

Институт автоматизации и процессов управления ДВО РАН

Центр коллективного пользования «Дальневосточный
вычислительный ресурс» ДВО РАН

Практика использования многопроцессорных вычислительных систем

к.ф.-м.н. Голенков Евгений Александрович,
к.т.н. Харитонов Дмитрий Иванович,
вед. инж.-прогр. Тарасов Георгий Витальевич,
инж.-прогр. Одякова Дарья Сергеевна

Владивосток
2011

План лекций. Расписание

1. Технологическая цепочка обработки данных на МВС (многопроцессорная вычислительная система)
2. Простой пример параллельной программы
3. Топология вычислительной системы и топология вычислительного процесса (с упором на оптимизацию)
4. Сложный пример параллельной программы
5. «Как собрать кластер самому?» - mini HOWTO

9 ноября 2011 года, 10:00 — 13:00: Лекция 1 и 2

10 ноября 2011 года, 10:00 — 13:00: Лекция 3 и 4

11 ноября 2011 года, 10:00 — 12:00: Лекция 5

Место проведения: ИАПУ ДВО РАН
к. 510 (ул. Радио, д. 5)

Лекция 1:

Технологическая цепочка обработки данных на МВС (многопроцессорная вычислительная система)

План:

- 1.Краткая архитектура МВС общего назначения
 - 2.Интерфейс пользователя, базовый набор протоколов
 - 3.Общая схема технологической цепочки обработки данных
 - 4.Подготовка программы (совместимость и переносимость)
-
- 5.Подготовка исходных данных (big-endian vs. little-endian)
 - 6.Запуск задачи (основы работы с системой очередей)
 - 7.Вычислительный процесс (контрольные точки)
 - 8.Формирование и получение результатов обработки
 - 9.Обработка результатов (визуализация, графика, видео)

Цели и задачи МВС

- Цель:
 - Быстрое, эффективное выполнение больших и сверхбольших объемов вычислений (обработки больших или сверхбольших объемов данных)
- Задачи:
 - Управление пользователями
 - Управление программами и заданиями пользователей
 - Управление данными пользователей
 - Мониторинг надежности и энергоэффективность

«...быстрое...», «...[сверх]большое...»

Абстрактный пример: задача моделирования климата
(Вл.В.Воеводин: Выч. мат. и струк. алг.)

Моделирование атмосферных процессов на протяжении 100 лет
Высота: допустим 40 слоев

Сетка: 1 градус

(грубо: $1852 \text{ м} * 60 = 111,12 \text{ км}$) = $180 \times 180 \times 2 * 40 \approx 2,6 \cdot 10^6$ ячеек

Модельное время:

100 лет / допустим 10 минут = $100 * 365 * 24 * 60 / 10 \approx 5,2 \cdot 10^6$ итераций

Итого: операций = $2,6 \cdot 10^6 * 5,2 \cdot 10^6 \approx 13,5 \cdot 10^{12}$

Величин: допустим 10 штук $\approx 1,4 \cdot 10^{14}$

Промежуточные вычисления: допустим еще 10^2 - 10^3

ИТОГО: 10^{16} — 10^{17} операций
(звезд в нашей галактике $\approx \approx \approx 100 \cdot 10^9$)

Стандартный ПК требуется:

$10^{17} \text{ оп} / 37 \cdot 10^9 \text{ оп/с} \approx 0,03 \cdot 10^8 \text{ с} \approx 833 \text{ ч} \approx 34 \text{ дня}$

Примеры из жизни: ТОИ (модель атмосферы), ИАПУ (инженерные расчеты), ИАПУ+ДВФУ (исследование наноструктур), ТИБОХ (моделирование лекарственных форм), ИАПУ (обработка спутниковых данных),

«...ЭФФЕКТИВНОЕ...»

- Рассмотренный пример очень идеален.

- Величина ускорения

$$\text{Speedup [S]} = T_1 / T_p \quad 100 / 25 (p = 10) = 4$$
$$100 / 25 (p = 100) = 4$$

- Величина эффективности

$$\text{Efficiency [E]} = S / p \quad 4 / 10 = 0,4 = 40\%$$
$$4 / 100 = 0,04 = 4\%$$

- Закон Амдаля (1967)

$$\text{Speedup [S]} = \frac{1}{\alpha + \frac{1-\alpha}{p}}$$

a\p	10	100	1000
0%	10	100	1000
10%	5,263	9,174	9,910
25%	3,077	3,883	3,988
40%	2,174	2,463	2,496

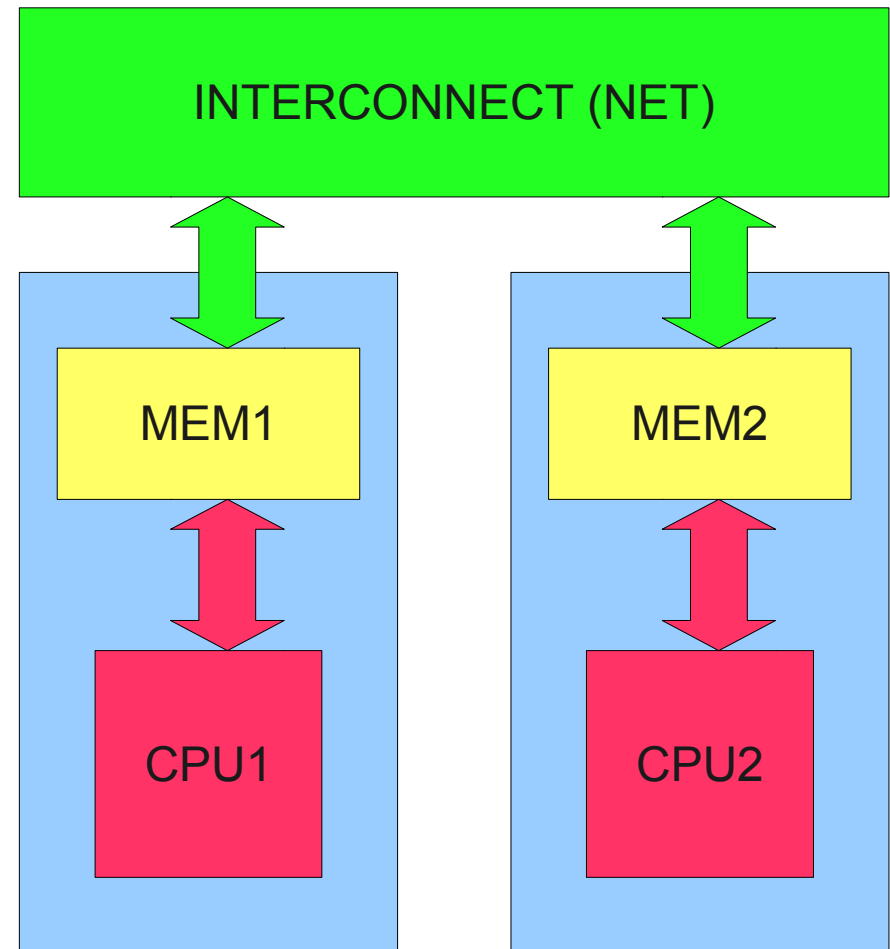
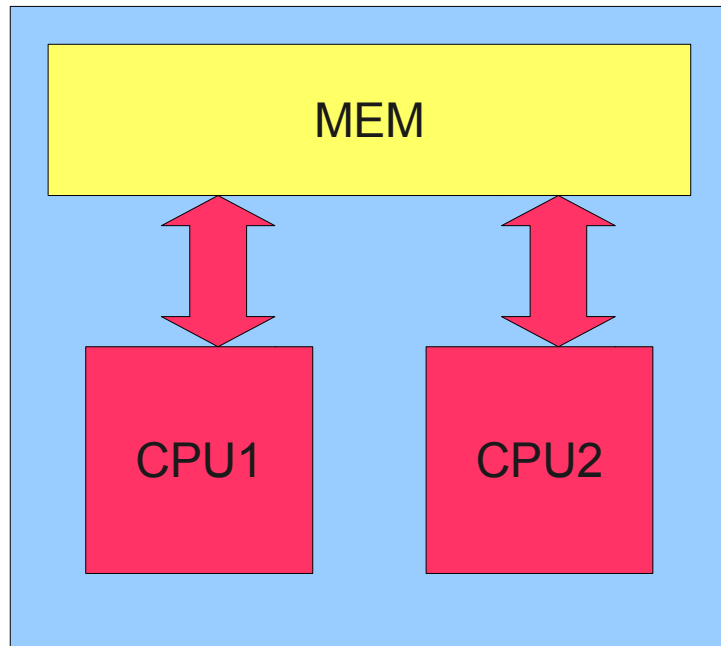
- Причины:

- Аппаратное обеспечение
- Программное обеспечение

Структура МВС

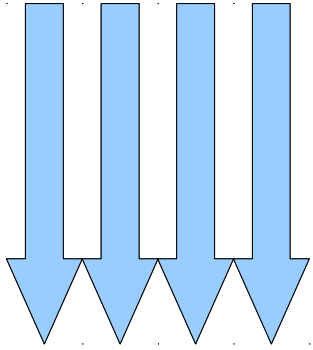
- Типичная МВС общего назначения состоит из:
 - Вычислительные узлы (рабочие лошадки)
 - Управляющая машина
 - Система хранения данных
 - Сетевая подсистема МВС

Архитектура общей памяти и распределенной памяти

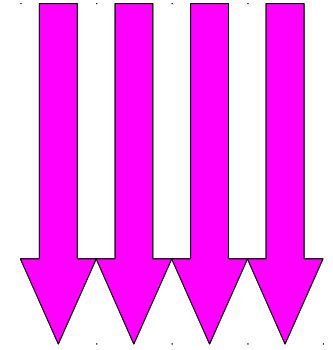


Систематика Флинна

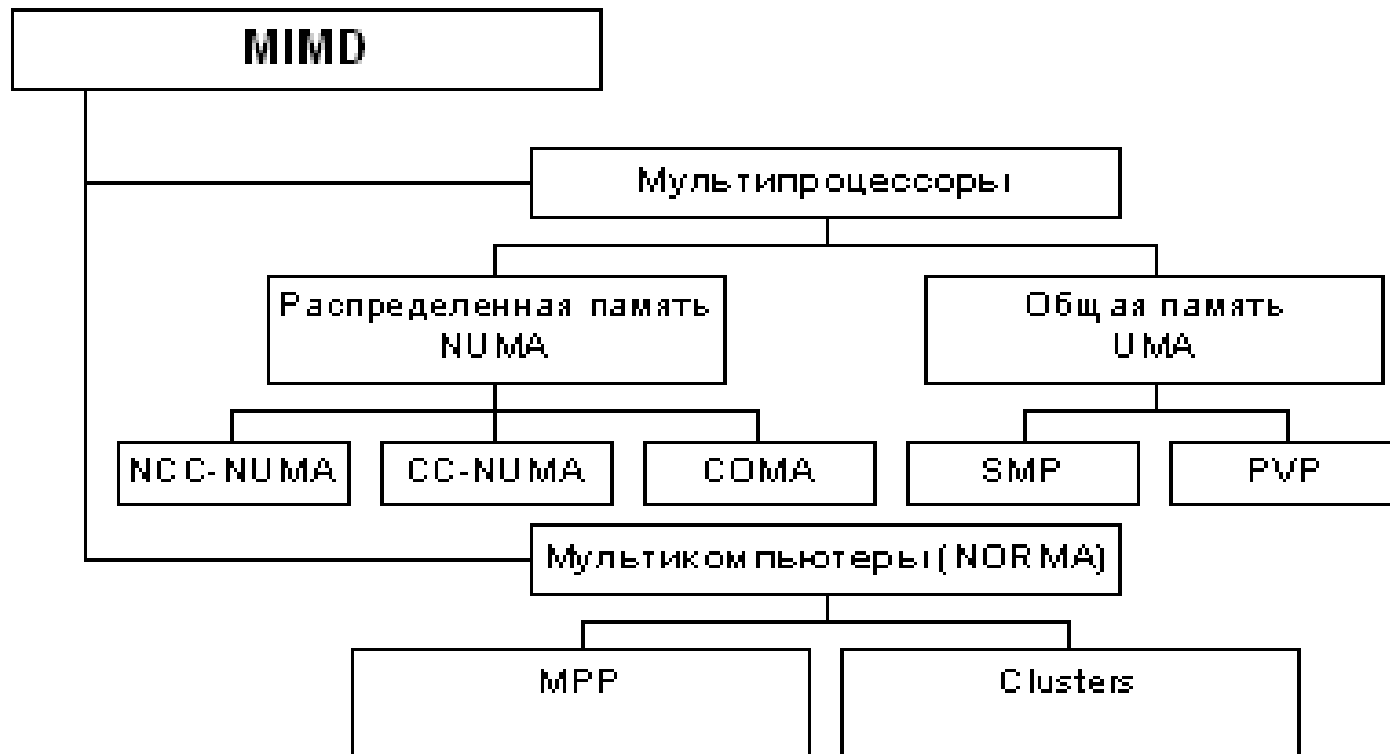
Instruction



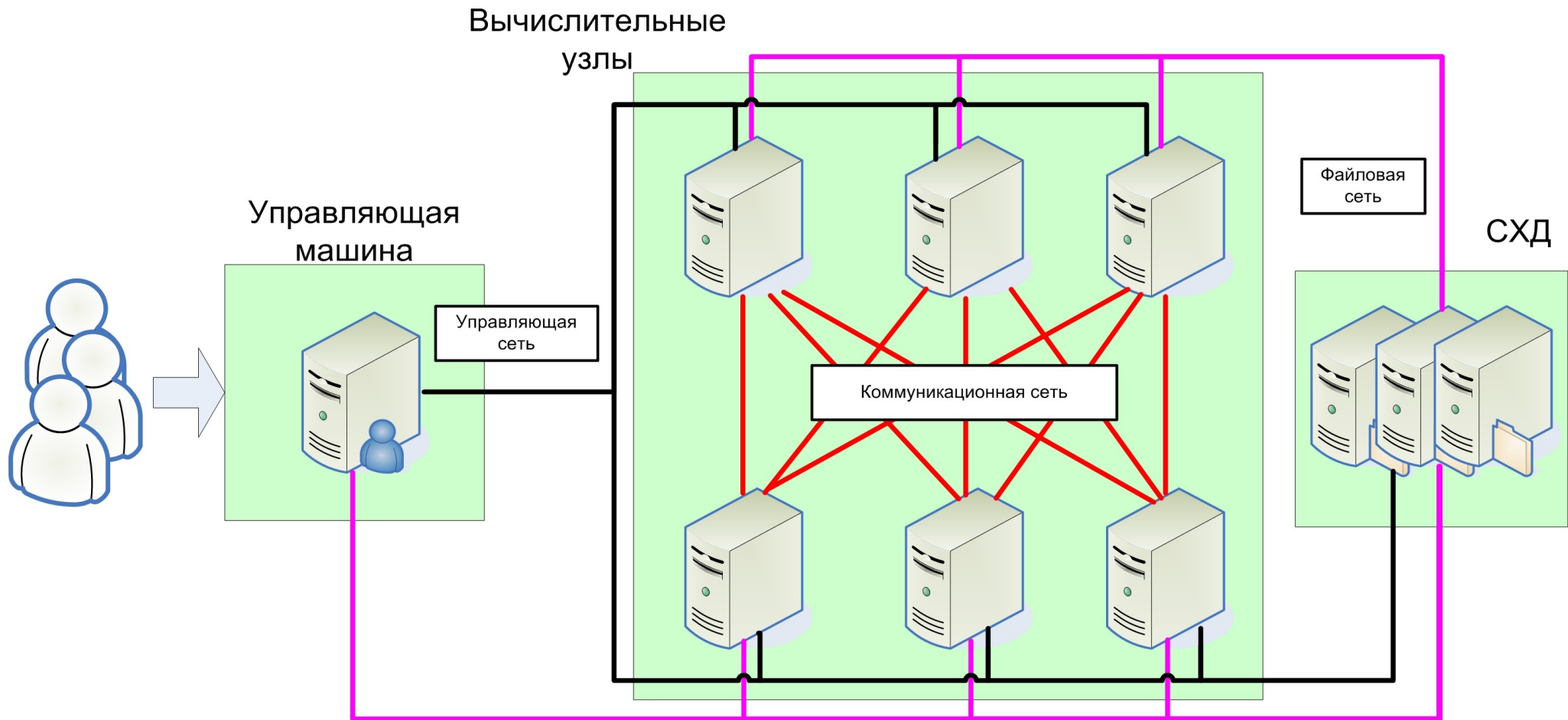
Data



SISD (Single Instruction, Single Data)
SIMD (Single Instruction, Multiple Data)
MISD (Multiple Instruction, Single Data)
MIMD (Multiple Instruction, Multiple Data)



Структура МВС. Логические элементы МВС



- Коммуникационная сеть (Myrinet, Infiniband, SCI, ...)
- Файловая сеть (10G Ethernet, Infiniband, ...)
- Управляющая сеть (Gigabit Ethernet)

Структура МВС. Управляющая машина



Управление пользователями:
(идентификация, авторизация)

Управление задачами:
(система управления
вычислительными узлами и
очередями)

Управление файлами:
(каталоги пользователей,
каталоги с предустановленным
программным обеспечением)

Принцип: Single System Image

Структура МВС. Система хранения данных



Управление сетевой файловой системой: NFS, GPFS, Lustre

Предоставление сетевых каталогов.

Система резервного копирования

Пользователь напрямую не работает с СХД. Аппаратный контроллер управления дисками (RAID-5)

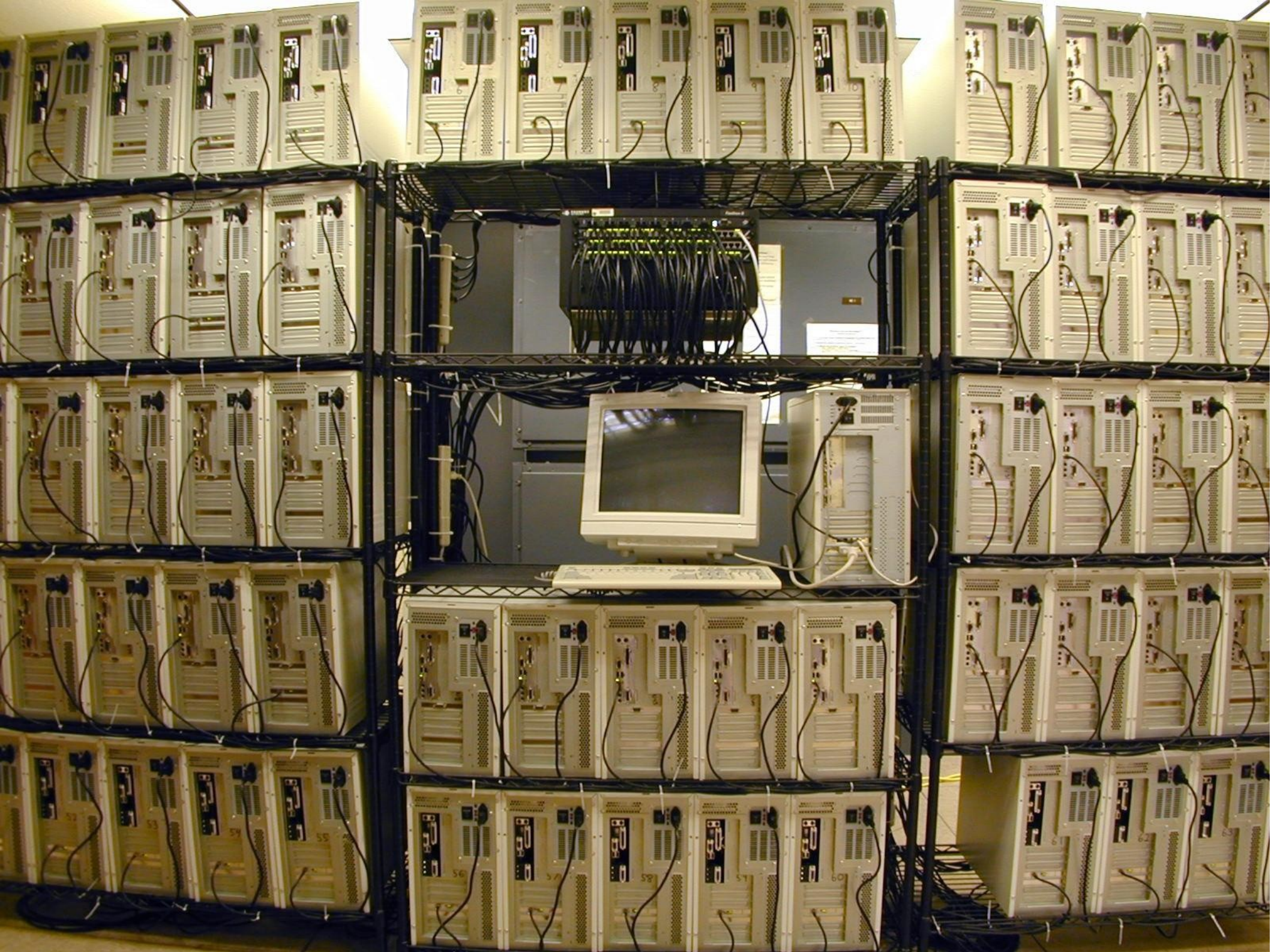
Работу с данными обеспечивается на системном уровне драйверами и службами сетевой файловой системы.

К вопросу об эффективности: Важна скорость доставки данных от СХД до узла и обратно!

Структура МВС. Вычислительные узлы

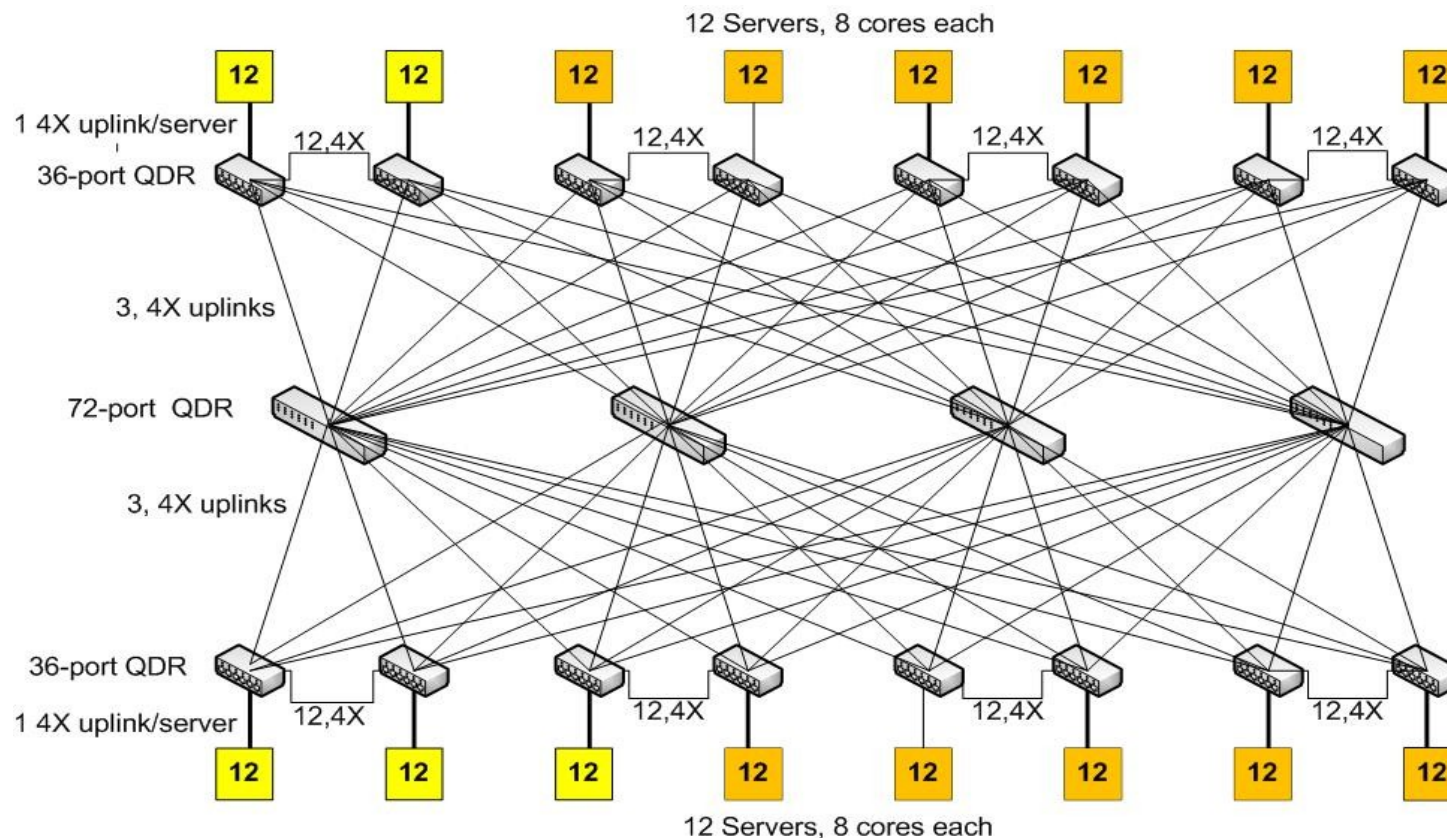
- Каждый узел:
 - Имеет 1, 2, 4 процессора (процессоры могут быть многоядерными) — рабочие лошадки
 - Локальная оперативную память, к которой эти процессоры имеют доступ
 - Локальная дисковая память для хранения образа операционной системы и промежуточных данных расчетов
 - Сетевые интерфейсы для связи с другими узлами и другими компонентами МВС.
 - Специализированные устройства (GPU, матричные процессоры и т. п.) - вспомогательный «табун»
 - Работает под своим образом операционной системы (традиционно — это Linux)





Архитектура МВС. Сетевая подсистема

- Характеристики:
 - Пропускная способность
 - Латентность (задержки при передаче)
 - Масштабируемость



Производительность МВС

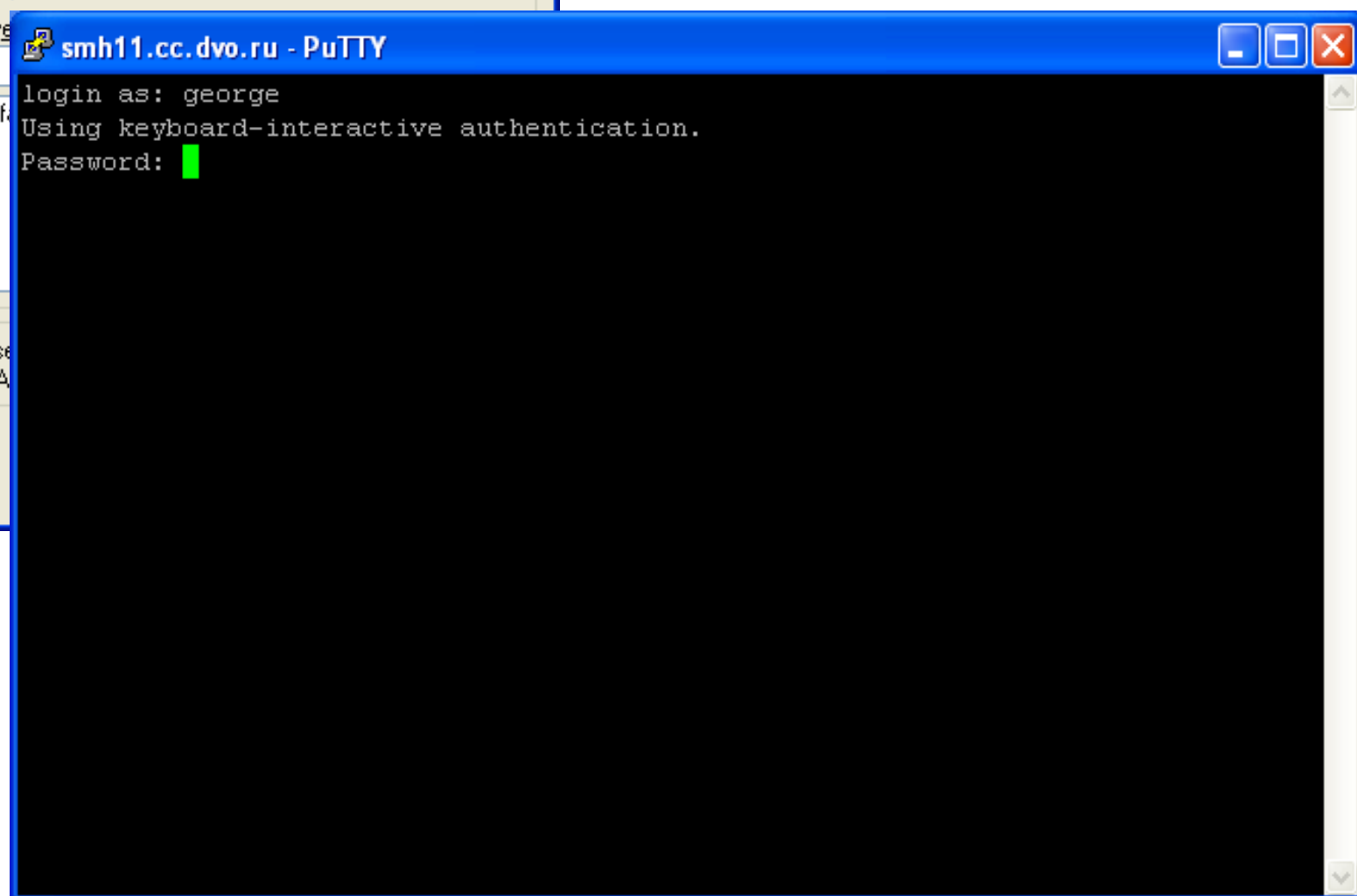
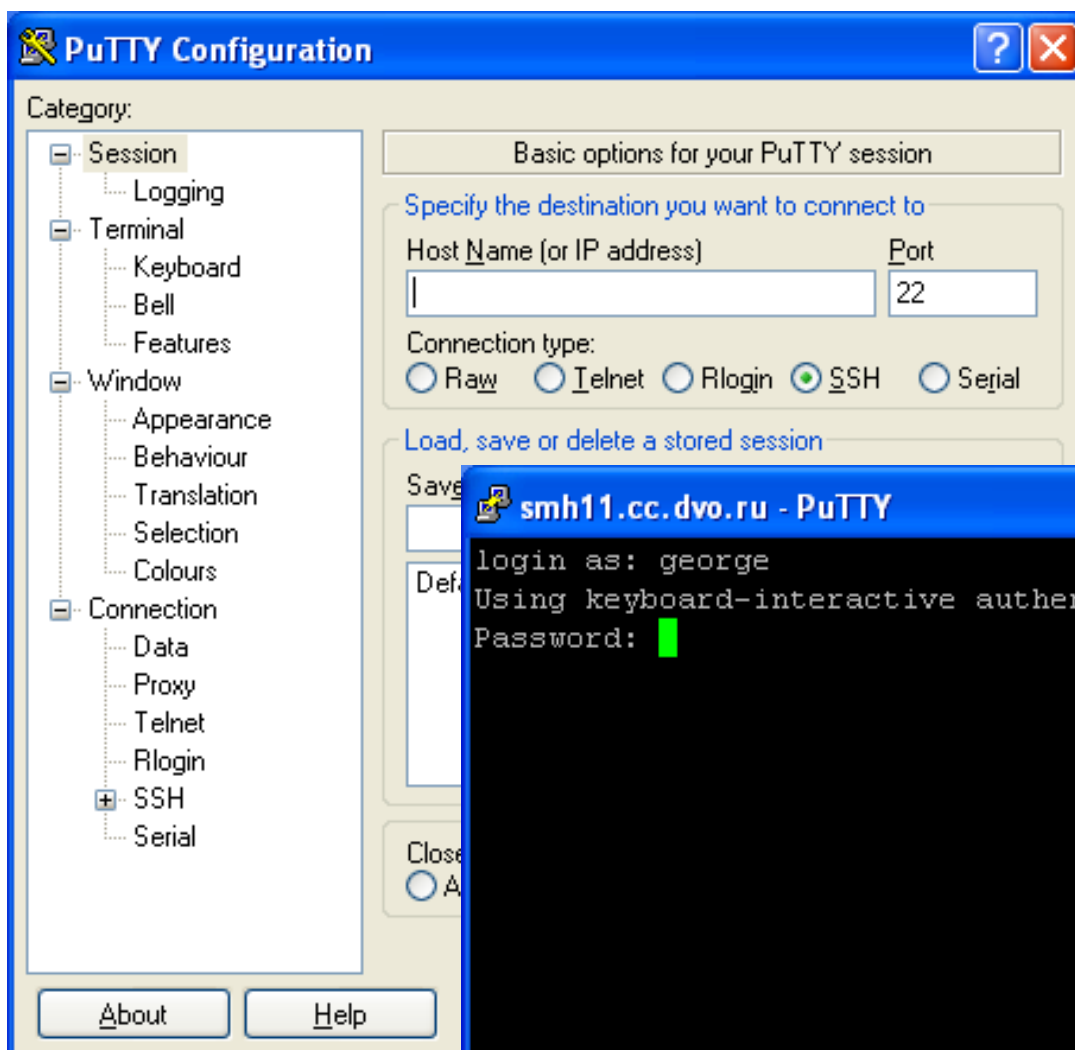
- Пиковая производительность R_{peak}
 - Формула расчета:
 - $Node * CPU * Core * FPU * Freq (GHz)$
 - Пример: Первая очередь кластера SMH11 (ИАПУ)
 - $10 * 4 * 12 * 4 * 1.7 GHz = 3264 GFLOPS$
- Реальная производительность R_{real}
 - HPL Benchmark (сбалансированная задача расчета системы линейных алгебраических уравнений)
 - Тестировали один узел $\approx 260 GFLOPS$
- Эффективность
 - $R_{real} / R_{peak} = Eff (SMH11 \approx 80\%)$

Интерфейс пользователя, базовый набор протоколов

- Учетные данные
 - Имя пользователя (george)
 - Пароль пользователя (secret)
- Типичный сценарий работы пользователя с МВС (управляющая машина):
 - Войти на управляющую машину
 - Подготовить программу, загрузить данные, запустить расчет
 - Получить номер задания (расчет)
 - Выйти
 - Снова войти на управляющую машину
 - Проверить состояние задания (выполняется, закончено). Скопировать результаты вычислений
 - Выйти.

Командный интерфейс

- Терминал
 - Для Windows: PuTTY
 - Для Linux: любой терминал (xterm, gnome-terminal, ...)
- Протокол
 - SSH (Secure Shell — сетевой протокол безопасного обмена данными)
 - Удаленный запуск команд и проброс ввода/вывода информации
 - Транспорт для других протоколов и сервисов



Файловый интерфейс

- Утилита WinSCP
 - scp — Secure Copy (команда ср поверх протокола SSH, протокол SFTP)
- Передача файлов между рабочим компьютером и управляющей машиной МВС
- Функции:
 - Создать / Удалить / Переместить / Редактировать
 - Аналогично с директориями

WinSCP Login

Session

Stored sessions

Environment

Directories

SSH

Preferences

Session

Host name:

smh11.cc.dvo.ru

Port number:

22

User name:

Password:

☐ Advanced options

About...

Language

/ - george@smh11.cc.dvo.ru - WinSCP

Local Mark Files Commands Session Options Remote Help



C:\Documents and Settings\Administrator\My Documents

Name	Ext	Size	Type	Changed	Attr
..			Parent directory	10/23/201...	r
My Music			File Folder	10/23/201...	r
My Pictures			File Folder	10/23/201...	r
Visual Studio 2008			File Folder	10/24/201...	
desktop.ini		84	Configuration S...	10/23/201...	ash

0 B of 84 B in 0 of 4

F2 Rename F4 Edit F5 Copy F6 Move F7 Create Directory F8 Delete F9 Properties F10 Quit

Default



/ <root>

Name	Ext	Size	Changed	Rights	Owner
..			10/14/2011 4:2...	rw-r--r--	root
bin			9/28/2011 12:5...	rw-r--r--	root
boot			9/30/2011 4:46...	rw-r--r--	root
dev			10/19/2011 9:2...	rw-r--r--	root
diskless			9/29/2011 6:27...	rw-r--r--	root
etc			11/8/2011 3:17...	rw-r--r--	root
home			11/8/2011 3:17...	rw-r--r--	root
lib			8/12/2011 12:5...	rw-rw-rw-	root
lib32			8/12/2011 5:00...	rw-r--r--	root
lib64			9/28/2011 12:5...	rw-r--r--	root
lost+found			8/11/2011 11:4...	rw-r--r--	root
media			8/4/2011 3:35...	rw-r--r--	root
mnt			10/5/2011 2:11...	rw-r--r--	root
opt			9/22/2011 5:08...	rw-r--r--	root
proc			10/19/2011 8:2...	r-xr-xr-x	root
root			11/9/2011 2:20...	rw-r--r--	root
sbin			10/7/2011 12:4...	rw-r--r--	root
sys			10/19/2011 8:2...	rw-r--r--	root
tmp			11/9/2011 4:10...	rw-rw-rw-	root
usr			8/12/2011 1:06...	rw-r--r--	root
var			8/19/2011 5:10...	rw-r--r--	root

0 B of 0 B in 0 of 20



SFTP-3



0:00:24

Общая схема технологической цепочки расчетов на МВС

- Подготовка программы (и сборка)
- Подготовка исходных данных
- Запуск и мониторинг задачи
- Получение результатов и их анализ

Этап 1: Подготовка программы

- Исходные коды программы
 - Набор текстовых файлов
- Компилятор
 - GCC (GNU Compiler Collection): C/C++, Fortran
- Система сборки
 - make

Linux vs. Windows

- Исходные тексты программы
 - Регистр символов:
 - Windows: не различает (`#include "MyClass.h"`)
 - Linux: различает
 - Кодировка
 - Windows: кодовая страница 1251
 - Linux: преимущественно UTF-8, устаревает KOI8-R
 - LF vs. CR+LF
 - Windows: строка заканчивается двумя байтами (0D0Ah)
 - Linux: строка заканчивается одним байтом (0Ah)
 - "Hello, World!\r\n" — "Hello, World!\n"

Полезные утилиты

- Редакторы
 - vim — мощный, но сложный
 - nano — простой, но легкий
 - mcedit — редактор (ala Far Manager)
- Текстовые конвертеры
 - iconv — перевод из одной кодировки в другую
- Интерфейсная оболочка
 - mc — аналог Far Manager

Linux vs. Windows

- Компилятор
 - Разная реализация стандарта языка C/C++
 - Разная реализация C Library и STL
 - fopen — работает одинаково
 - atoi — Windows + Linux, itoa — Windows
 - Разрядность: 32 бита и 64 бита
 - long — разный размер
 - void* — разный размер

Система сборки GNU Make

- Программа состоит из:
 - Исходных текстов (*.c/*.cpp)
 - Заголовочных файлов (*.h)
- Система сборки — по исходным текстам сформировать исполняемое представление
 - Стадия компиляции — получаем объектный код
 - Стадия связывания (Linking) — полученный объектный код преобразуется в исполняемое представление

Структура Makefile

- **Выражения:**
Переменная = Значение
- **Правила:**
Цель : Зависимости (подстановка переменных)
Действия для достижения цели
(подстановка переменных)


```
INCdep = $(INCdir)/hpl_misc.h $(INCdir)/hpl_blas.h $(INCdir)/hpl_auxil.h
HPL_au0obj = HPL_dlacpy.o HPL_dlatcpy.o HPL_fprintf.o HPL_warn.o HPL_abort.o
HPL_dlaprnt.o HPL_dlange.o
HPL_au1obj = HPL_dlamch.o
HPL_auxobj = $(HPL_au0obj) $(HPL_au1obj)
```

```
all      : lib
lib       : lib.grd
lib.grd  : $(HPL_auxobj)
           $(ARCHIVER) $(ARFLAGS) $(HPLlib) $(HPL_auxobj)
           $(RANLIB) $(HPLlib)
           $(TOUCH) lib.grd
```

```
HPL_dlacpy.o      : ../HPL_dlacpy.c $(INCdep)
                  $(CC) -o $@ -c $(CCFLAGS) ../HPL_dlacpy.c
HPL_dlatcpy.o     : ../HPL_dlatcpy.c $(INCdep)
                  $(CC) -o $@ -c $(CCFLAGS) ../HPL_dlatcpy.c
HPL_fprintf.o     : ../HPL_fprintf.c $(INCdep)
```