Введение в параллельные вычисления

д.т.н. Мокрова Наталия Владиславовна

асп. Морунов Егор, Сырко Денис

пятница	ауд. 119	
10:45 – 12:20	Лекция	
	1 неделя	2 неделя
12:50 - 16:10	Лаб – КС44	Лаб - КС40

Основные понятия

- Что такое «программирование?»
- Что такое программа?
- Что такое параллельный?

- Программирование – процесс создания программ, то есть разработка программного обеспечения. Большая часть работы программиста связана с написанием исходного кода на одном из языков программирования.
- Программой для ЭВМ является представленная в объективной форме совокупность данных и команд, предназначенных для функционирования ЭВМ и других компьютерных устройств с целью получения определенного результата, включая подготовительные материалы, полученные в ходе разработки программы для ЭВМ, и порождаемые ею аудиовизуальные отображения.

Гражданский кодекс РФ. ст. 1261

- Параллельный – происходящий одновременно.

Литература

- Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. СПб.: БХВПетербург, 2002.
- Немнюгин С.А., Стесик О.Л. Параллельное программирование для многопроцессорных вычислительных систем. СПб.: БХВПетербург, 2002.
- Эндрюс Г.Р. Основы многопоточного, параллельного и распределенного программирования. М.: Издательский дом «Вильямс», 2003.
- Антонов А.С. Параллельное программирование с использованием технологии MPI. М.: Издво МГУ, 2004.
- Антонов А.С. Параллельное программирование с использованием технологии OpenMP. М.: Издво МГУ, 2009.
- Воеводин Вл.В., Жуматий С.А. Вычислительное дело и кластерные системы. М.: Издво МГУ, 2007.
- Сальников А.Н. Вычислительная система на примере кластера UNIX серверов. Взгляд пользователя. Конспект лекций. МГУ, 2007.
- Богачев К.Ю. Основы параллельного программирования. М.: Бином, 2003
- Grama A., Gupta A., Karypis G., Kumar V. Introduction to Parallel Computing, Second Edition. Addison Wesley, 2003.
- Williams. C++ concurrency in action. Practical multithreading.

Супер-ЭВМ и сверхвысокая производительность

- Расчёт конфигурации подобных систем? Сколько стоят лишь 4 Тбайта оперативной памяти? Какие задачи настолько важны?
- Оптимизация процесса добычи нефти. Имеем подземный нефтяной резервуар с каким-то число пробуренных скважин: по одним на поверхность откачивается нефть, по другим обратно закачивается вода. Нужно смоделировать ситуацию в данном резервуаре, чтобы оценить запасы нефти или понять необходимость в дополнительных скважинах.
- Упрощенная схема: моделируемая область отображается в куб, размеры куба, при которых можно получать правдоподобные результаты – 100*100*100 точек. В каждой точке куба надо вычислить от 5 до 20 функций: три компоненты скорости, давление, температуру, концентрацию компонент (вода, газ и нефть – минимальный набор компонент). Значения функций находятся как решение нелинейных уравнений, что требует от 200 до 1000 арифметических операций. Если исследуется нестационарный процесс, то делается 100-1000 шагов по времени.

$$\mathbf{10^6(\text{точек сетки})*10(\text{функций})*500(\text{операций})*500(\text{шагов по времени}) = 2.5*10^{12}}$$

- **2500 миллиардов** арифметических операций для выполнения одного лишь расчета!
- А изменение параметров модели?
  А отслеживание текущей ситуации при изменении входных данных?

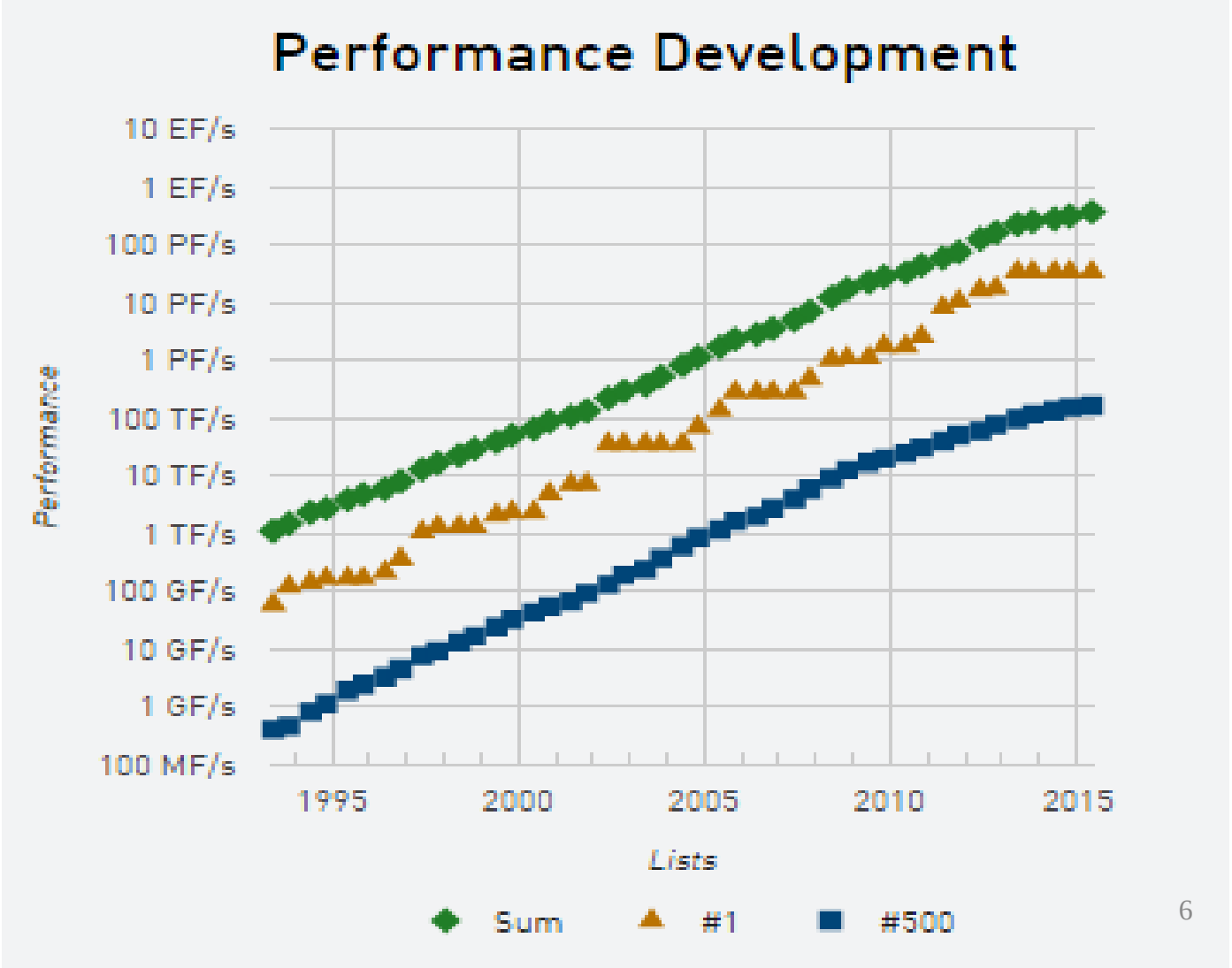
Как измеряют производительность компьютеров?

- Единица – измерений – Flops (флопс – операций с плавающей точкой в секунду).
- Пиковая производительность – максимальная теоретическая производительность.
- Бенчмарк – эталонная тестовая программа для оценки производительности.
- High Performance Linpack – тест, основанный на задаче решения системы линейных алгебраических уравнений.
- В настоящее время производительность наиболее мощных компьютеров достигла петафлопа 10^{15}.
- Как тестировать новые многоядерные вычислительные устройства Tesla, Nec?

Динамика прироста вычислительной мощности

- http://www.top500.org

- За полвека производительность компьютеров выросла более, чем в **семьсот миллионов** раз (http://parallel.ru)



Статистика 47-й редакции списка мощнейших компьютеров мира

от 20 июня 2016 года

http://parallel.ru/news/top500_47edition.html

- Сменился лидер списка, им стал китайский суперкомпьютер Sunway TaihuLight, разработанный в National Research Center of Parallel Computer Engineering & Technology и установленный в National Supercomputing Center in Wuxi. Суперкомпьютер использует китайские процессоры Sunway SW26010 260C, его пиковая производительность составляет 125 PFlop/s, а производительность на тесте Linpack - 93 PFlop/s.
- На втором месте списка остался прежний лидер, суперкомпьютер Tianhe-2, который установлен в National Supercomputer Center in Guangzho (Китай). Его пиковая производительность составляет 54.9 PFlop/s, а производительность на тесте Linpack - 33.86 PFlop/s.
- На третьем месте списка остался суперкомпьютер Titan Cray XK7, установленный в Oak Ridge National Laboratory (США), с производительностью на тесте Linpack 17.59 PFlop/s.
- Всего в текущей редакции списка уже 95 систем с производительностью на тесте Linpack более 1 PFlop/s (в ноябре было 81 такая система).
- Суммарная производительность систем в списке выросла за полгода с 420 до 566.7 PFlop/s.

Ограничения параллельных вычислений

Показано, на какое максимальное ускорение работы программы можно рассчитывать в зависимости от доли последовательных вычислений и числа доступных процессоров.

Предполагается, что параллельная секция может быть выполнена без каких-либо дополнительных накладных расходов. Так как в программе всегда присутствует инициализация, ввод/вывод и некоторые сугубо последовательные действия, то недооценивать данный фактор нельзя – практически вся программа должна исполняться в параллельном режиме, что можно обеспечить только после анализа всей программы (!)

Число ПЭ	Доля последовательных вычислений				
	50%	25%	10%	5%	2%
2	1.33	1.60	1.82	1.90	1.96
8	1.78	2.91	4.71	5.93	7.02
32	1.94	3.66	7.80	12.55	19.75
512	1.99	3.97	9.83	19.28	45.63
2048	2.00	3.99	9.96	19.82	48.83

Максимальное ускорение работы программы в зависимости от доли последовательных вычислений и числа используемых процессоров

Закон Амдала

Позволяет оценить ускорение S, которое может быть получено на компьютере из p процессоров при данном значении f

$$S \leq \frac{1}{f+(1-f)/p}$$

где f – доля операций, которые нужно выполнять последовательно $0 <= f <= 1$

Программа полностью параллельна ($f = 0$) и полностью последовательна ($f = 1$).

- Если 9/10 программы исполняется параллельно, а 1/10 последовательно, то ускорения более, чем в 10 раз получить в принципе невозможно вне зависимости от качества реализации параллельной части кода и числа используемых процессоров.
- Следствие закона Амдала: для того чтобы ускорить выполнение программы в q раз необходимо ускорить не менее, чем в q раз не менее, чем (1-1/q)-ю часть программы. Следовательно, если есть желание ускорить программу в 100 раз по сравнению с ее последовательным вариантом, то необходимо получить не меньшее ускорение не менее, чем на 99.99% кода!
- Вывод – прежде, чем переделывать код для перехода на параллельный компьютер надо основательно подумать. Если доля последовательных операций велика, то на значительное ускорение рассчитывать явно не приходится и нужно думать о замене отдельных компонент алгоритма.

Необходимо тщательное согласование структуры программ и алгоритмов с особенностями архитектуры параллельных вычислительных систем.

См. 60 землекопов

Архитектура фон Неймана

Как ускорить работу компьютера?

- Усовершенствование оборудования (повышение тактовой частоты процессоров, шин, памяти).
- Параллельная обработка.
- Конвеерная обработка.
- Многоядерные архитектуры.

Процессор (CPU)

Конвеерная обработка

Цикл выполнения команды:

- выборка команды;
- декодирование команды, вычисление адреса операнда;
- выборка операнда;
- выполнение команды;
- обращение к памяти;
- запись результата в память.

Классификация существующих параллельных вычислительных систем

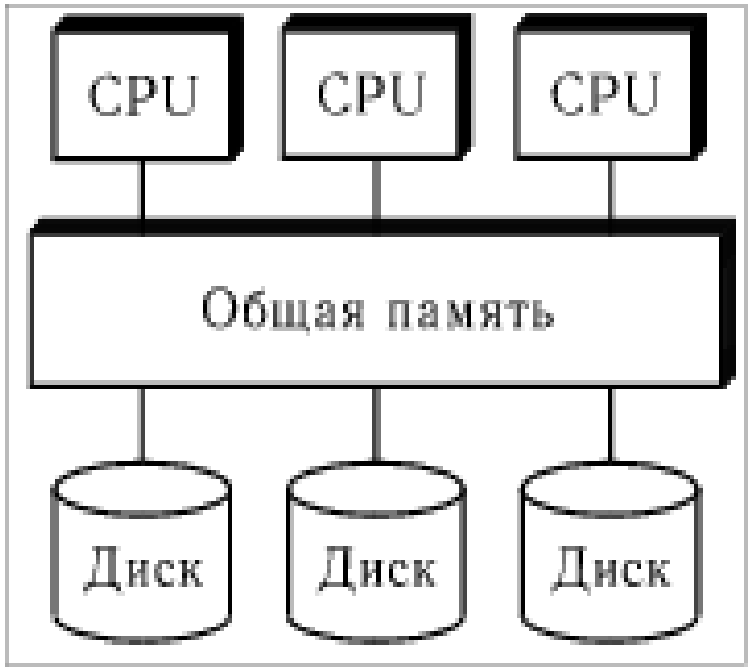
По способу доступа к памяти

- MPP – системы распределенной памяти;
- SMP – симметричные мультипроцессорные системы общей памяти;
- NUMA – память физически распределена, но логически общедоступна (среднее между SMP и MPP);
- PVP-системы с использованием общей или распределенной памяти с поддержкой команд обработки векторных данных (векторно-конвейерные процессоры);
- Кластеры – экономичный вариант MPP.

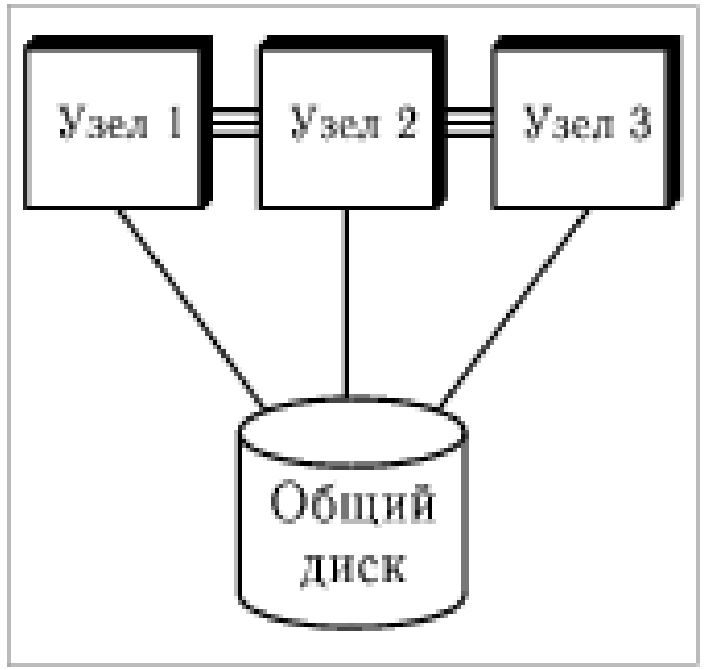
Большую популярность приобретают идеи комбинирования различных архитектур в одной системе и построения неоднородных систем.

При организациях распределенных вычислений в глобальных сетях (Интернет) говорят о мета-компьютерах, которые, строго говоря, не представляют из себя параллельных архитектур.

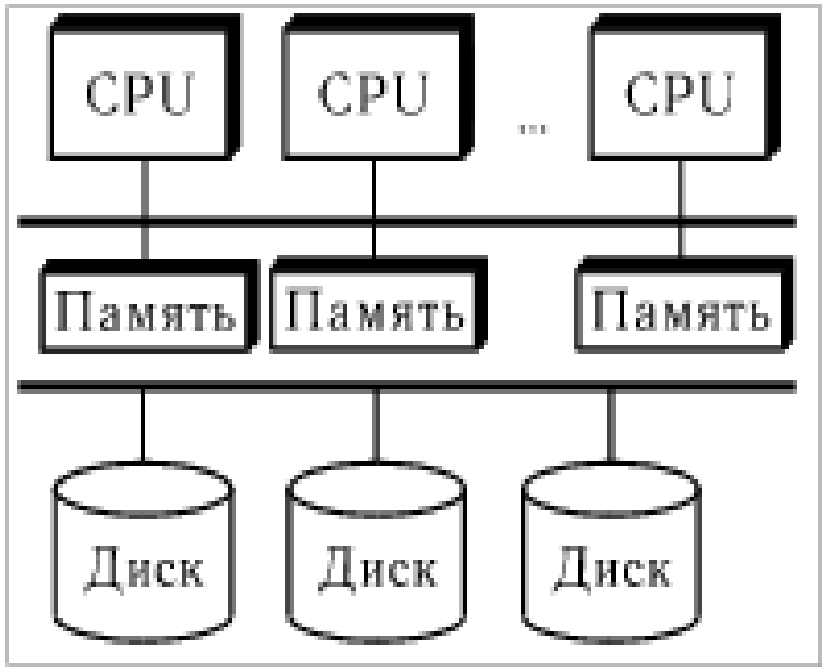
Схемы параллельных вычислительных систем



SMP система



Кластер



MPP система

MPP

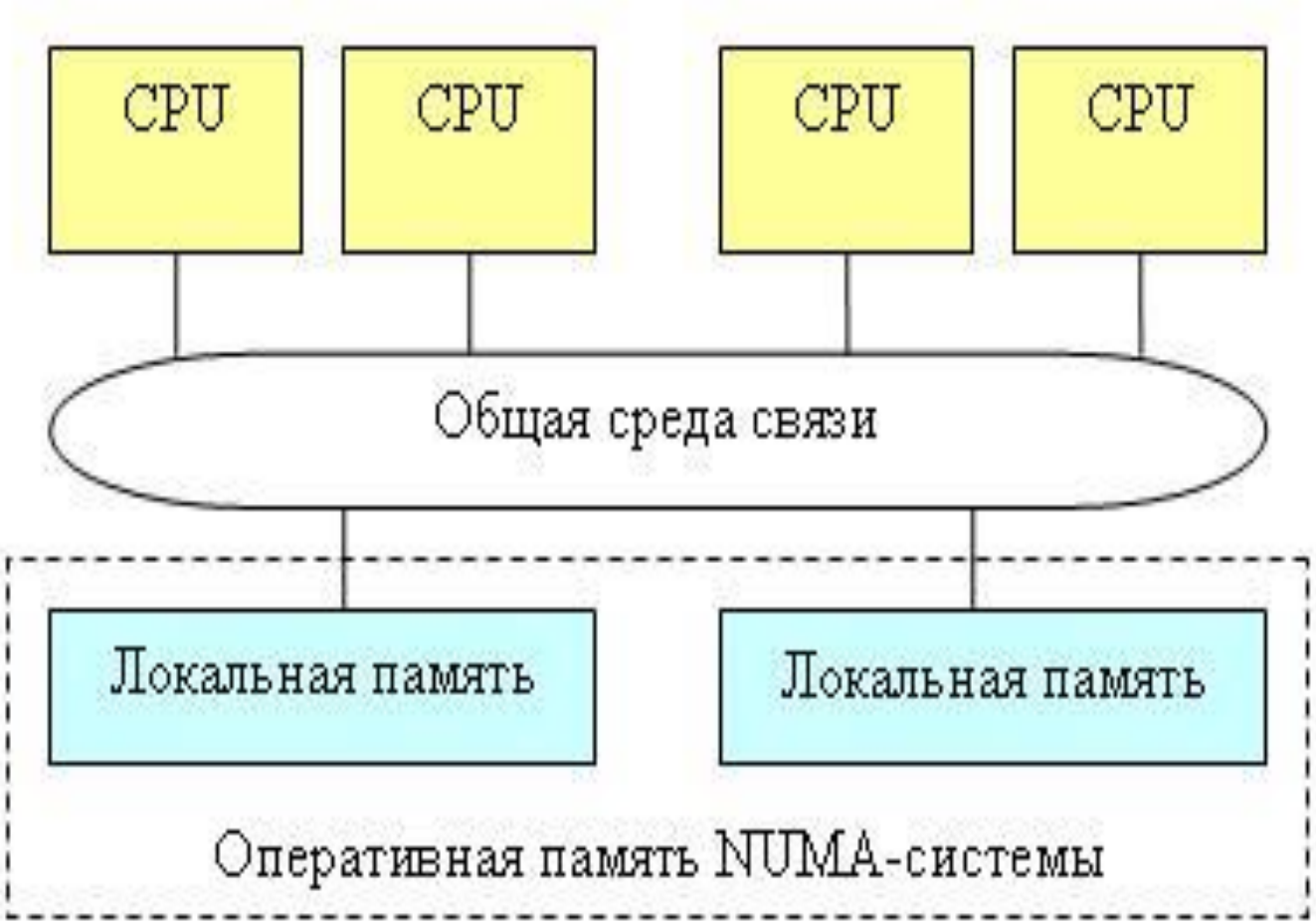
Архитектура	Система из однородных вычислительных узлов, включающих: один или несколько центральных процессоров (обычно RISC), локальную память (прямой доступ к памяти других узлов невозможен), коммуникационный процессор или сетевой адаптер иногда - жесткие диски (как в SP) и/или другие устройства В/В Могут быть добавлены специальные узлы ввода-вывода и управляющие узлы, связанные через коммуникационную среду (высокоскоростная сеть, коммутатор и т.п.)
Примеры	IBM RS/6000 SP2, Intel PARAGON/ASCI Red, CRAY T3E, Hitachi SR8000, транспьютерные системы Parsyte.
Масштабируемость	Общее число процессоров в реальных системах достигает нескольких тысяч (ASCI Red, Blue Mountain).
Операционная система	Существуют два основных варианта: Полноценная ОС работает только на управляющей машине (front-end), на каждом узле работает сильно урезанный вариант ОС, обеспечивающие только работу расположенной в нем ветви параллельного приложения. Пример: Cray T3E. На каждом узле работает полноценная UNIX-подобная ОС (вариант, близкий к кластерному подходу). Пример: IBM RS/6000 SP + ОС AIX, устанавливаемая отдельно на каждом узле.
Модель программирования	Программирование в рамках модели передачи сообщений (MPI, PVM, BSPlib)

SMP

Архитектура	Система из нескольких однородных процессоров и массива общей памяти (обычно из нескольких независимых блоков). Все процессоры имеют доступ к любой точке памяти с одинаковой скоростью. Процессоры подключены к памяти либо с помощью общей шины (базовые 2-4 процессорные SMP-сервера), либо с помощью crossbar-коммутатора (HP 9000). Аппаратно поддерживается когерентность кэшей.
Примеры	HP 9000 V-class, N-class; SMP-сервера и рабочие станции на базе процессоров Intel (IBM, HP, Compaq, Dell, ALR, Unisys, DG, и др.).
Масштабируемость	Наличие общей памяти упрощает взаимодействие процессоров между собой, но накладывает ограничения на их число – не более 32 в реальных системах. Для построения масштабируемых систем на базе SMP используются кластерные или NUMA-архитектуры.
Операционная система	Система работает под управлением единой ОС (обычно UNIX-подобной, но для Intel-платформ поддерживается Windows NT). ОС автоматически распределяет процессы/нити по процессорам (scheduling), но иногда возможна и явная привязка.
Модель программирования	Программирование в модели общей памяти. (POSIX threads, OpenMP). Для SMP-систем существуют сравнительно эффективные средства автоматического распараллеливания.

NUMA – системы (Nonuniform memory access, неоднородный доступ к памяти)

Так устроена Tesla



(citforum.ru)

NUMA

Архитектура	Состоит из однородных базовых модулей (плат), состоящих из небольшого числа процессоров и блока памяти. Модули объединены с помощью высокоскоростного коммутатора. Поддерживается единое адресное пространство, аппаратно поддерживается доступ к удаленной памяти. Доступ к локальной памяти в несколько раз быстрее, чем к удаленной. Если аппаратно поддерживается когерентность кэшей во всей системе, говорят об архитектуре cc-NUMA (cache-coherent NUMA)
Примеры	HP HP 9000 V-class в SCA-конфигурациях, SGI Origin2000, Sun HPC 10000, IBM/Sequent NUMA-Q 2000, SNIRM600.
Масштабируемость	Масштабируемость ограничивается объемом адресного пространства, возможностями аппаратуры поддежки когерентности кэшей и возможностями операционной системы по управлению большим числом процессоров. На настоящий момент, максимальное число процессоров в 256 (Origin2000).
Операционная система	Вся система работает под управлением единой ОС, как в SMP. Но возможны варианты динамического "подразделения" системы, когда отдельные "разделы" системы работают под управлением разных ОС (например, Windows NT и UNIX в NUMA-Q 2000).
Модель программирования	Аналогично SMP.

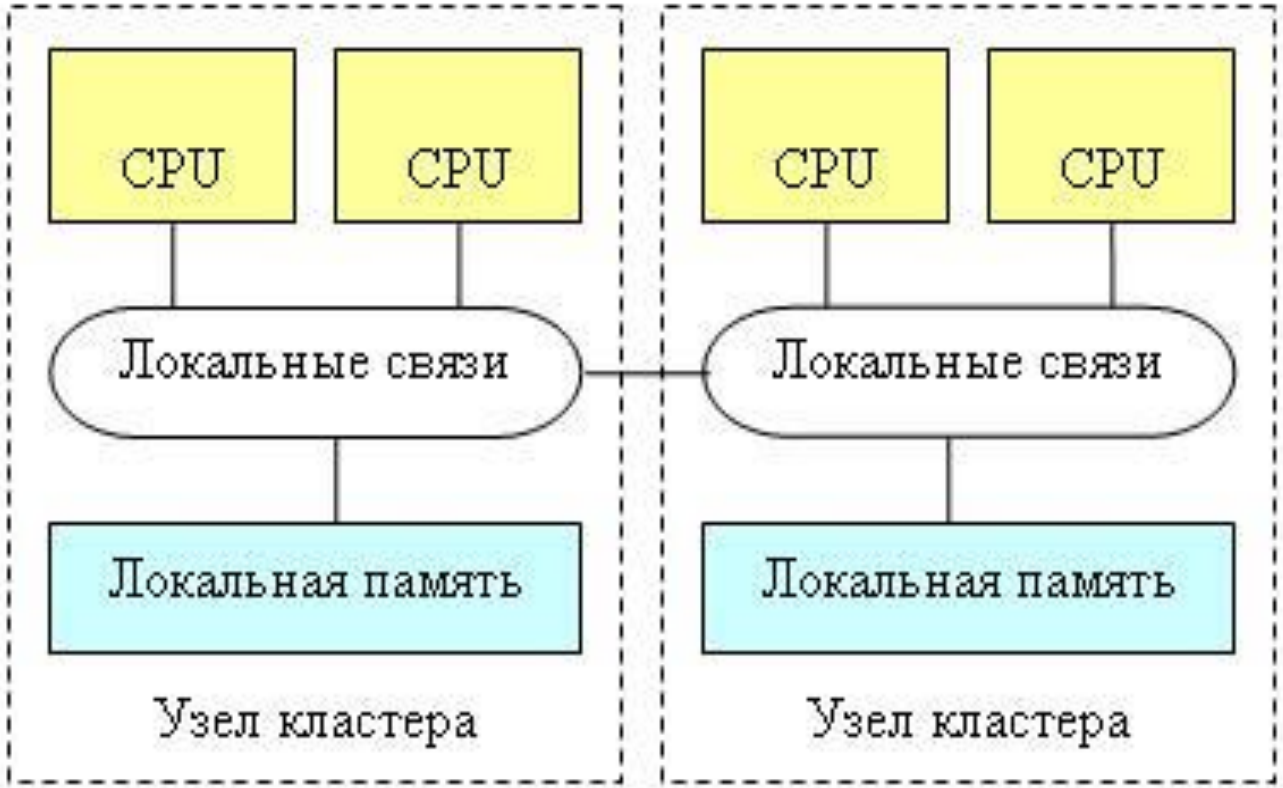
PVP

Архитектура	Основной признак PVP-систем – наличие специальных векторно-конвейерных процессоров, в которых предусмотрены команды однотипной обработки векторов независимых данных, выполняющиеся на конвейерных функциональных устройствах. Как правило, несколько таких процессоров (1-16) работают одновременно над общей памятью (аналогично SMP) в рамках многопроцессорных конфигураций. Несколько таких узлов могут быть объединены с помощью коммутатора (аналогично MPP).
Примеры	NEC SX-4/SX-5, линия векторно-конвейерных компьютеров CRAY: от CRAY-1, CRAY J90/T90, CRAY SV1,CRAY X1, серия Fujitsu VPP.
Модель программирования	Эффективное программирование подразумевает векторизацию циклов (для достижения разумной производительности одного процессора) и их распараллеливание (для одновременной загрузки нескольких процессоров одним приложением).

Кластерные системы

Кластер – набор связанных компьютеров, который используется как единый вычислительный ресурс.

- Легко масштабируется.
- Управляется обычной операционной системой Linux.
- Имеет простую коммуникационную среду.
- Узлы кластера могут быть однопроцессорными и многопроцессорными.
- Кластер может быть создан на основе компьютерного класса.
- Кластер может быть собран в стойку из серверов-лезвий.

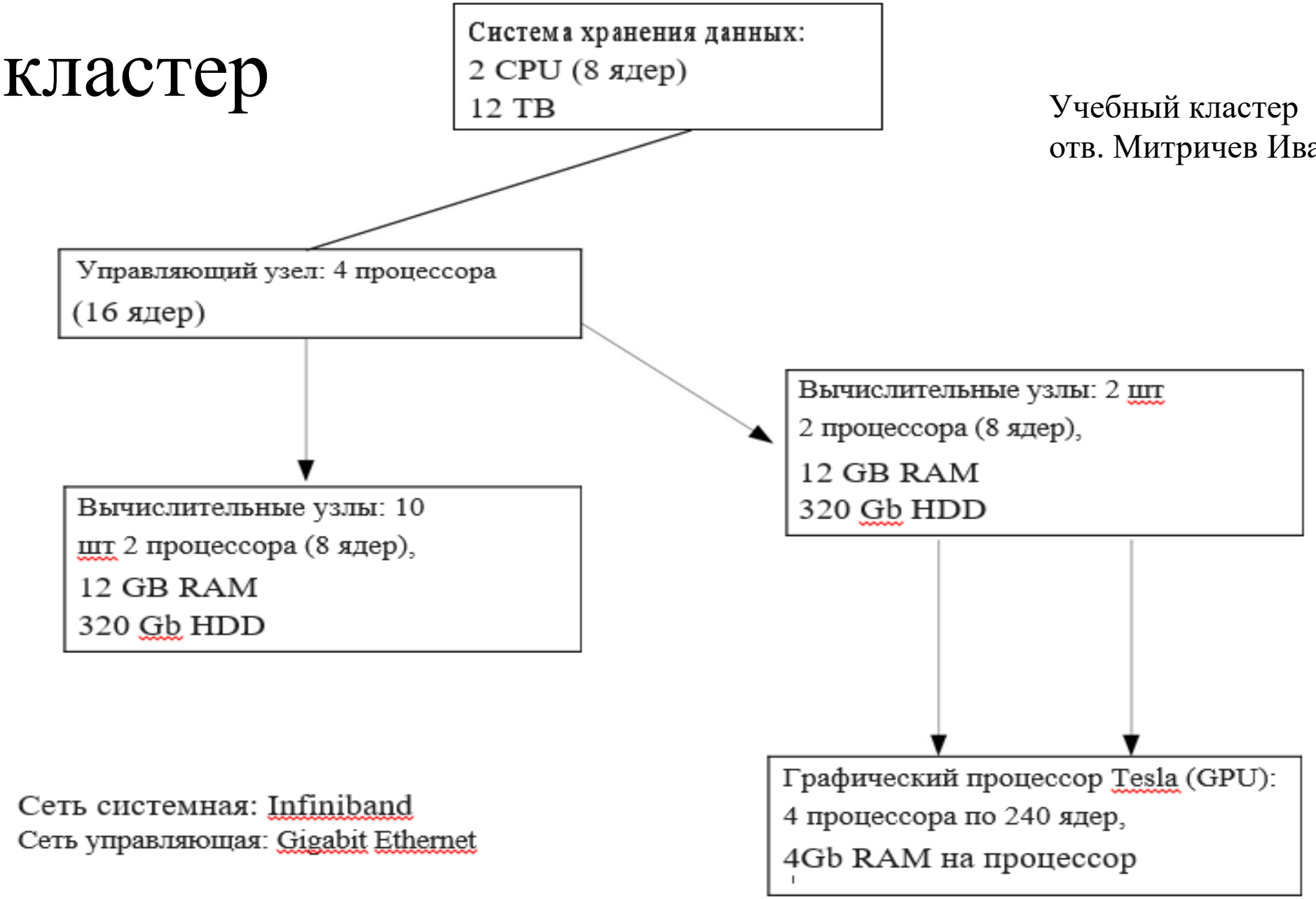


citforum.ru

Кластерные системы

Архитектура	Набор рабочих станций (или ПК) общего назначения, используется в качестве дешевого варианта массивно-параллельного компьютера. Для связи узлов используется одна из стандартных сетевых технологий (Fast/Gigabit Ethernet, Myrinet) на базе шинной архитектуры или коммутатора. Объединение в кластер компьютеров разной мощности или разной архитектуры дает огетерогенные (неоднородные) кластеры. Узлы кластера могут одновременно использоваться в качестве пользовательских рабочих станций, узлы могут быть существенно облегчены и/или установлены в стойку.
Примеры	NT-кластер в NCSA, Beowulf-кластеры.
Операционная система	Используются стандартные для рабочих станций ОС, свободно распространяемые – Linux/FreeBSD, со специальными средствами поддержки параллельного программирования и распределения нагрузки.
Модель программирования	Программирование в рамках модели передачи сообщений (чаще – MPI). Дешевизна подобных систем оборачивается большими накладными расходами на взаимодействие параллельных процессов между собой, что сужает потенциальный класс решаемых задач.

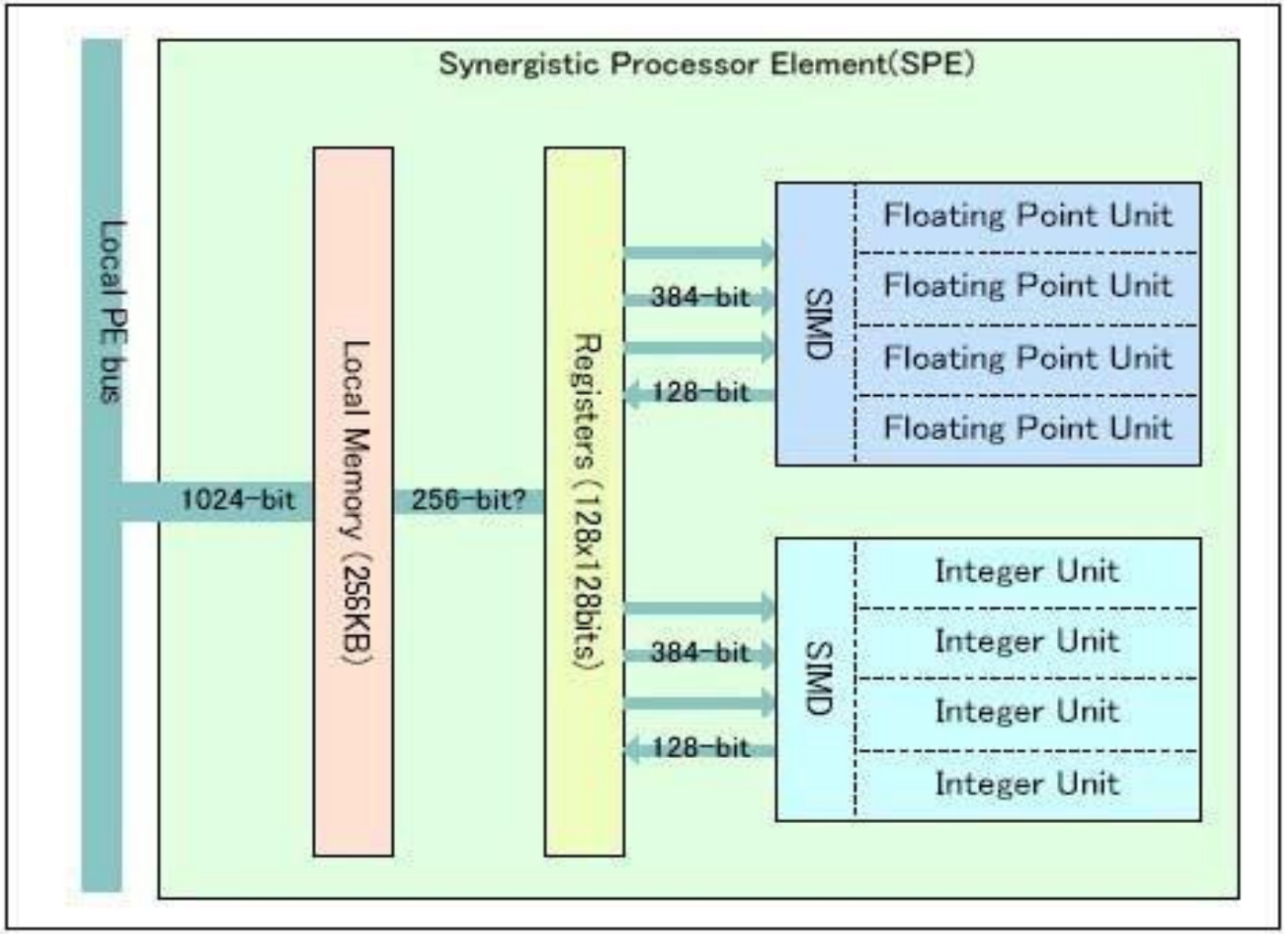
Наш кластер



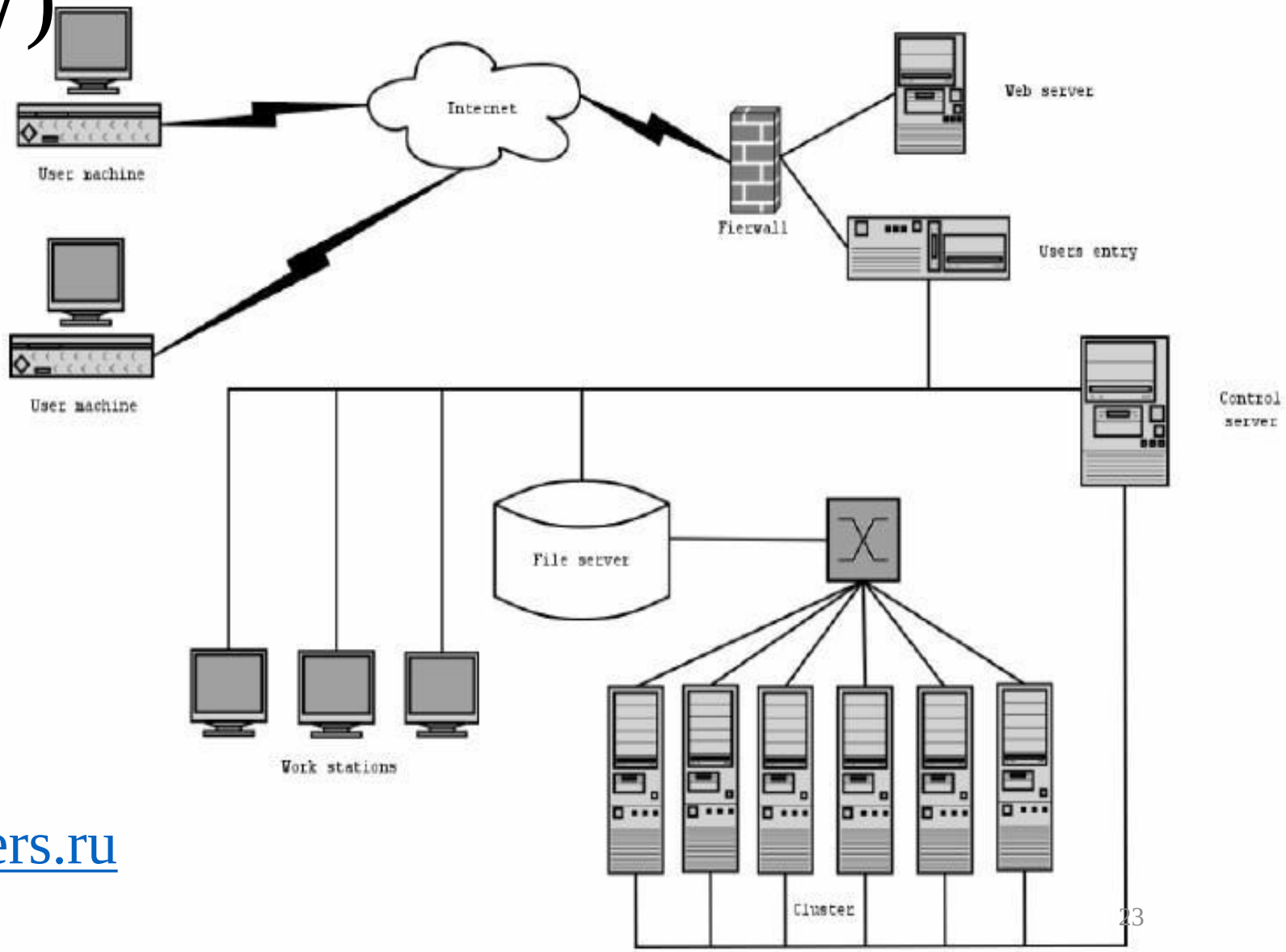
Учебный кластер
отв. Митричев Иван

Векторно-конвеерные системы

Cray, NEC, SSE (Streaming SIMD Extensions).



Состав вычислительного комплекса (Сальников, 2007)



http://www.supercomputers.ru

Работа в консоли (Linux)

Alt + F2 – запуск приложения по команде.

	KDE	Gnome
1. Консоль	konsole	gnome-terminal
2. Редактор	Kwrite	gedit
3. Вид (обычно)	Панель внизу	Панель вверху
	Виджеты	Файлы

- Сохранение?
- Компиляция g++ 001.cpp -o name0209
- Запуск ./name0209
- Time time ./name0209

- Альтернатива:
http://coliru.stacked-crooked.com/
g++ -std=c++11 -pthread main.cpp && ./a.out

Hello world

```
#include <iostream>
#include <thread>

using namespace std;

void thread_function() {
        cout<<"My first thread"<<endl;
        }

int main() {
        thread t1(thread_function);
        t1.join();
        return 0;
}
```

- вызов функции из потока;
- запустить поток;
- инструкция главному потоку подождать завершения дочернего потока.

Hello mas

1. Присутствие.
2. Решение задачи стандарт.
3. Решение задачи поток.
4. Вывод – размерность, выгода.

Время работы

```
clock_t start, end;

start = clock();
```

5. Оптимизация.
6. Отчет.

Где искать информацию

Обучающие примеры

- http://www.baptistewicht.com/2012/03/cpp11concurrencypart1startthreads/
- http://www.baptistewicht.com/2012/03/cp11concurrencytutorialpart2protectshareddata/
- http://www.baptistewicht.com/2012/04/c11concurrencytutorialadvancedlockingandcondition-variables/
- http://www.baptistewicht.com/2012/07/c11concurrencytutorialpart4atomictype/
- http://www.baptistewicht.com/2012/07/c11synchronizationbenchmark/
- http://solarianprogrammer.com/2011/12/16/cpp11threadtutorial/
- http://solarianprogrammer.com/2012/02/27/cpp11threadtutorialpart2/
- http://solarianprogrammer.com/2012/05/09/cpp11threadtutorialpart3/

Справочник

- http://en.cppreference.com/w/cpp/thread