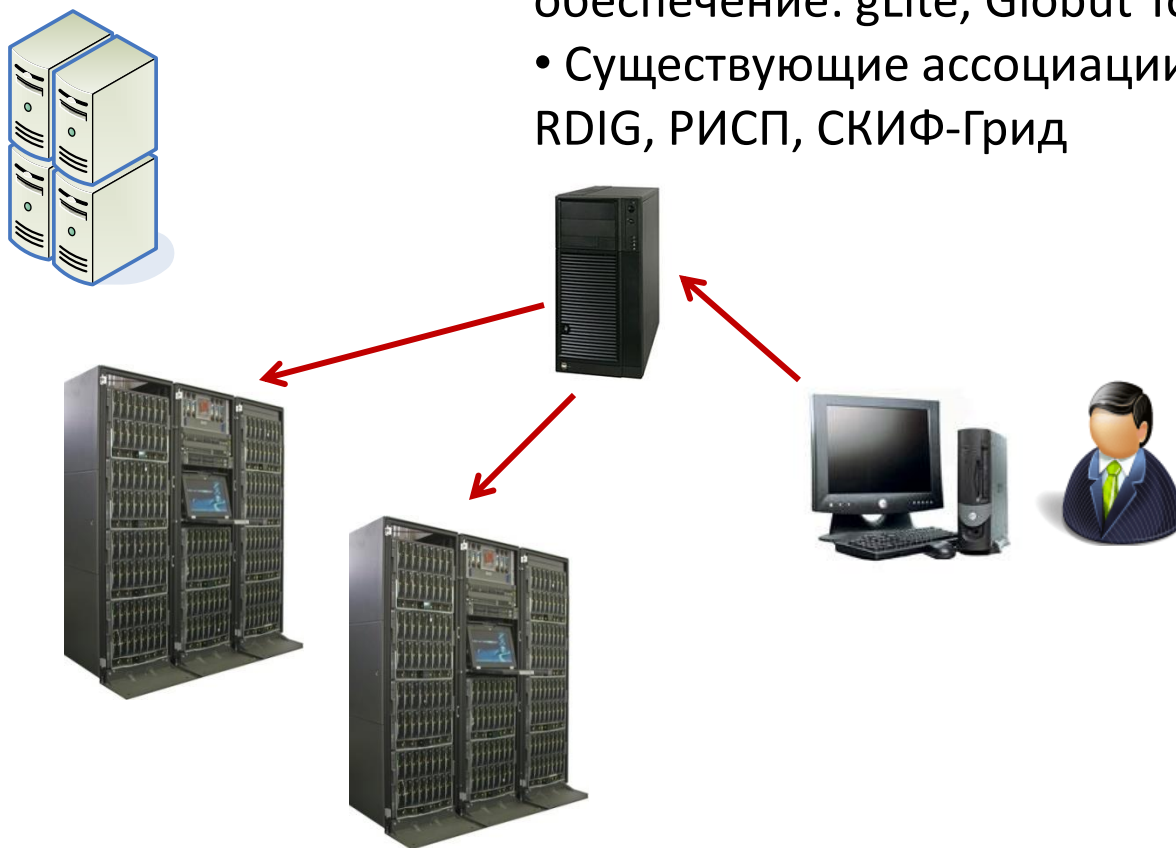
- Используют вычислительные ресурсы простаивающих рабочих станций предприятий и домашних ПК для проведения вычислений;
- Системы для проведения вычислений в рамках Гридов рабочих станций – BOINC, XWHEP, Condor;
- Проекты: [MilkyWay@home](#) (построение трехмерной модели млечного пути), [SETI@home](#) (обработка сигналов с целью обнаружения внеземного разума).





# Сервисные Грид-системы

- Применяется организованный доступ к вычислительным ресурсам
- Промежуточное программное обеспечение: gLite, Globus Toolkit, Unicore
- Существующие ассоциации: EGEE, DEISA, RDIG, РИСП, СКИФ-Грид



# Облачные вычисления

Википедия: «**Облачные**» **вычисления** (cloud computing) — это модель обеспечения повсеместного и удобного сетевого доступа по требованию к общему пулу конфигурируемых вычислительных ресурсов (сетям передачи данных, серверам, устройствам хранения данных, приложениям и сервисам), которые могут быть оперативно предоставлены и освобождены с минимальными эксплуатационными затратами и/или обращениями к провайдеру.



# Требования к облакам

- 1. Самообслуживание по требованию:** потребитель самостоятельно определяет и изменяет вычислительные потребности, такие как серверное время, скорости доступа и обработки данных, объём хранимых данных без взаимодействия с представителем поставщика услуг;
- 2. Универсальный доступ по сети** — услуги доступны потребителям по сети передачи данных вне зависимости от используемого терминального устройства;
- 3. Объединение ресурсов** — поставщик услуг объединяет ресурсы для обслуживания большого числа потребителей в единый пул для динамического перераспределения мощностей между потребителями в условиях постоянного изменения спроса на мощности;



# Требования к облакам

**4. Эластичность** — услуги могут быть предоставлены, расширены, сужены в любой момент времени, без дополнительных издержек на взаимодействие с поставщиком, как правило, в автоматическом режиме;

**5. Учёт потребления**, поставщик услуг автоматически исчисляет потреблённые ресурсы и на основе этих данных оценивает объём предоставленных потребителям услуг.



# Типы облачных сервисов

1. **Software as a service (SaaS)** — приложения, которые поставляется конечному пользователю в «облачной» инфраструктуре как службы через Internet. Потребителю предоставляется возможность использования прикладного ПО провайдера
2. **Platform as a service (PaaS)** — платформа разработки и развертывания приложений поставляется в виде службы для разработчиков, позволяющей быстро создавать и развертывать приложения SaaS.
3. **Infrastructure as a service (IaaS)** — оборудование, такое как вычислительные серверы, системы хранения и сетевые элементы, предоставляются в виде служб.



# Примеры облачных сервисов

SaaS	Google Docs
PaaS	Google App Engine
IaaS	Amazon Cloud



# ОСНОВНЫЕ ХАРАКТЕРИСТИКИ ПРОИЗВОДИТЕЛЬНОСТИ ПАРАЛЛЕЛЬНЫХ ПРОГРАММ



# Ускорение (наблюдаемое)

$$S = \frac{T_s}{T_p}$$

$T_p$  - время параллельных вычислений

$T_s$  - время последовательны вычислений





# Производительность пиковая и реальная

**Пиковая производительность** – максимальное количество операций, которые вычислительное устройство может выполнить за единицу времени.

**Реальная производительность** – количество операций, которое вычислительное устройство реально выполняет.

$$p = \frac{W}{T}$$

**Загруженность** = (реальная производительность)/(пиковая производительность)

$$l = \frac{p}{\pi}$$



# Линейное и «сверхлинейное» ускорение

Линейное ускорение:  $S = n$ .

Эффект «сверхлинейного» ускорения:  
наблюдаемое ускорение больше числа  
процессоров:  $S > n$ .

Причина – не учитывается загруженность  
процессоров, либо изменение количества  
операций.



# Закон Амдала

$\beta$  -доля последовательных вычислений

$W$  -общий объем работы

$$S \leq \frac{W}{\beta \cdot W + (1 - \beta)W/n} = \frac{n}{\beta \cdot n + (1 - \beta)}$$

$$S \leq \frac{n}{\beta \cdot n + (1 - \beta)} \leq \frac{1}{\beta}$$



# Эффективность

Эффективность – отношение ускорения к числу процессоров. Показывает насколько эффективно используются аппаратные ресурсы.

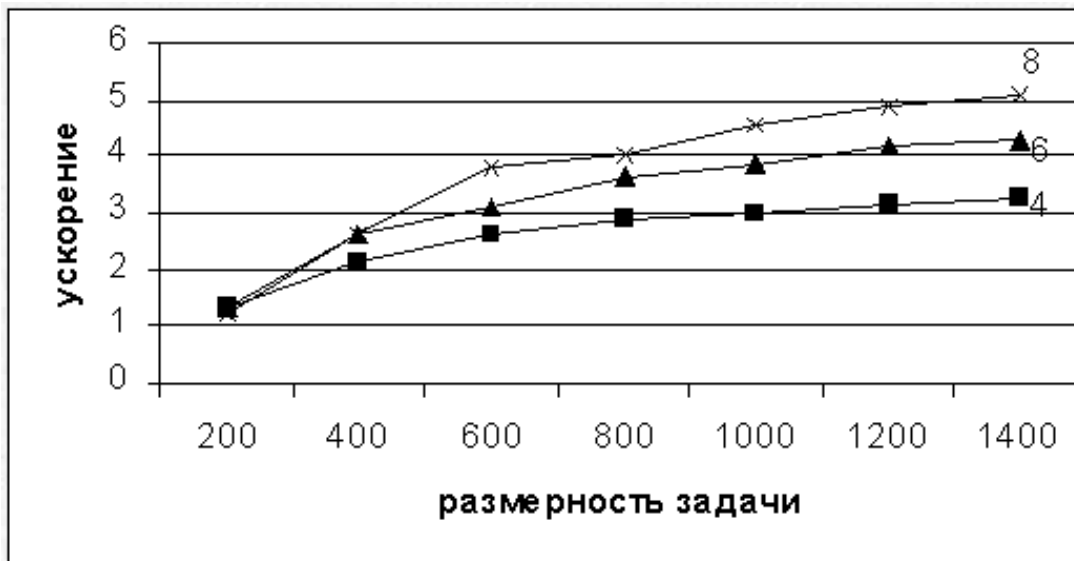
$$E = \frac{S}{n} \leq 1$$



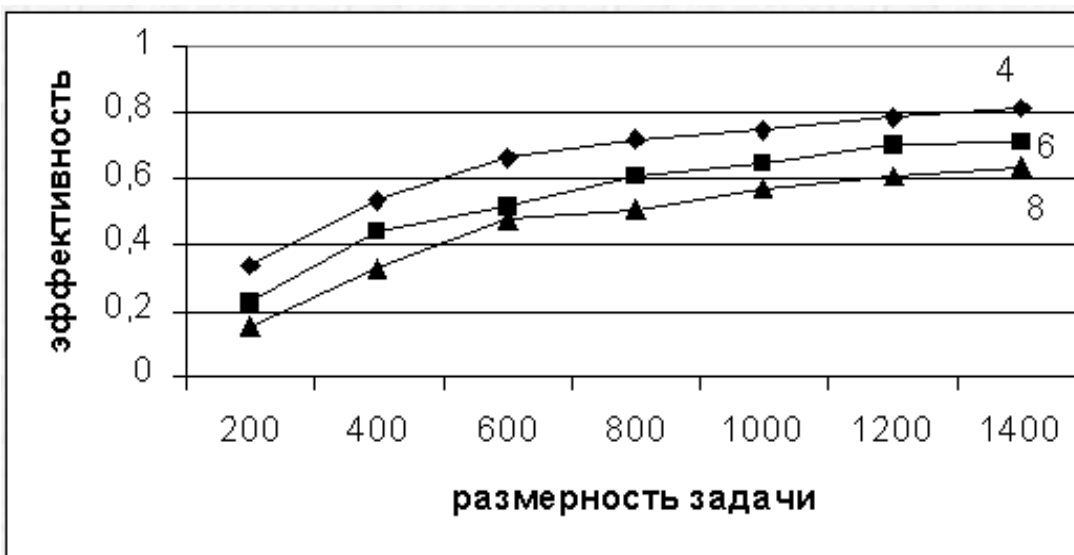
# Масштабируемость

*Вики:* **Масштаби́руемость** ([англ. scalability](#)) — в [электронике](#) и [информатике](#) означает способность системы, сети или процесса справляться с увеличением рабочей нагрузки (увеличивать свою производительность) при добавлении ресурсов (обычно аппаратных). Масштабируемость — важный аспект [электронных систем](#), [программных комплексов](#), [систем баз данных](#), [маршрутизаторов](#), [сетей](#) и т. п., если для них требуется возможность работать под большой нагрузкой. Система называется *масштабируемой*, если она способна увеличивать производительность пропорционально дополнительным ресурсам.





Тест LINPACK  
(LU-разложение):  
кластер из 8  
компьютеров



Эффективность и  
ускорение при  
разном  
количестве  
процессоров.

(информация с сайта  
Кемеровского ГУ)

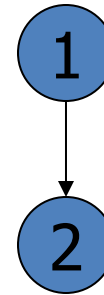


# Информационные зависимости

Зависимость по данным:

1:  $a = 1;$

2:  $b = a;$



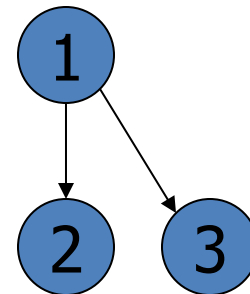
Зависимость по управлению:

1:  $\text{if}(a) \{$

2:  $x = c + d;$

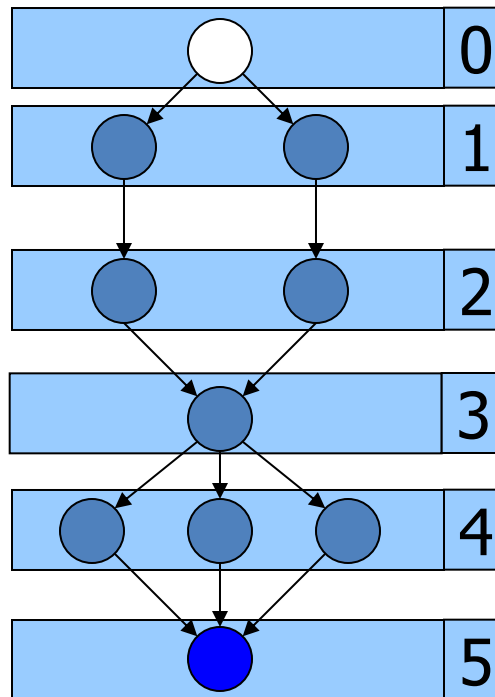
3:  $y = 1;$

4:  $\}$



# Граф зависимостей

Операции, соединенные путем из дуг, не могут выполняться одновременно.

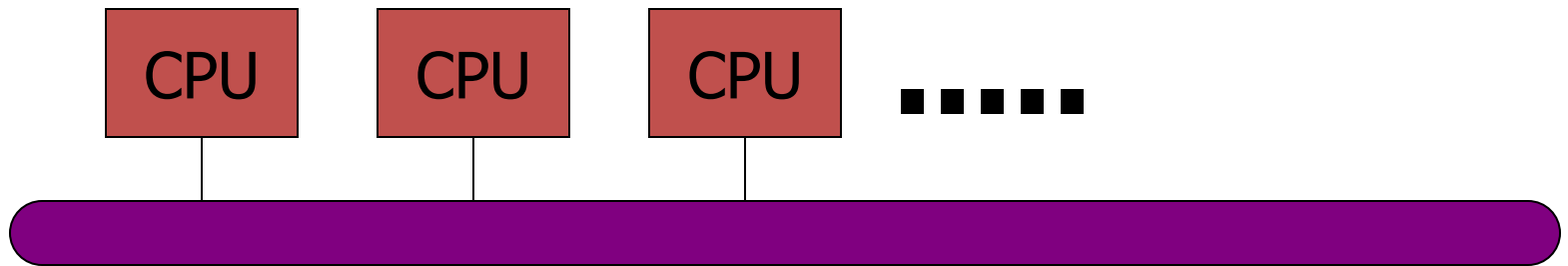


Другие операции могут выполняться одновременно при наличии требуемых функциональных устройств.





# Концепция неограниченного параллелизма

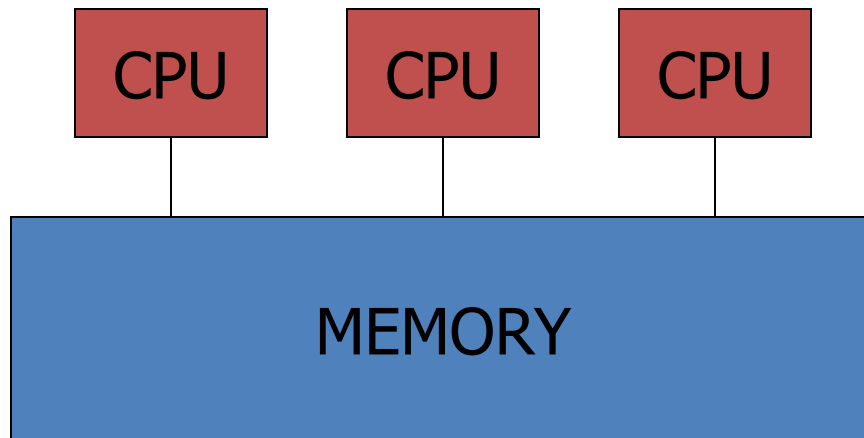


Количество процессоров неограниченно.

Концепция может применяться для исследования максимально возможного ускорения.



# Упрощенная модель параллельной машины с общей памятью



Процессоры работают синхронно по шагам: на каждом шаге выполняется операция (выборка операндов + арифметическая операция + запись в память). Шаг занимает 1 такт.



# Лемма Брента

Пусть  $q$  – число операций алгоритма, выполнение каждой операции занимает в точности одну единицу времени (такт),  $t$  – время выполнения на системе с достаточным числом одинаковых процессоров, то на системе, содержащей  $n$  процессоров, алгоритм может быть выполнен за время, не превосходящее  $t + (q - t)/n$ .



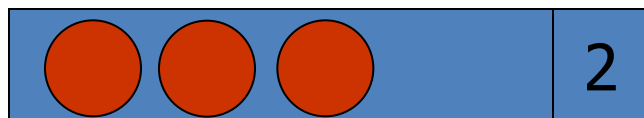
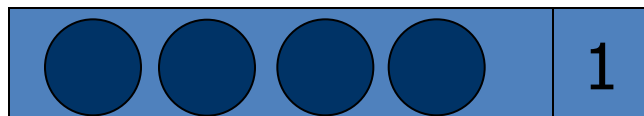
# Асимптотические свойства формулы Брента

$$t + \frac{q - t}{1} = q$$

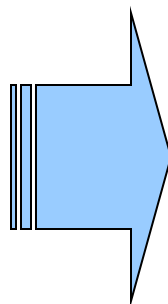
$$t + \frac{q - t}{n} \xrightarrow{n \rightarrow \infty} t$$



Количество CPU не  
ограничено

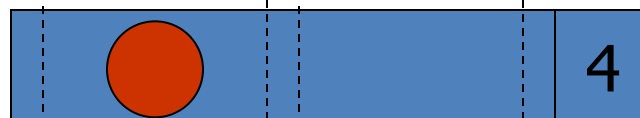
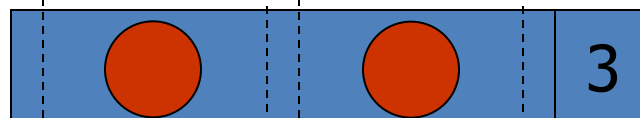
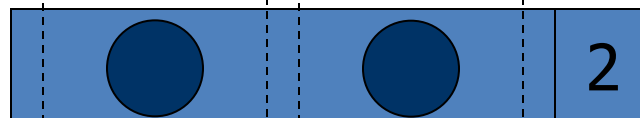
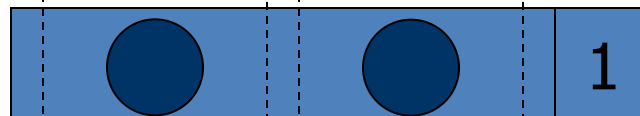


•  
•  
•  
•  
•



CPU I

CPU II



•  
•  
•  
•  
•

•  
•  
•  
•  
•



Пусть для бесконечного числа процессоров на  $i$ -м шаге выполнялось  $s_i$  операций, тогда при наличии  $n$  процессоров потребуется не более  $\left\lceil \frac{s_i}{n} \right\rceil$  операций.

$$\left\lceil \frac{s_i}{n} \right\rceil \leq \frac{s_i}{n} + 1 - \frac{1}{n} = \frac{s_i - 1}{n} + 1$$

$$t_n \leq \sum_{i=1}^t \left\lceil \frac{s_i}{n} \right\rceil \leq \sum_{i=1}^t \left( \frac{s_i - 1}{n} + 1 \right) = t + \frac{\sum_{i=1}^t (s_i - 1)}{n} = t + \frac{q - t}{n}$$



**СПАСИБО ЗА ВНИМАНИЕ!**  
**ЗАДАВАЙТЕ ВОПРОСЫ?**

