Кластерные системы

Судаков А.А.

“Параллельные и распределенные вычисления” Лекция 11

План

- История
- Кластеры типа Beowulf
- Кластеры типа MOSIX
- Кластеры типа SSI
- Балансирующие кластеры
- Высоконадежные кластеры
- Виртуальные машины

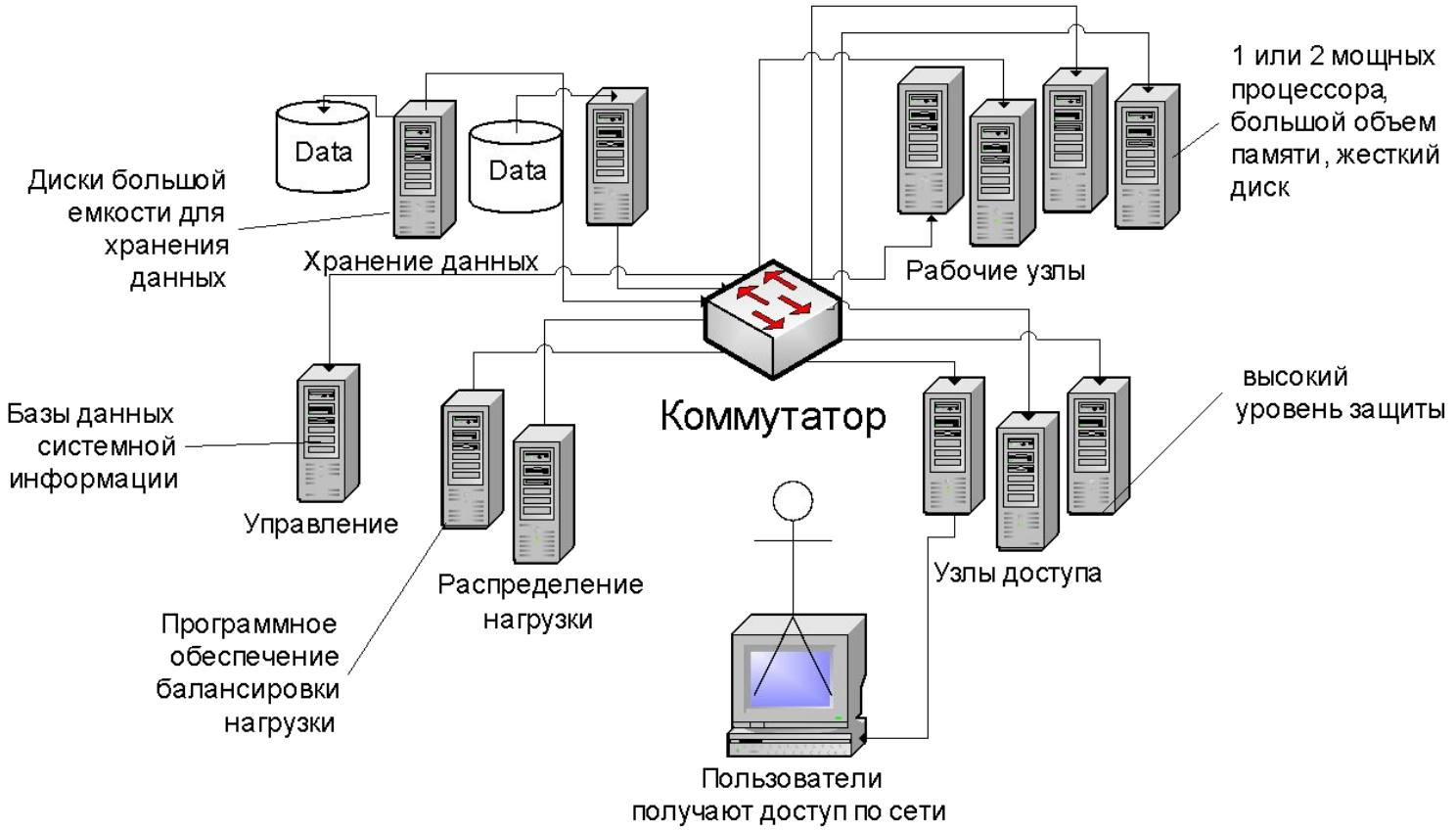
Литература

- http://www.clusterresources.com
- http://www.mosix.org
- http://www.openmosix.org
- http://bproc.sourceforge.net/
- http://www.linuxvirtualserver.org/

Кластеры типа Beowulf

- Типа Beowulf
  - Компьютеры широкого использования
  - Распределенный образ операционной системы
  - Централизованная модель
  - Гетерогенность
  - Возможные расширения
    - Общая память
    - Миграция процессов
    - Чекпоинт/рестарт

Схема


Data
Data
Диски большой емкости для хранения данных
Хранение данных
1 или 2 мощных процессора, большой объем памяти, жесткий диск
Рабочие узлы
Коммутатор
Базы данных системной информации
Управление
Распределение нагрузки
Программное обеспечение балансировки нагрузки
Узлы доступа
высокий уровень защиты
Пользователи получают доступ по сети

Узлы кластера

- *рабочие узлы* (worker node)
  - выполнение рассчета
- *хранения данных* (storage node)
  - хранение доступных данныех
- *узлы управления* (management node)
  - программное обеспечение для администрирования системы
  - базы данных системной информации (NIS master, LDAP master)
- *узлы доступа* (login node)
  - вход пользователей из Интернет
- *узлы распределения нагрузки* (workload management node)
  - Сервер и планировщик системы управления нагрузкой
- Коммуникации
  - Сеть быстрого обмена данными
  - Сеть мониторинга

Работа узлов в кластере

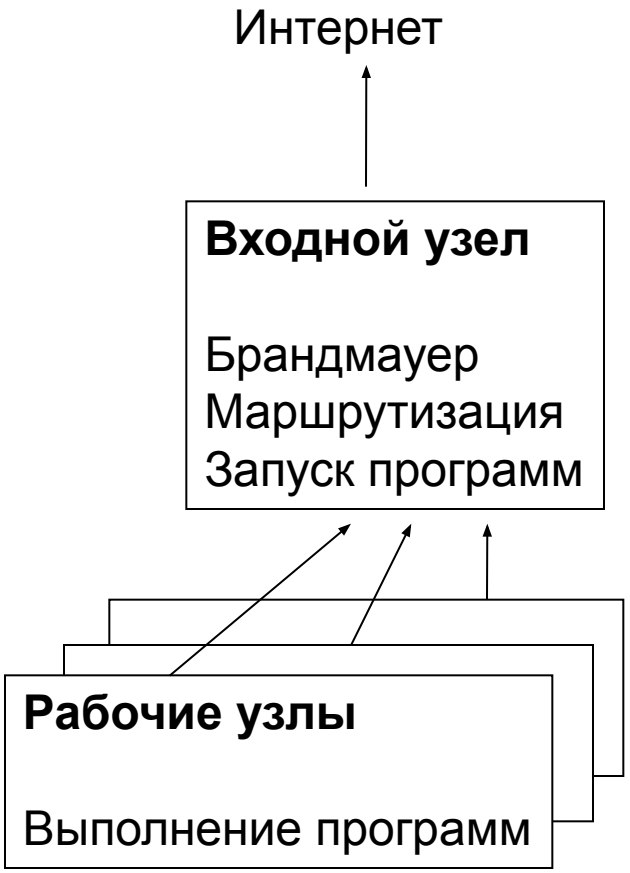
- Работа кластера как одной системы определяется программным обеспечением
- Узлы кластера должны "доверять" друг другу
  - возможность запускать программы на разных узлах кластера без ввода пароля
  - Файлы данных должны быть доступны всем узлам
- Модель программирования
  - Обмен сообщениями (MPI, PVM)
  - Общая память
    - На многопроцессорных узлах
    - При наличии соответствующей сети (SCI, QSNet)
  - Комбинированная

Запуск программ на кластере

- Для запуска на любом узле кластера
  - Ssh, rsh
  - Агенты системы распределения нагрузки
- Запуск параллельных программ
  - Ssh или rsh для запуска соответствующего процесса на удаленной машине

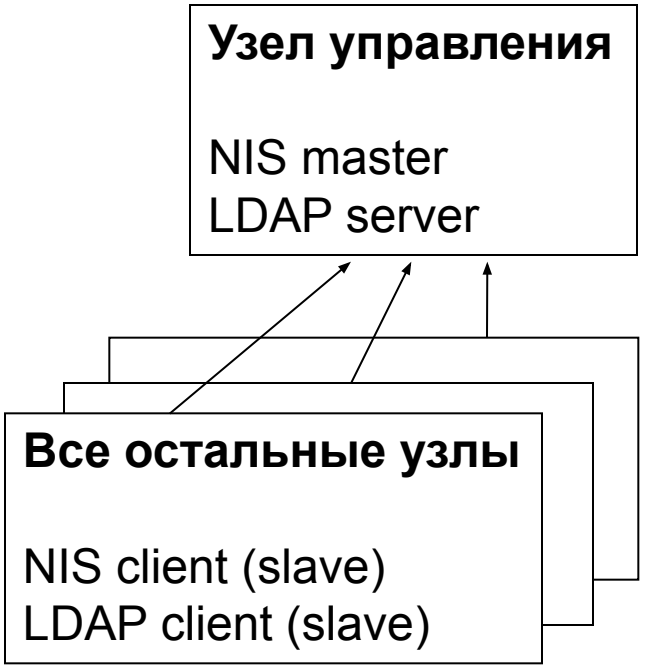
Входные узлы

- Входной узел
  - Брандмауэр
  - Маршрутизатор для узлов кластера
  - Кэширующий DNS
  - Запуск программ на рабочих узлах
  - Интерфейс системы мониторинга
- Часто пользователи не имеют прямого доступа ни на какие узлы, кроме входных



Базы данных системной информации

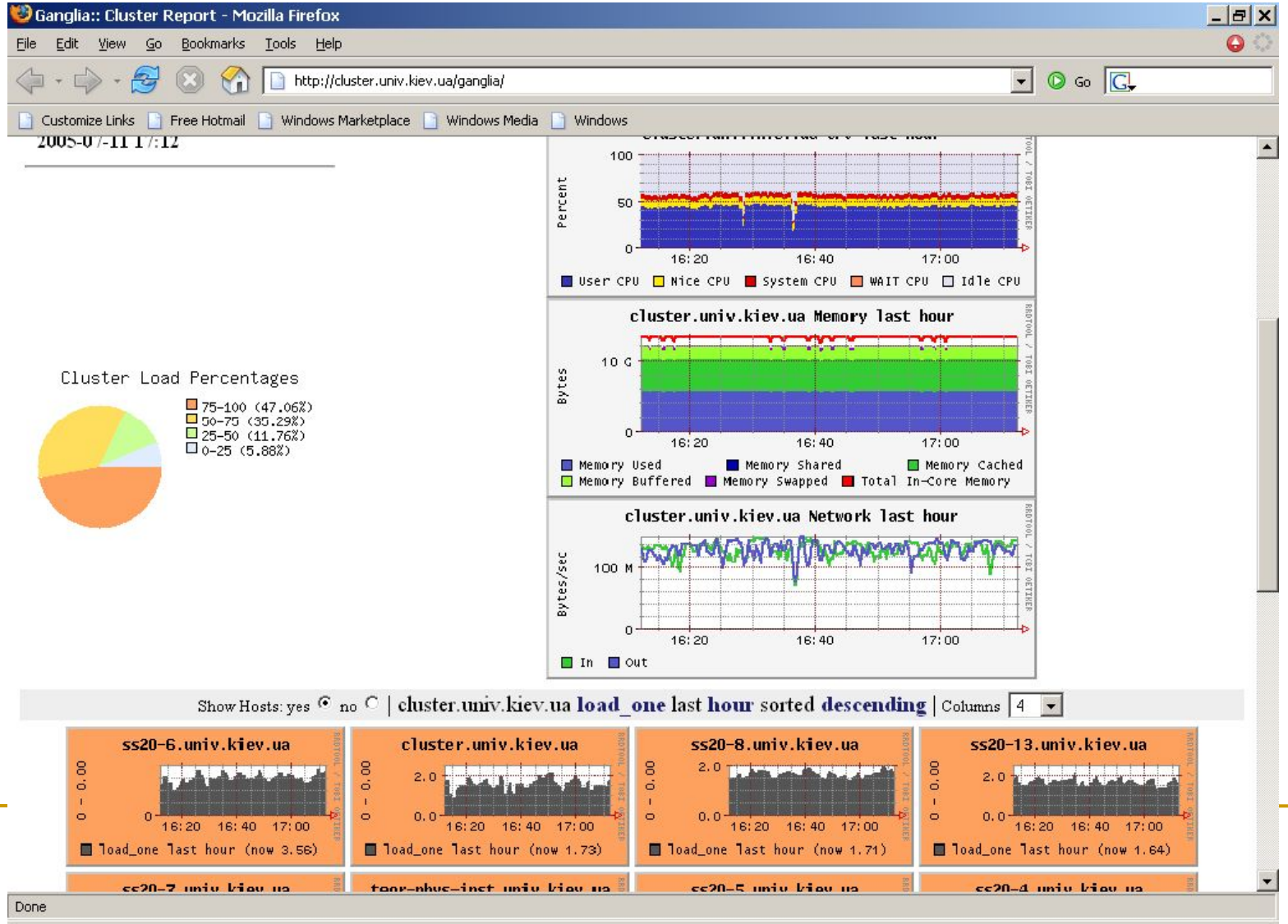
- Узел управления содержит авторитетные копии баз данных системной информации
  - Список пользователей
  - Список групп
- Новая информация добавляется и изменяется только на узле управления
- Остальные узлы
  - Обращаются к базе данных при необходимости
  - Могут содержать реплики главной базы данных
- При любом акте авторизации или аутентификации
  - Обращения идут к к главному серверу
  - Может выполняться кэширование на локальных узлах (NSCD)



Система мониторинга

- Рабочие узлы
  - Агенты сбора информации
- Узел коллектора (входной или управления)
  - Агрегация и анализ информации
- Узел управления
  - Система включения-выключения узлов

GANGLIA


Ganglia:: Cluster Report - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://cluster.univ.kiev.ua/ganglia/
Customize Links Free Hotmail Windows Marketplace Windows Media Windows
Cluster Load Percentages
75-100 (47.06%)
50-75 (35.29%)
25-50 (11.76%)
0-25 (5.88%)
Percent
User CPU Nice CPU System CPU WAIT CPU Idle CPU
cluster.univ.kiev.ua Memory last hour
Bytes
Memory Used Memory Shared Memory Cached
Memory Buffered Memory Swapped Total In-Core Memory
cluster.univ.kiev.ua Network last hour
Bytes/sec
In Out
Show Hosts: yes no | cluster.univ.kiev.ua load_one last hour sorted descending | Columns 4
ss20-6.univ.kiev.ua
load_one last hour (now 3.56)
cluster.univ.kiev.ua
load_one last hour (now 1.73)
ss20-8.univ.kiev.ua
load_one last hour (now 1.71)
ss20-13.univ.kiev.ua
load_one last hour (now 1.64)
Done

Система распределения нагрузки

- Задачи
  - Максимально эффективное использование ресурсов кластера
  - Максимальная скорость вычислений
  - Удовлетворение требований пользователей в необходимых ресурсах
    - Память
    - Дисковое место
    - Специальные ресурсы (стример)

Повышение эффективности использования ресурсов

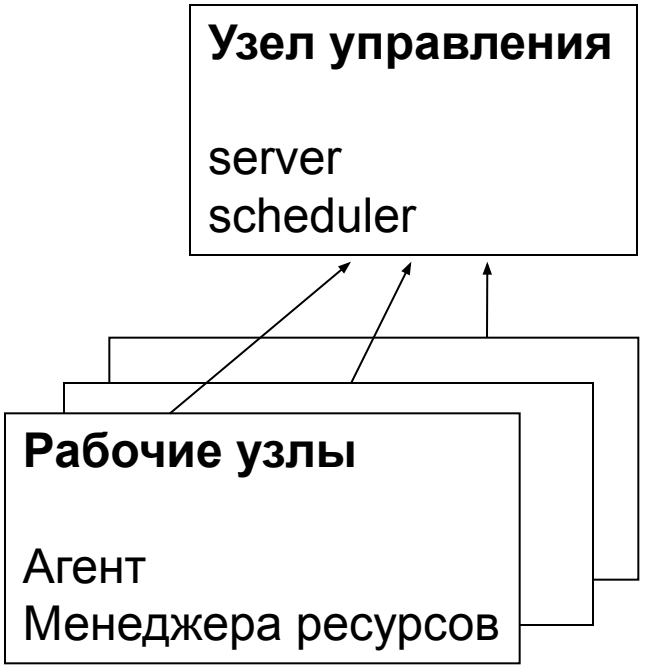
- Пакетный режим
  - Пользователи указывают необходимые ресурсы
  - Пользователи ставят свои задачи в очередь
  - Система выполняет задачи в очереди в порядке приоритетности
- Распределение нагрузки
  - Запуск программ оптимальным (в плане скорости) образом
- Контроль использования ресурсов
  - При превышении лимитов задача завершается принудительно

Реализации систем пакетного режима

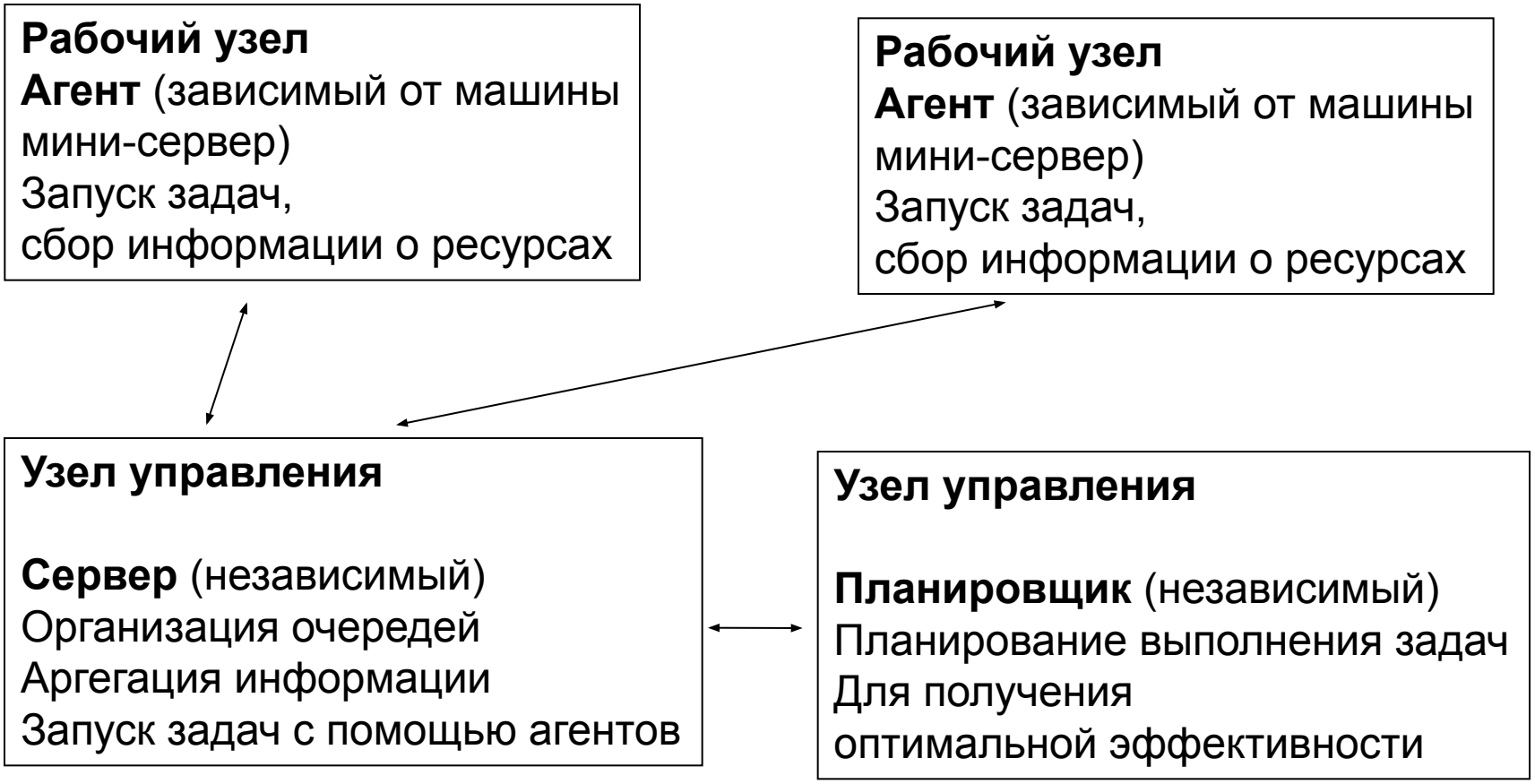
- NQS – network queuing system
- PBS – portable batch system
- torque
- LL – load leveler
- BQS - Batch Queing System
- CONDOR
- LSF

PBS

- Агент менеджера ресурсов (pbs_mom) – machine oriented miniserver
  - Сбор информации о загруженности своего узла
  - Запуск программ на своем узле
  - Остановка задач на своем узле
- Сервер (pbs_server)
  - Агрегация информации от агентов
  - Поддержка очередей
  - Запуск задач посредством агентов на заданных серверах
- Планировщик
  - Выдача информации серверу об оптимальном выборе ресурсов для разных задач



Структурная схема PBS



Агент менеджера ресурсов

- Клиент-серверная идеология
  - Может подсоединяться к серверу и планировщику для передачи информации
  - Слушает порт с возможностью подсоединения
- Периодически передает серверу информацию о загруженности и занятых ресурсах
- Выполняет указания сервера и планировщика по запуску и остановке задач

```
 813 ?        S     29:34 /usr/local/torque/sbin/pbs_mom
11866 ?        S      0:00  \_ -bash
11867 ?        S      0:01      \_ pbs_demux
11920 ?        S      0:00      \_ /bin/bash -x /var/spool/torque/mom_priv/jobs/8097.cluste.SC
11940 ?        S      0:00          \_ mpirun -np 4 /usr/local/gromacs/i686-pc-linux-gnu/bin/mdrun_mpi
```

Информация о ресурсах

s6
    state = free
    np = 6
    ntype = cluster
    jobs = 0/8054.cluster.univ.kiev.ua, 1/8054.cluster.univ.kiev.ua, 2/8092.cluster.univ.kiev.ua, 3/8092.cluster.univ.kiev.ua
    status = arch=linux,uname=Linux ss20-6.univ.kiev.ua 2.4.29 #4 SMP Sat Mar 12 18:51:26 EET 2005 i686,sessions=11486 11485 13077 13179
        13180,nsessions=5,nusers=1,idletime=75517,totmem=3137424kb,availmem=2764512kb,physmem=1032920kb,ncpus=4,loadave=2.81,gres=sse2:1+old:1+sse:1+ia32:1,netload=4091552988,size=12243388kb:16513960kb,state=free,rectime=1121093394

s16
    state = free
    np = 6
    ntype = cluster
    jobs = 0/8106.cluster.univ.kiev.ua, 1/8105.cluster.univ.kiev.ua
    status = arch=linux,uname=Linux ss20-16.univ.kiev.ua 2.6.12 #2 SMP Sat Jun 25 11:53:19 EEST 2005 x86_64,sessions=2057 27304 27314 27315 14563
        25193,nsessions=6,nusers=3,idletime=257369,totmem=4153400kb,availmem=3755768kb,physmem=2056928kb,ncpus=4,loadave=1.90,gres=new:1+ia32e:1+x86_64:1+sse2:1+sse3:1+sse2:1+sse:1+ia32:1,netload=344160256,size=18281492kb:30254032kb,state=free,rectime=1121093374

s17
    state = free
    np = 6
    ntype = cluster
    jobs = 0/8054.cluster.univ.kiev.ua, 1/8054.cluster.univ.kiev.ua, 2/8092.cluster.univ.kiev.ua, 3/8092.cluster.univ.kiev.ua
    status = arch=linux,uname=Linux ss20-17.univ.kiev.ua 2.4.29 #3 SMP Wed Feb 23 12:42:34 EET 2005 i686,sessions=14044 14043 17177 17178,nsessions=4,nusers=1,idletime=80182,totmem=5193984kb,availmem=4895448kb,physmem=1032504kb,ncpus=4,loadave=2.15,gres=sse2:1+old:1+sse:1+ia32:1,netload=1580948483,size=1097220kb:4128448kb,state=free,rectime=1121093374

Сервер

- Организация очередей
- Агрегация использования ресурсов
- Очередь (класс)
  - В каждую очередь попадают задачи с определенными требованиями по ресурсам
  - Задачи с одинаковыми требованиями выполняются последовательно
- Сервер указывает агентам, что запускать

Очереди

```
#
# Create and define queue mono_long
#
create queue mono_long
set queue mono_long queue_type = Execution
set queue mono_long Priority = 4
set queue mono_long max_running = 23
set queue mono_long resources_max.nodect = 1
set queue mono_long resources_min.walltime = 36:00:00
set queue mono_long resources_default.walltime = 72:00:00
set queue mono_long max_user_run = 10
set queue mono_long enabled = True
set queue mono_long started = True
#
# Create and define queue stereo_short
#
create queue stereo_short
set queue stereo_short queue_type = Execution
set queue stereo_short Priority = 4
set queue stereo_short resources_max.walltime = 64:00:00
set queue stereo_short resources_min.nodect = 2
set queue stereo_short max_user_run = 4
set queue stereo_short enabled = True
set queue stereo_short started = True
#
# Create and define queue stereo_long
#
```

```
server: cluster.univ.kiev.ua

Queue            Memory CPU Time Walltime Node Run Que Lm  State
---------------- ------ -------- -------- ---- --- --- --  -----
mono_short         --      --    36:00:00    1   0   0 --   E R
default            --      --       --      --   0   0 --   E R
mono_long          --      --       --       1   4   0 23   E R
stereo_short       --      --    64:00:00  --    0   0 --   E R
stereo_long        --      --       --      --   3   1 --   E R
alien              --      --       --      --   0   0 --   E R
                                                --- ---
                                                  7   1
```

Планировщик

- На основе информации
  - менеджера ресурсов
  - Требований пользователя
  - Ограничений
- Запланировать выполнение всех задач с максимальной эффективностью

Реализации планировщиков

- PBS_SCHEDULER
  - Входит в систему PBS
- MAUI
  - Самый распространенный кластерный планировщик
  - Организация качества обслуживания
    - Приоритеты
    - Вытеснение
    - Назначение ограничений

Сценарий запуска (паспорт задачи)

- Задача может быть запущена на любом узле кластера
- Чтобы все запустилось правильно пользователь должен указать как запускать задачу
  - Имя и путь к программе
  - Необходимые ресурсы
- PBS предоставляет пользователю переменные среды
  - Имена машин на которых были запущены задачи
  - Имена машин и каталог из которых бала запущена задача
  - …

Пример запуска

- Сценарий

```
#PBS
cat $PBS_NODEFILE
sleep 20
```

- Запуск

```
qsub -lnodes=2:ppn=2 tst
8115.cluster.univ.kiev.ua
```

- Выполнение

[saa@cluster pbs]$ qstat

```
Job id           Name             User             Time Use S Queue
---------------- ---------------- ---------------- -------- - -----
8054.cluster     dopc_ann_4       yesint           00:01:29 R stereo_long
8092.cluster     dopc_f2n8_ann    yesint           00:00:55 R stereo_long
8097.cluster     eEF1A2_n5        kanibolotsky     00:00:01 R stereo_long
8102.cluster     eEF1A2_n10       kanibolotsky            0 Q stereo_long
8103.cluster     ...01_restart-07 platon           47:06:22 R mono_long
8104.cluster     ...02_restart-07 platon           47:30:02 R mono_long
8105.cluster     ...04_restart-06 platon           47:19:13 R mono_long
8106.cluster     ...03_restart-06 platon           22:02:52 R mono_long
8115.cluster     tst              saa              00:00:00 R stereo_short
```

- Результат

s16
s16
s15
s15

Особенности кластера типа beowulf

- Преимущества
  - Гетерогенность – не помеха
  - Независимость от программного и аппаратного обеспечения
  - Простота организации
  - Высокая надежность
  - Высокая масштабируемость
  - Низкое соотношение цена/производительность
- Недостатки
  - Статическое распределение нагрузки
  - Необходимость сложных настроек для получения оптимальной производительности

bproc

- Общее пространство процессов для beowulf кластеров (SSI)
- Процессы выполняются на удаленных узлах и могут мигрировать между узлами, хотя видятся как запущенные на входном узле
- Миграция путем полной копии ресурсов (виртуальной памяти, открытых файлов)

Кластер типа MOSIX

- MOSIX – Multicomputer Operating System for Unix
- Кластер с одной копией операционной системы
- Динамическая балансировка нагрузки путем миграции процессов с вытеснением

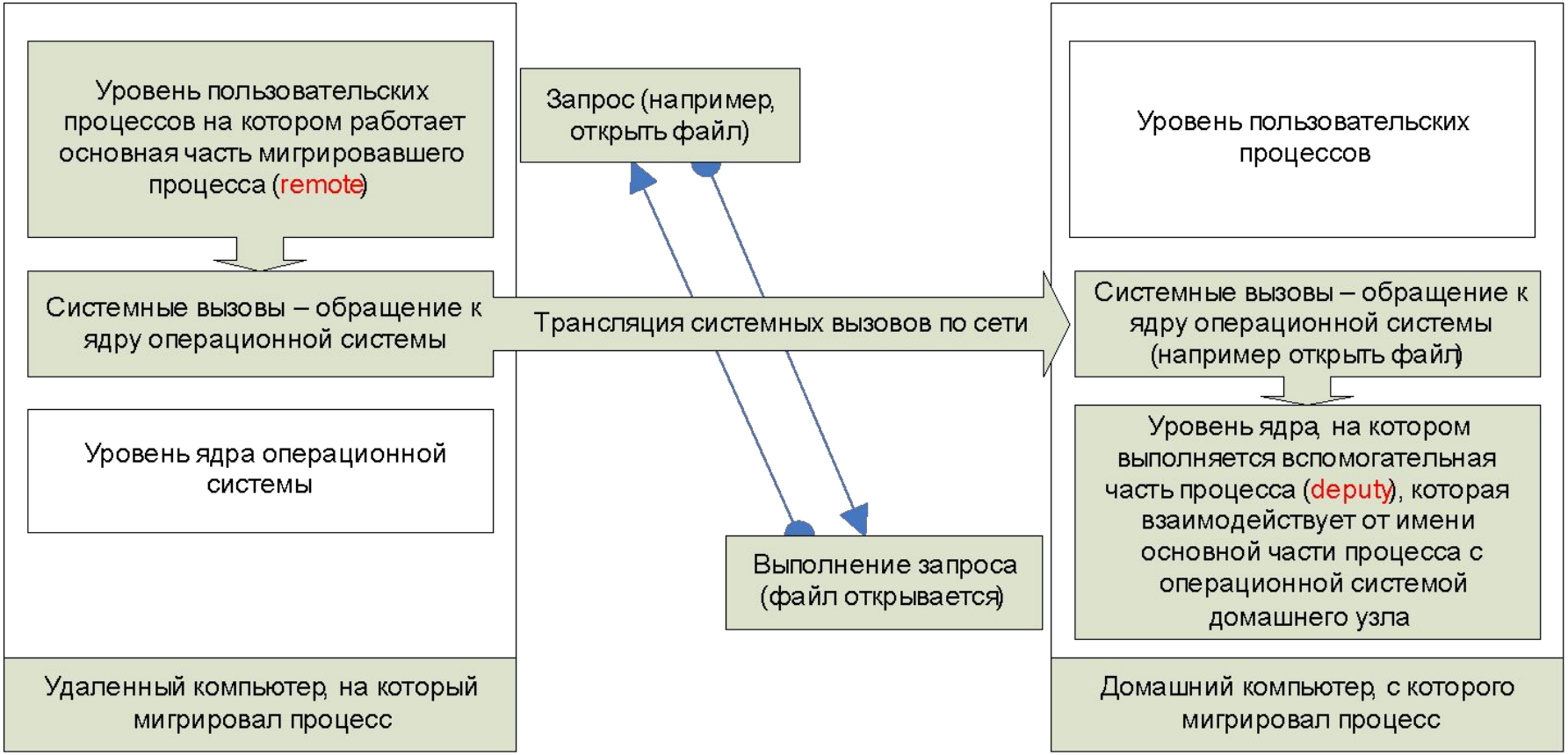
Компоненты

- Децентрализованная схема
  - Каждый узел имеет одинаковую структуру и все узлы равноправны
- Каждый узел имеет
  - Информационный агент
    - Передача информации об использовании ресурсов своего узла
    - Прием информации о ресурсах других узлов
  - Агент миграции
    - Принимает запросы от других узлов на миграцию процессов с них
    - Запускает мигрировавшие процессы
  - Планировщик
    - Принимает решение о миграции процессов со своего узла

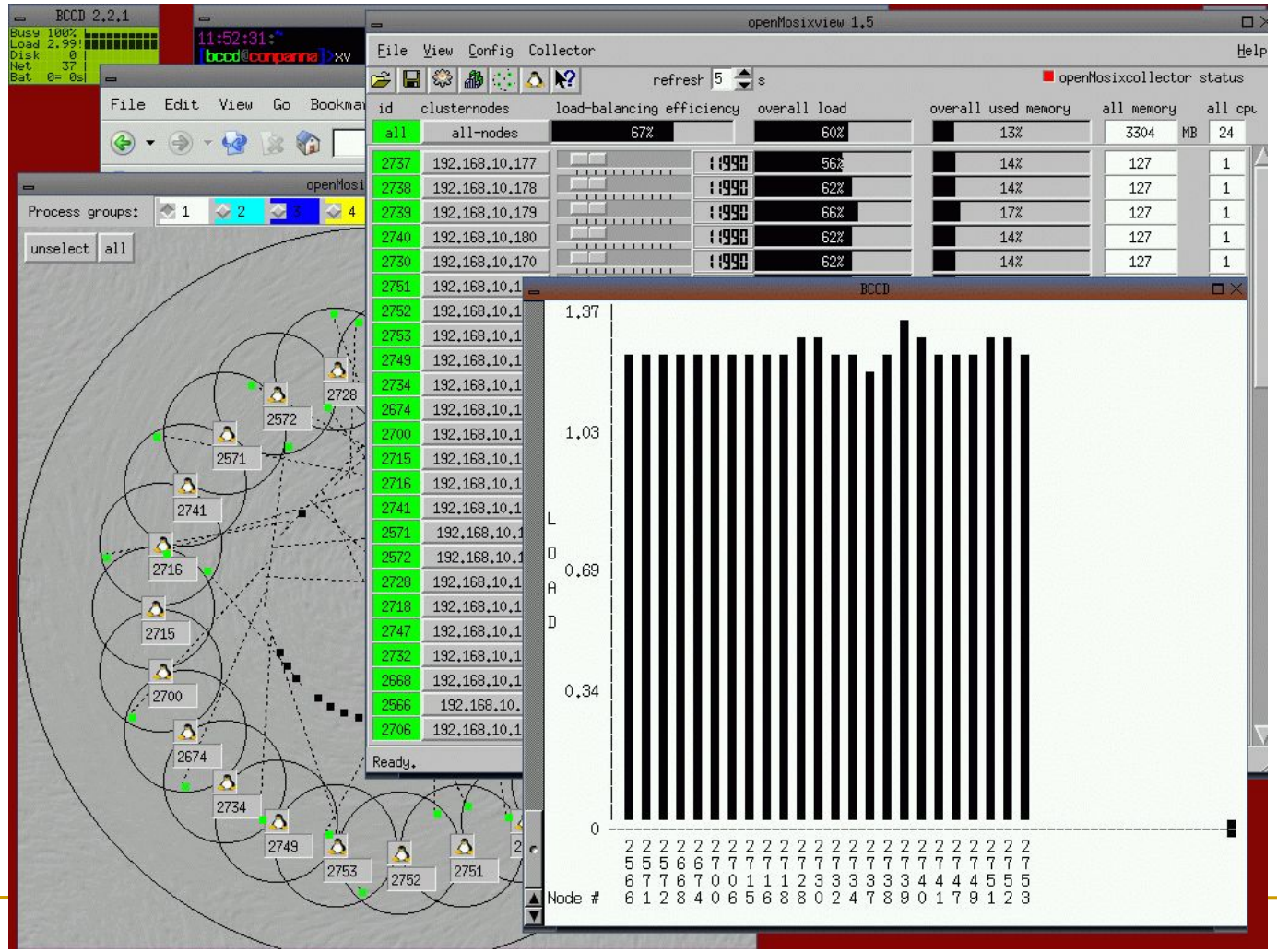
Концепция домашнего узла

- Каждый процесс остается связанным с той машиной, на которой он был запущен
- При миграции процесс разбивается на две части
  - Deputy – режим ядра, связанный с домашним узлом
  - Remote – режим ядра и режим задачи, связанный с удаленным узлом
- При выполнении системного вызова удаленной частью процесса, вызов транслируется на домашний узел по сети

Выполнение системного вызова



Интерфейс


BCCD 2.2.1
openMosixview 1.5
File View Config Collector Help
openMosixcollector status
id clusternodes load-balancing efficiency overall load overall used memory all memory all cpu
all all-nodes 67% 60% 13% 3304 MB 24
2737 192.168.10.177 56% 14% 127 1
2738 192.168.10.178 62% 14% 127 1
2739 192.168.10.179 66% 17% 127 1
2740 192.168.10.180 62% 14% 127 1
2730 192.168.10.170 62% 14% 127 1
Ready.
Process groups:
unselect all
BCCD
1.37
1.03
0.69
0.34
0
LOAD
Node #
2572
2571
2741
2716
2715
2700
2674
2734
2749
2753
2752
2751
2728

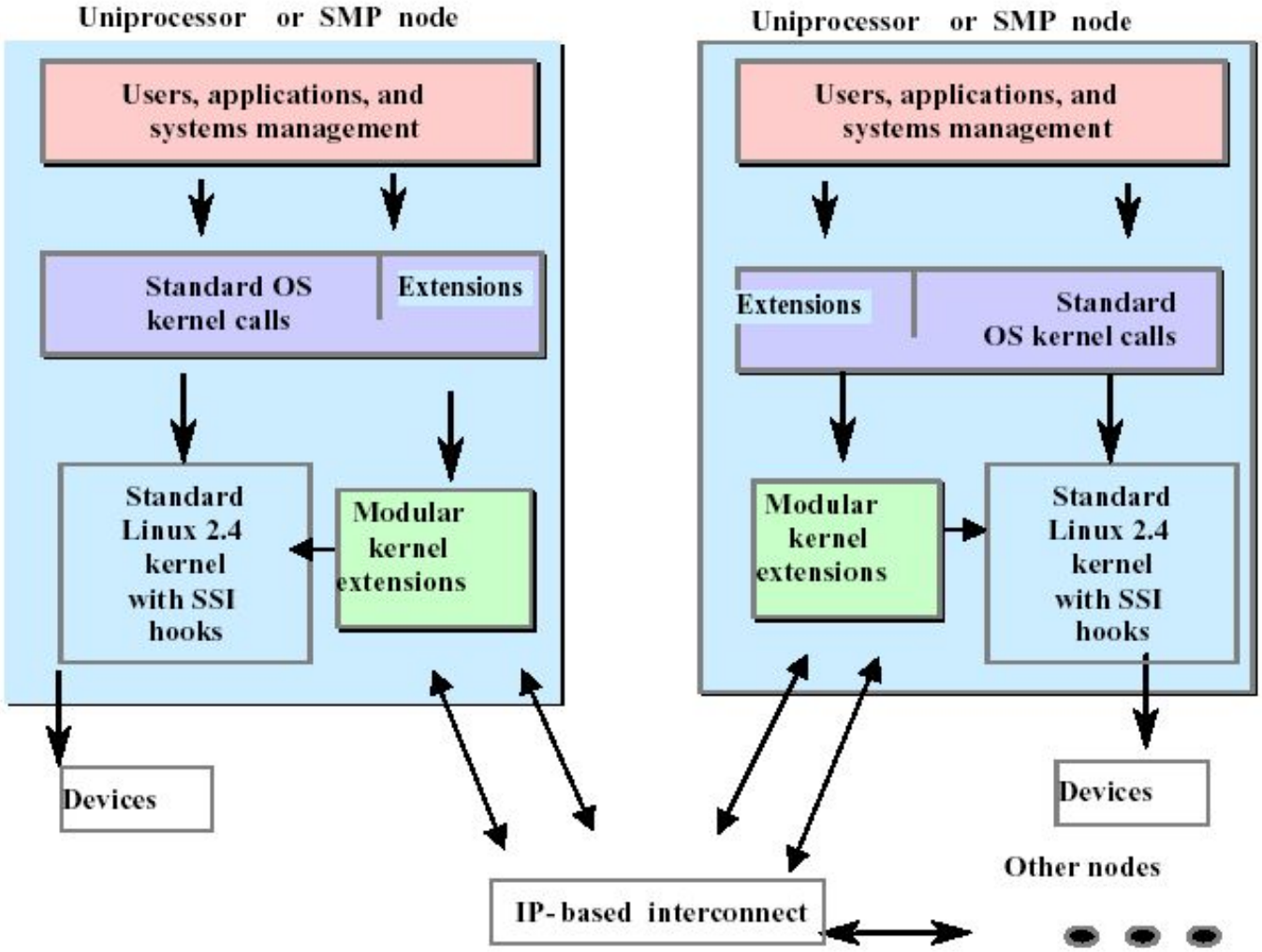
Особенности

- Преимущества
  - Эффективен для гетерогенных кластеров
  - Не требует специфических настроек
  - Динамическая масштабируемость
  - Возможность использования существующих ресурсов для создания метакластеров
- Недостатки
  - Требует однотипного (не сильно отличающегося) аппаратного и программного обеспечения
  - Не удовлетворяет требованиям стабильности и безопасности
  - Концепция домашнего узла – ограничивает возможности

SSI Linux

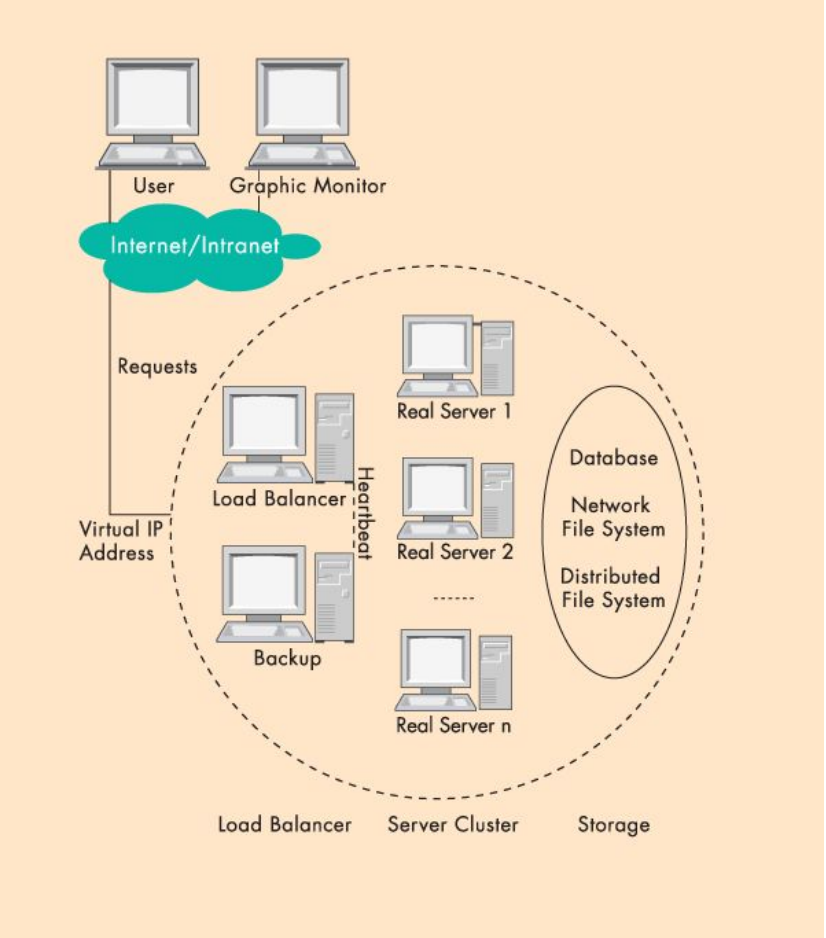
- Полнофункциональный кластер с общим образом операционной системы
  - Общее пространство процессов
  - Общая память
  - Общая файловая система
  - Общие средтства коммуникации
- Интегрирует в себе многие другие проекты

Структурная схема



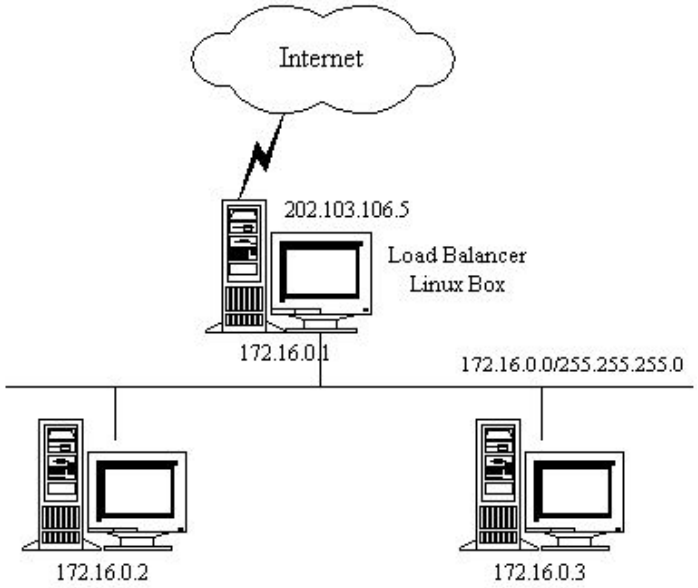
Балансирующие кластеры

- Кластер серверов видится как одна машина
- Внутри запросы к общему адресу распределяются между серверами, входящими в кластер



Linux Virtual Server

- NAT
- Direct route
- IP tunnel



Высоконадежные кластеры

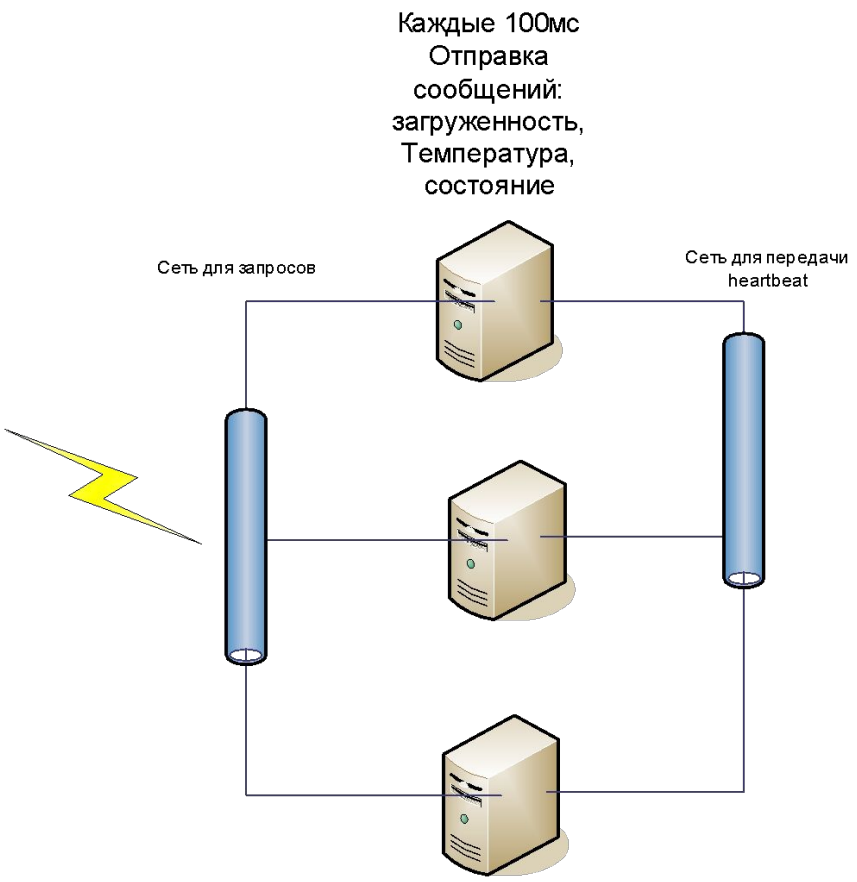
- Отказоустойчивость (fault-tolerance)
  - При возникновении сбоев может сам восстановиться
- Высокая готовность (high availability)
  - При обнаружении ошибки быстро готов к выполнению работы

Условия обеспечения высокой надежности

- Обнаружение отказов
- Избыточность – высокая готовность
  - Запасные компоненты, готовые сразу же включиться в работу
- Журналирование (транзакции) - отказоустойчисвость
  - Сохранение промежуточных действий с возможностью вернуться к последней успешной операции
- Механизм устранения неисправных компонент

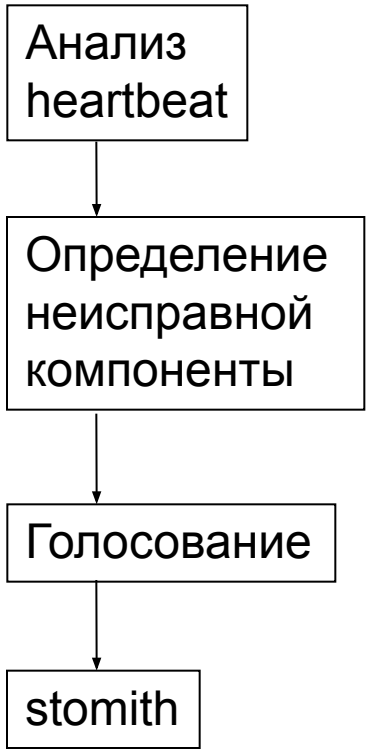
Обнаружение отказов

- Keep-alive, hello, heartbeat
  - Периодически отправляемая информация о состоянии каждого узла (компоненты)
- Если сообщения от компоненты не поступают или поступают с неверными параметрами, то фиксируется отказ компоненты



Устранение неисправной компоненты

- Stomish
  - Shoot Other Machine In The Head
  - Задача – быстро устранить неисправную машину
  - Метод – выключение питания с помощью управляемого выключателя питания
  - Исполнитель – одна из машин кластера
    - Временный координатор



Избыточность

- Избыточность данных
  - Зеркалирование – создание полной копии
  - Репликация – восстановление из копии
  - Multipass – обеспечение нескольких путей к данным
- Избыточность функций
  - Дублирование – несколько серверов, процессов, сетевых адаптеров и др. устройств с одинаковыми функциями

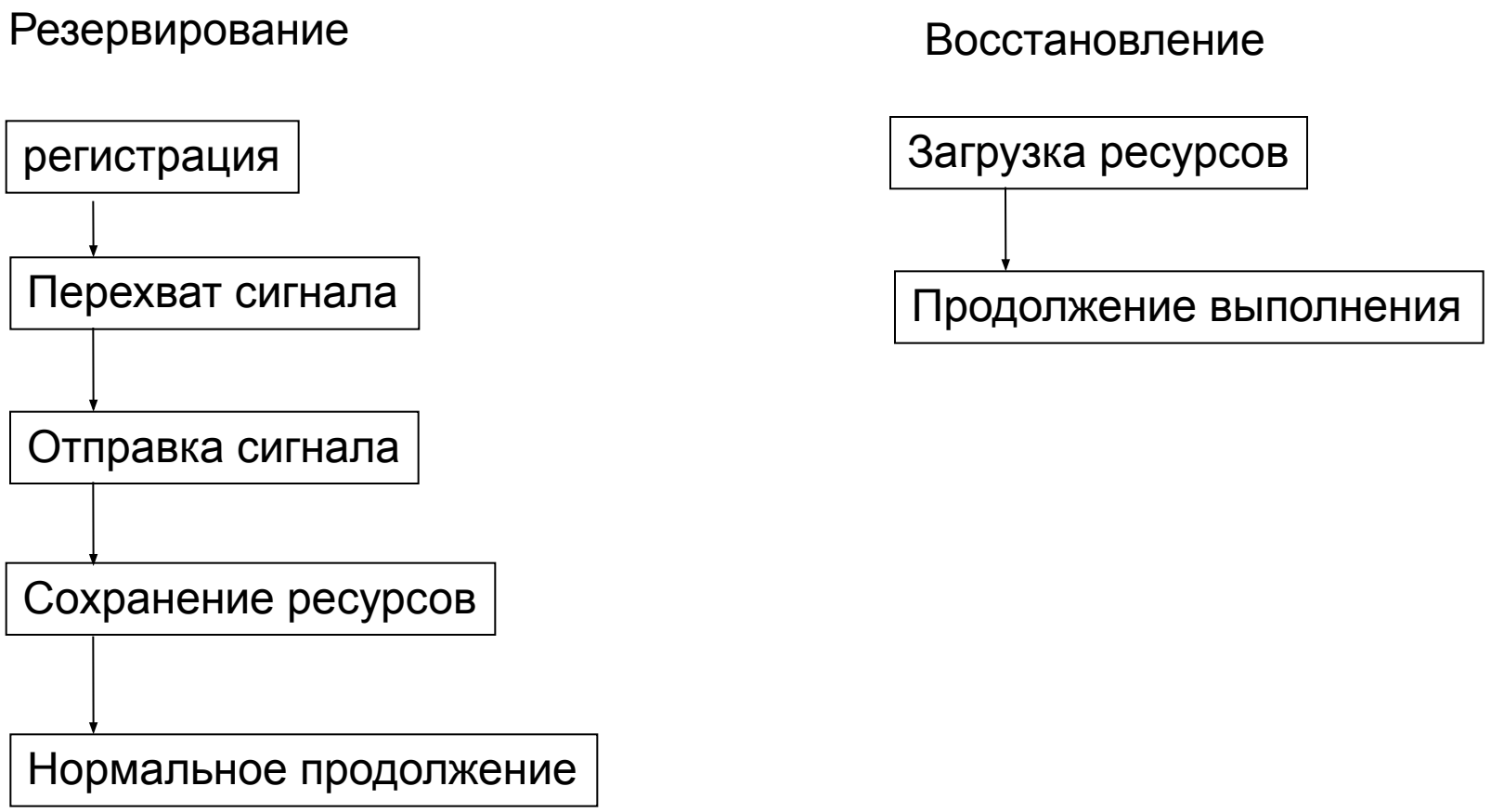
Журналирование

- Журналирование данных
  - Перед тем, как записывать данные на постоянное место, они записываются во временное хранилище - журнал
  - После этого данные записываются на постоянное место
- Транзакции
  - Несколько последовательных операций выполняются как одна атомарная операция
- Checkpoint/restart
  - Создается копия структур данных процесса

CHPOX – CHeckPOinting linuX

- Сохранение текущего состояния процесса в дисковый файл
- Восстановление процессов из файла в точке, где они были записаны
- Процессы после восстановления продолжают выполняться

Схема работы



Пример работы

```
$ cat test.sh
#! /bin/sh
I=1
while true;
do
        echo $I
        I=$(($I+1))
        sleep 5
done

$ ./test.sh
1
2
3
```

```
$ ps xo pid,cmd|grep test
  780 /bin/sh ./test.sh
  788 grep test
$ chpoxctl add 780 40 9 /t
mp/test.dump
$ kill -40 780
$ ld-chpox /tmp/test.dump
14
15
16
17
18
19
20
21
```

Виртуальные машины

- Эмуляция аппаратного обеспечения компьютера с помощью программных средств
- Создание иллюзии того, что операционная система выполняется на аппаратном обеспечении

Примеры виртуальных машин

- Xen – виртуализация ресурсов компьютера
- Qemu – эмулятор аппаратного обеспечения
- VMWare – эмулятор аппаратного обеспечения
- UML – user mode Linux

Вопросы