Распределенные службы имен

Судаков А.А.

“Параллельные и распределенные вычисления” Лекция 7

План

- Назначение служб имен
- Основные подходы к построению
- DNS – Система Интернет имен
- NIS – Система совместного использования файлов конфигурации
- LDAP – Служба каталогов
- NetBIOS – Система имен в локальных сетях
- Jini – Система имен Интернет-служб
- Переключатель служб имен
- Авторизация и аутентификация

Литература

- DNS RFCs http://www.dns.net/dnsrd/rfc/
- http://www.tldp.org/HOWTO/NIS-HOWTO/
- RFC1487 LDAP
- Directory software http://www.padl.com/
- NETBIOS RFC http://www.faqs.org/rfcs/rfc1001.html

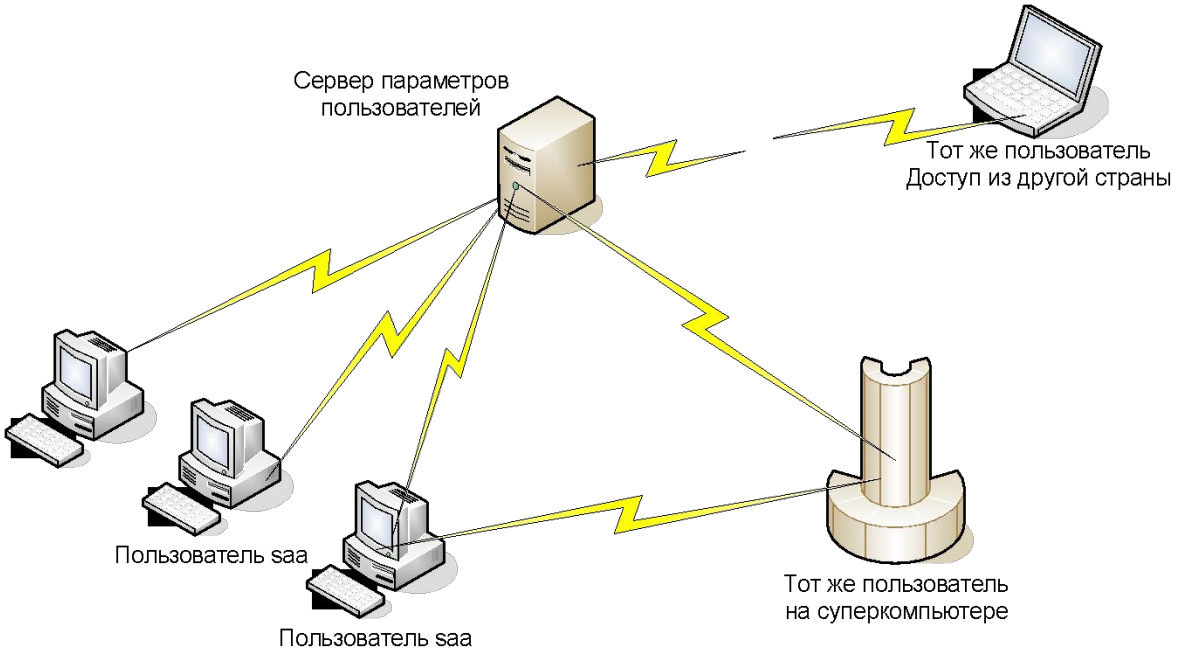
Назначение служб имен

- Нахождение объектов по некоторым их параметрам
  - Нахождение всех параметров пользователя по их псевдонимам (nick, login)
  - Нахождение IP адресов машин по их именам
  - Нахождение портов RPC служб по номеру RPC программы
  - Нахождение CORBA серверов по их IOR или именам интерфейсов

Службы имен

- Служба имен – база данных информации, представленной в виде соответствия `имя<->значение`, которая позволят выполнять поиск значений, соответствующих заданному имени
- В любой операционной системе присутствует множество баз данных разной информации
  - База данных пользователей (/etc/passwd)
  - База данных групп (/etc/group)
  - База данных служб (/etc/services)
- Для создания распределенных приложений и систем необходима распределенные службы имен, те которые могут быть доступных из разных пространств адресов

Распределенные службы имен



- Преимущества
  - Централизация управления – вся информация добавляется и изменяется в одном месте
  - Обеспечение надежности – может существовать несколько копий системы

Основные подходы к построению

- Клиент – серверная модель
  - Создается специальный выделенный сервер, который хранит базу данных и обслуживает запросы
  - К серверу подключаются клиенты, которые хотят выполнить поиск
- Централизованная модель
  - Серверу административно выделяется фиксированный адрес и порт транспортного протокола
  - Эти же параметры конфигурируются на клиентах
- Децентрализованная модель
  - Все клиенты и серверы являются равноправными
  - Одна из машин автоматически выбирается координатором (master)
  - Все остальные автоматически выполняют поиск координатора и пользуются его услугами
  - Сервера регистрируются на координаторе, а клиенты обращаются к координатору для поиска

Служба доменных имен DNS

- Преобразование Интернет имен машин в IP адреса и наоборот
  - Имя - > IP прямое преобразование
  - IP -> имя обратное преобразование
- Для чего
  - Людям удобнее работать с именами, а не с IP адресами
  - Позволяет обеспечить дополнительную безопасность путем дополнительно жесткой привязки имени к IP адресу, чтобы не могли просто так поменять

Интернет имена

- Имя (FQDN - Fully Qualified DNS Name)
  - cluster.univ.kiev.ua
- Классификация
  - Географическая информация
  - Виды деятельности
  - …

Исторические сведения

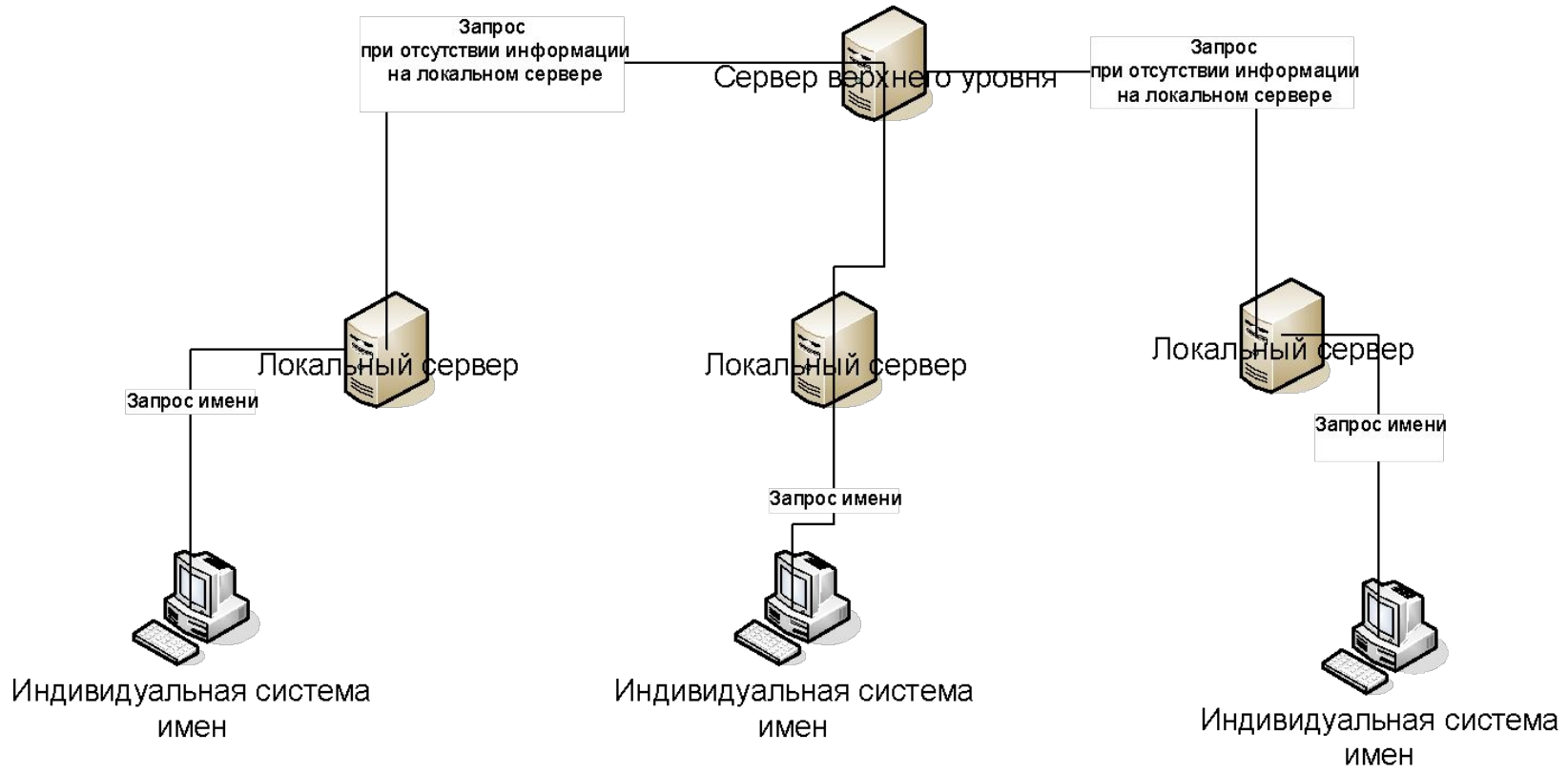
- Во времена ARPAnet все имена присваивались InterNIC и содержались в файле hosts, который при добавлении новой машины высылался всем администраторам сети
- Сейчас это файл есть на всех машинах

```
127.0.0.1       localhost.localdomain localhost
10.25.5.1       ss20-1.univ.kiev.ua     s1      node1   ss20-1
10.25.5.2       ss20-2.univ.kiev.ua     s2      node2   ss20-2
10.25.5.3       ss20-3.univ.kiev.ua     s3      node3   ss20-3
10.25.5.4       ss20-4.univ.kiev.ua     s4      node4   ss20-4
10.25.5.5       ss20-5.univ.kiev.ua     s5      node5   ss20-5
10.25.5.6       ss20-6.univ.kiev.ua     s6      node6   ss20-6
10.25.5.7       ss20-7.univ.kiev.ua     s7      node7   ss20-7
10.25.5.8       ss20-8.univ.kiev.ua     s8      node8   ss20-8
10.25.5.9       ss20-9.univ.kiev.ua     s9      node9   ss20-9
```

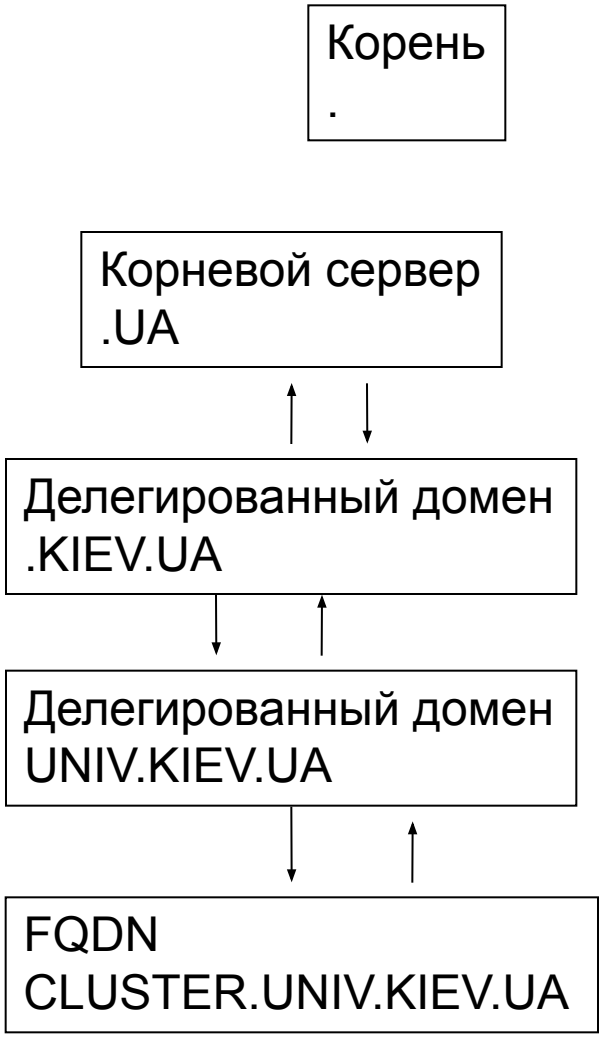
Недостатки старого метода

- Файл, который содержит все Интернет имена стал бы очень большим
- Отправка данных всем становится узким местом
- Была разработана иерархическая система имен

Иерархическая схема системы



Иерархическая схема имен



Обратный поиск

- Поиск имени по IP адресу
- Адрес 10.25.0.1
- Ему соответствует имя обратного поиска
  - 1.0.25.10.in-addr.arpa
  - Имя строится по тому же иерархическому принципу: сетевой адрес – корень
- Для локальных

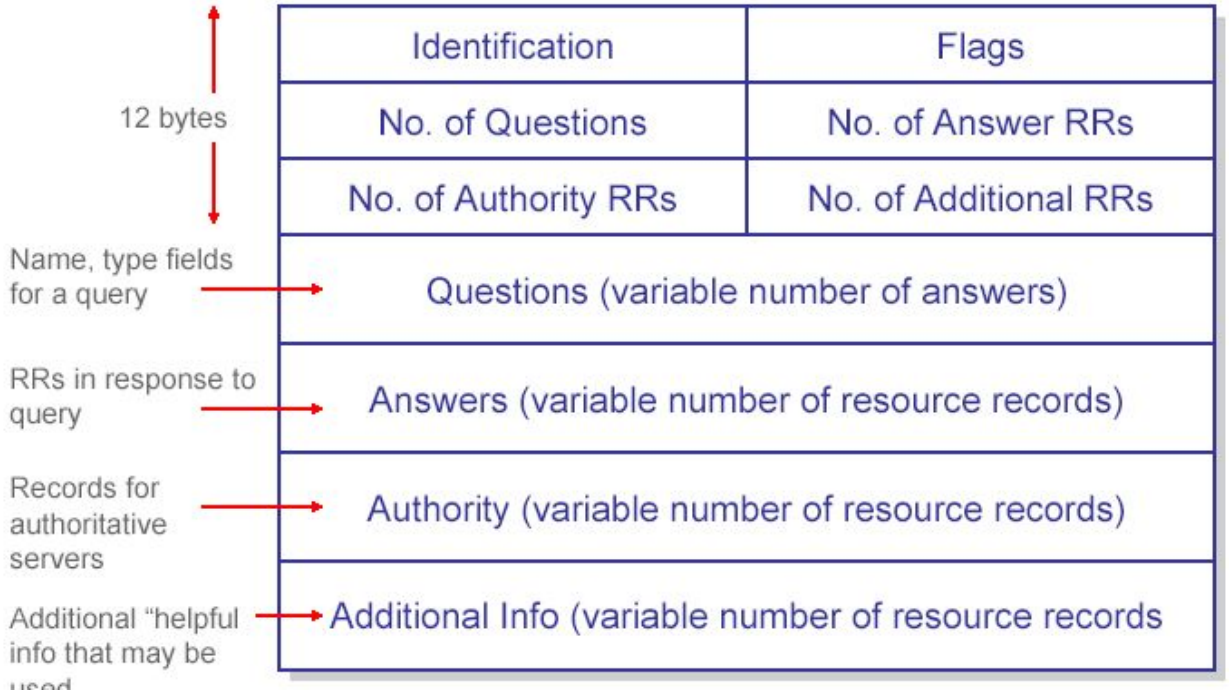
Типы запросов

- Рекурсивный запрос
  - Сервер берет на себя всю работу по определению адреса по доменному имени
  - Обычно выполняется локальными серверами
- Итеративный запрос
  - Сервер отвечает, что не знает этого имени и выдает адрес сервера, к которому нужно обратится
  - Обычно выполняются корневыми или промежуточными серверами

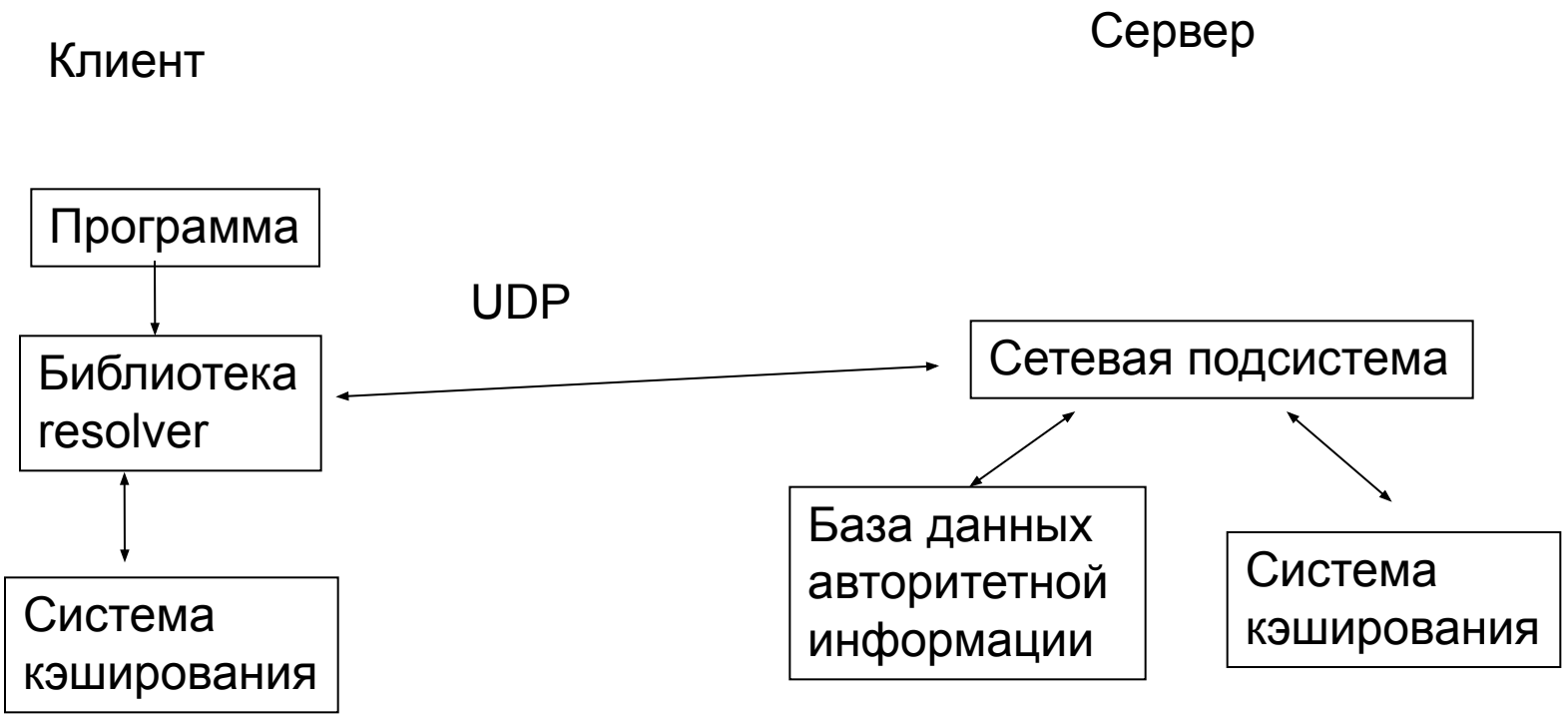
Протокол запросов-ответов

- Транспортный протокол UDP, порт 53
- Клиент отправляет сообщение серверу, заданному в конфигурации клиента, как локальный
- Сервер получает запрос, выполняет поиск, оправляет ответ
- Нет гарантии, что сервер или клиент получит сообщение
- В случае отсутствия ответа выполнятся повторный запрос
- Если несколько раз не получили ответа, то считается, что сервер не найден

Формат сообщения



Основные компоненты клиентов и серверов



База данных имен

- Хранится в виде файлов или в реляционной базе данных
- Файл

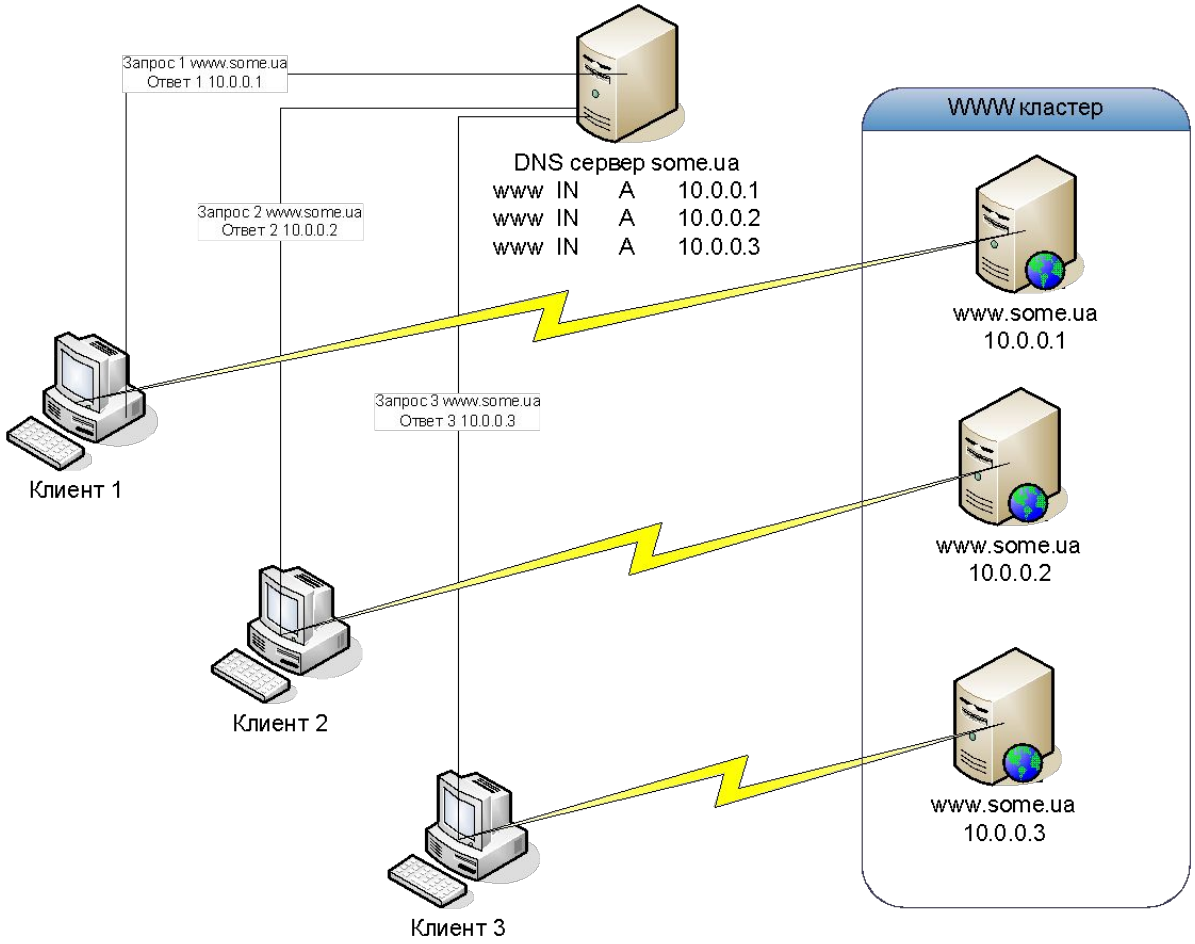
```
$TTL    86400
@       IN      SOA     ss20-1.univ.kiev.ua. root.localhost.  (
                                1997022700 ; Serial
                                28800      ; Refresh
                                14400      ; Retry
                                3600000    ; Expire
                                86400 )    ; Minimum
        IN      NS      ss20-1.univ.kiev.ua.

ss20-1  IN      A       10.25.5.1
ss20-2  IN      A       10.25.5.2
ss20-3  IN      A       10.25.5.3
ss20-4  IN      A       10.25.5.4
ss20-5  IN      A       10.25.5.5
ss20-6  IN      A       10.25.5.6
ss20-7  IN      A       10.25.5.7
ss20-8  IN      A       10.25.5.8
```

Балансировка нагрузки

- Есть несколько серверов, которые выполняют одну функцию
- Для обеспечения производительности
  - Разным серверам ставится в соответствие одно имя, но разные IP
  - При обращении клиентов по имени они будут получать последовательно разные IP адреса и обращаться к разным серверам

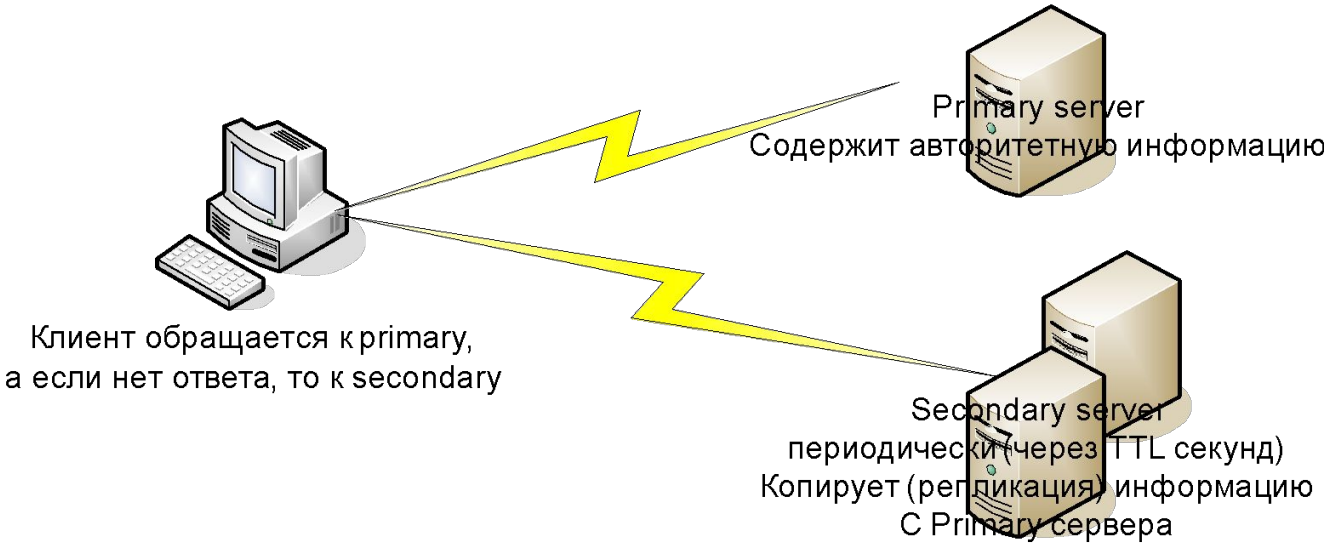
Пример: балансировка для WWW кластера



Программное обеспечение

- Сервер BIND
- Клиент
  - Resolver
    - /etc/hosts.conf
    - /etc/resolv.conf

Обеспечение надежности



Network Information System (NIS)

- Совместное использование файлов конфигурации на разных машинах
  - Как сделать так, чтобы на всех машинах организации (100 машин) были одинаковые имена пользователей?
  - Если нужно изменить или добавить пользователя, как это сделать быстро и без ошибок?
  - Как это же сделать с другими файлами конфигурации ?

Исторические сведения

- Разработана фирмой SUN вначале 1980-х в рамках ONC
- Позже появилась модификация NIS+ с шифрованием
- Сейчас используется в локальных сетях для создания распределенной базы данных конфигурационных файлов при не очень высоких требованиях к безопасности

Структура системы

- YP домен – набор машин, которые совместно используют файлы конфигурации
- YP master (server) – машина, которая содержит авторитетные файлы конфигурации и может раздавать их другим
- YP slave (server) – машина, которая содержит копии файлов конфигурации и тоже может их раздавать другим
- YP client (ypbind) – машина, которая использует копии файлов конфигурации

YP домен

- Ypbind (client) устанавливается на все машины, которые должны использовать разделяемые файлы конфигурации
- Ypserv (master) устанавливается только на одной машине домена и содержит авторитетные данные, которые при изменении отправляются всем вторичным серверам и клиентам
- Ypserv (slave) устанавливается на всех серверах, которые используются для репликации авторитетной информации

YP master

- Каждому совместно используемому файлу конфигурации соответствует NIS карта
- Карта – файл, который содержит всю информацию файла конфигурации и может быть отправлен клиенту
- При изменении файла конфигурации необходимо перестроить карту
- При получении запроса от клиента сервер отправляет ему карту

YP slave

- Вспомогательные сервера также могут отправлять карты клиентам
- На вспомогательных серверах файлы конфигурации изменять нельзя
- Для приема всех карт используется вспомогательный сервер ypxfrd

YP клиент

- Определяет сервер
- При обращении к некоторым данным из файлов конфигурации отправляет запрос серверу
- Если есть локальная копия карты, то используется эта карта

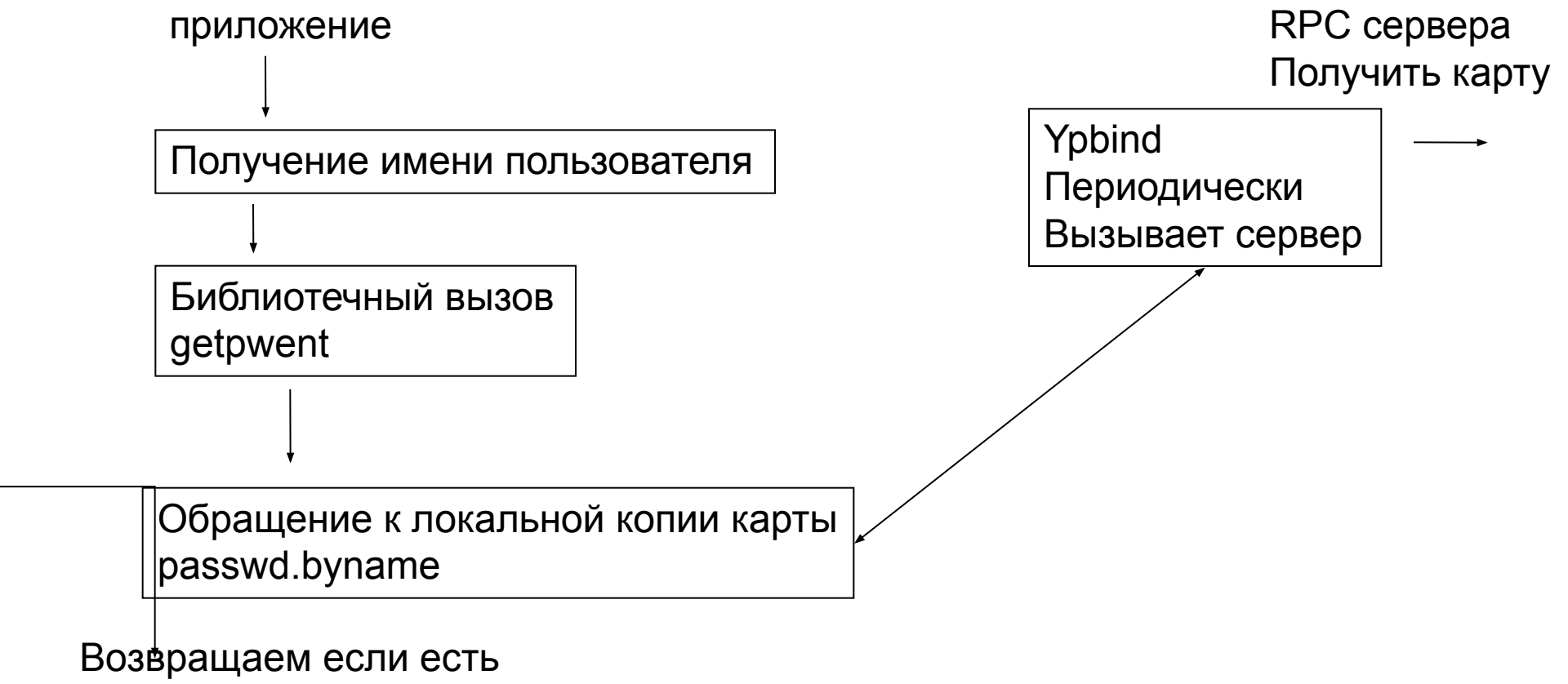
Взаимодействие клиентов и серверов

- RPC
  - ypserv 100004
  - ypbind 100007
  - ypxfrd 100069
  - yppasswd 100009

Карты

- passwd.byname
- passwd.byuid
- groups.byname
- groups.byuid
- hosts
- netgroup

Схема работы клиента



Недостатки NIS

- Рассчитан на широковещательные запросы и следовательно только на локальные сети
- Слабые средства обеспечения безопасности
  - Пароли передаются по сети (в зашифрованном виде)

Схема работы клиента

Служба каталогов (directory service)

- Служба каталогов – иерархическая база данных информации, организованная в виде дерева с отношениями родитель-потомок, по аналогии с организацией файловой системы
- С помощью службы каталогов можно выполнять поиск необходимых данных
- Первоначально такие системы разрабатывалась для создания информационных систем масштабов Интернет в 1990-х годах
- Для работы с распределенными каталогами был разработан протокол LDAP

Пример

- Файловая система

/

/home

/home/saa

/home/saa/bin/hack

- Интернет-каталог

/Ukraine

/Ukraine/Kyiv

/Ukraine/Kyiv/University

/Ukraine/Kyiv/University/cluster

Интернет каталог

c=UA
o=Kyiv University, c=UA
ou=ICC,o=Kyiv University,c=UA
ou=cluster,ou=ICC,o=Kyiv University,c=UA

c – country
o – organization
ou – organization unit
cn – common name

Полный путь к элементу каталога называется Distinguished Name (DN)
/C=ch/O=AliEn/OU=ALICE/CN=alien.univ.kiev.ua/SE

Объекты

- Каждый элемент каталога является объектом (аналог файла)
- Объекты бывают
  - Контейнерными (аналог каталога файловой системы)
  - «Листовыми» (аналог файла)
- Каждый объект имеет набор атрибутов и набор значений этих атрибутов
- Какие атрибуты и типы значений в каком объекте допустимы определяется схемой объекта

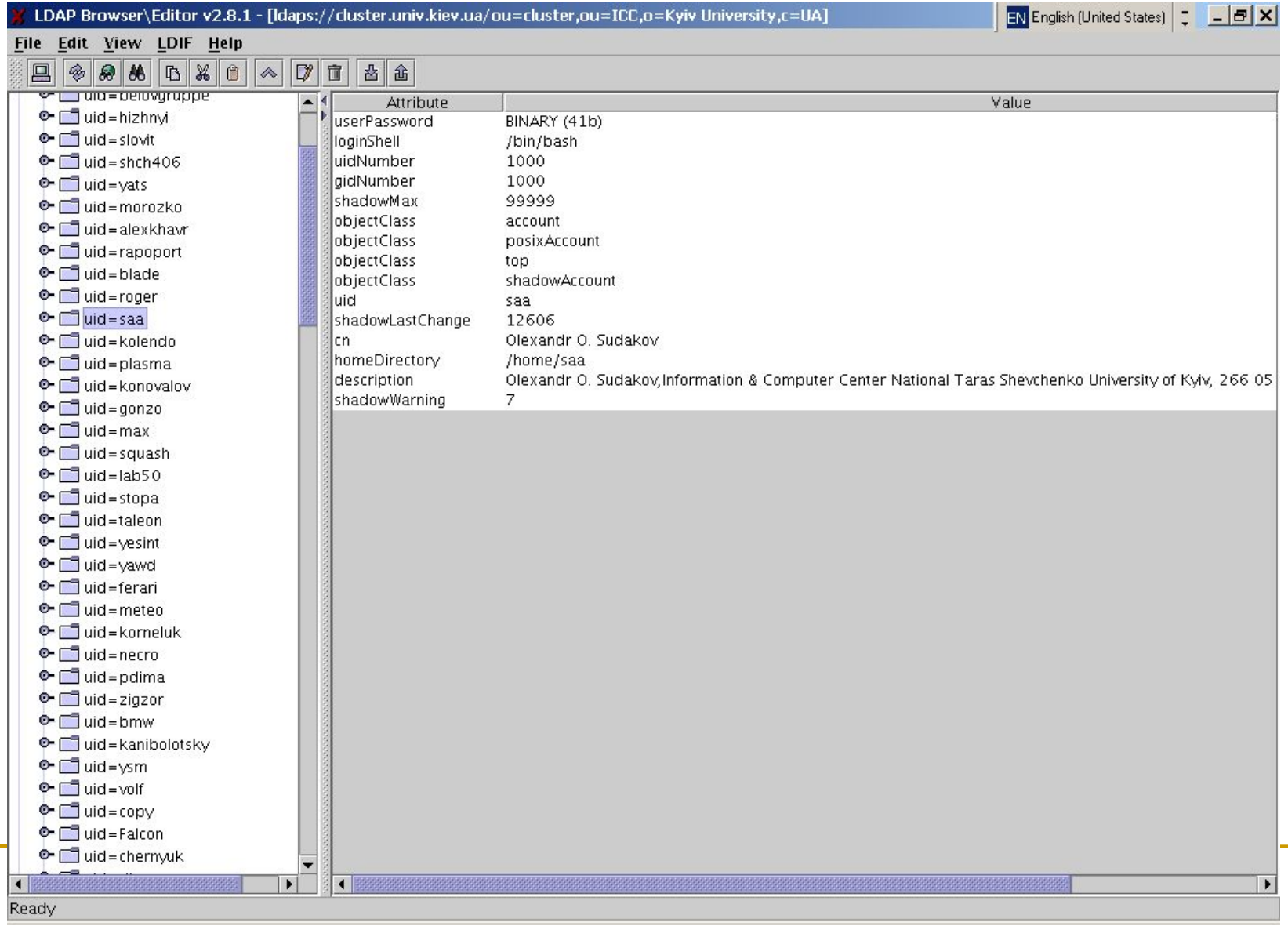
Пример объекта каталога

```
# saa, People, cluster, ICC, Kyiv University, UA
dn: uid=saa,ou=People,ou=cluster,ou=ICC,o=Kyiv University,c=UA
uid: saa
cn: Olexandr O. Sudakov
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
shadowMax: 99999
shadowWarning: 7
loginShell: /bin/bash
uidNumber: 1000
gidNumber: 1000
homeDirectory: /home/saa
description: Olexandr O. Sudakov,Information & Computer Center National Taras
 Shevchenko University of Kyiv, 266 05 30, saa@univ.kiev.ua
shadowLastChange: 12606
```

Схема LDAP

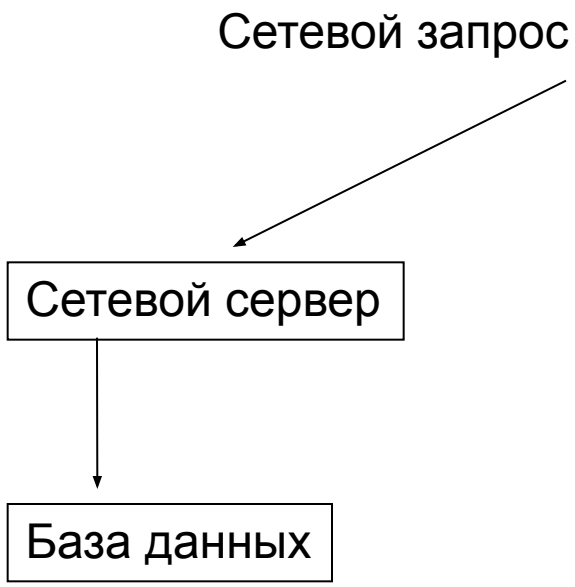
- Схема определяет
  - тип объекта
  - какие атрибуты и какого типа могут быть в объекте
  - Какие атрибуты обязательны, а какие – нет
- Существуют стандартизированные схемы
- Можно создавать свои схемы путем наследования существующих

Пример LDAP каталога



LDAP сервер

- Функции
  - Хранение
  - Поиск
  - Аутентификация
  - Запись
  - Шифрование/дешифрование



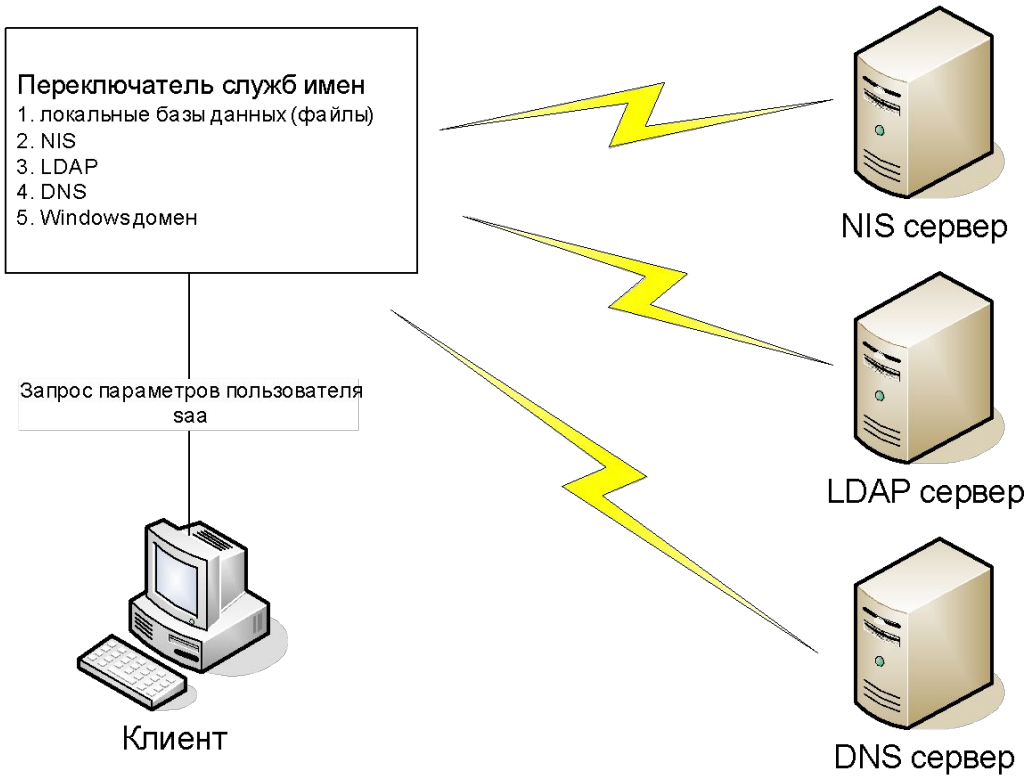
Обеспечение надежности и балансировка

- LDAP – иерархическая система
  - Для каждого элемента каталога может существовать свой сервер
- LDAP – поддерживает вспомогательные сервера, которые хранят ту же информацию, что и главные
- Вспомогательные сервера можно использовать для балансировки нагрузки

Использование LDAP

- Служба имен для очень больших организаций
- GRID системы
- Службы имен для распределенных систем с сильно географически распределенными компонентами
- В локальных сетях, как замена NIS

Переключатель служб имен



Nsswsitch (NSS)

- Middleware
  - Виртуализация служб имен
  - Быстрое подключение новой службы имен
- Впервые появилось в ОС Solaris
- Сейчас есть в каждой современной операционной системе

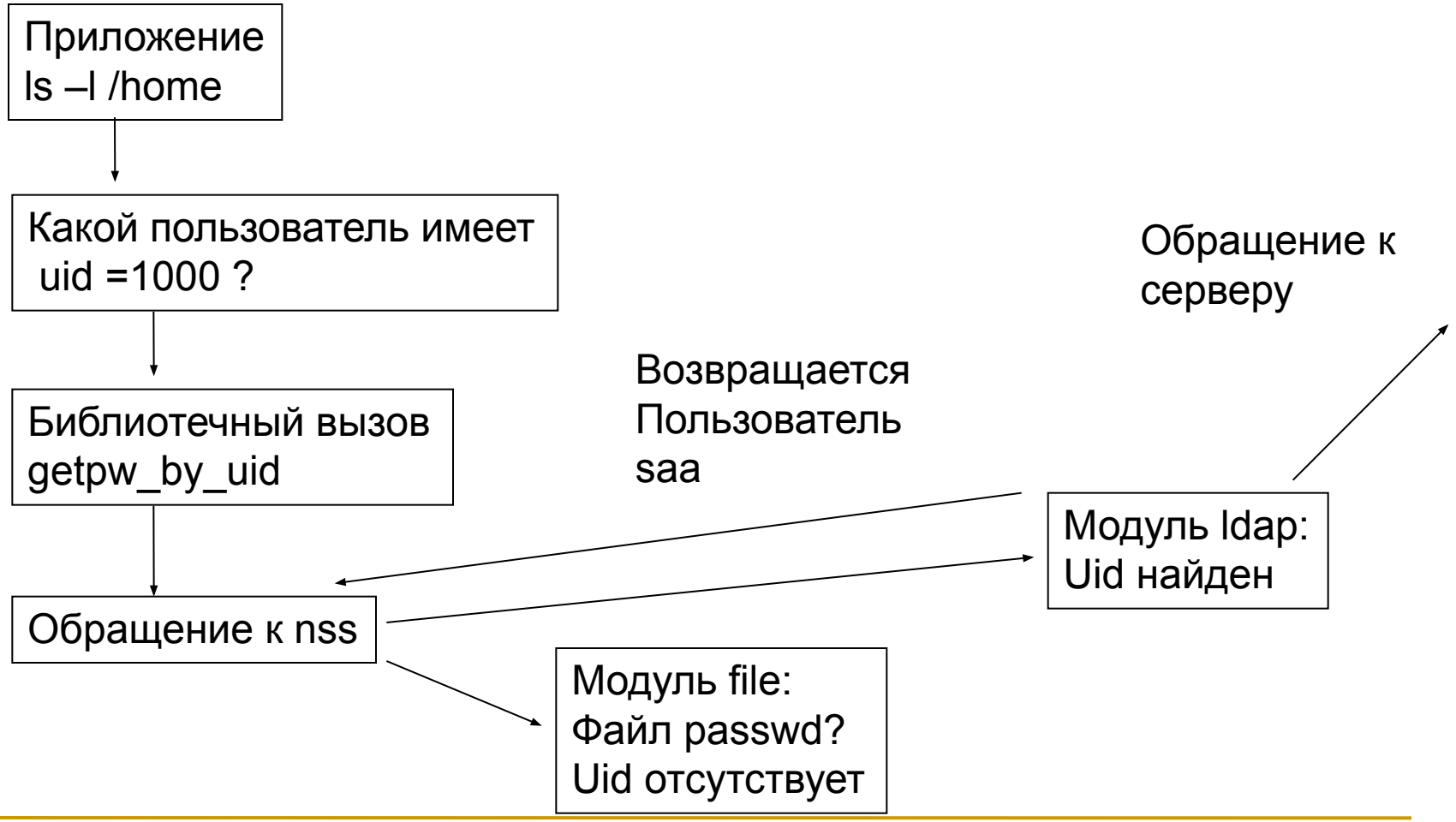
Linux NSS

- Включено в библиотеку glibc
- В конфигурационном файле администратор может установить необходимые параметры

```
passwd:     files ldap
shadow:     files ldap
group:      files ldap

#hosts:     db files nisplus nis dns
hosts:      files dns ldap
```

Пример работы клиента



Кэширование nss

- Обращение каждый раз к удаленным серверам приводит к большим затратам времени и перегруженности сети
- В nss существует специальная система кэширования всех известных данных
  - nscd

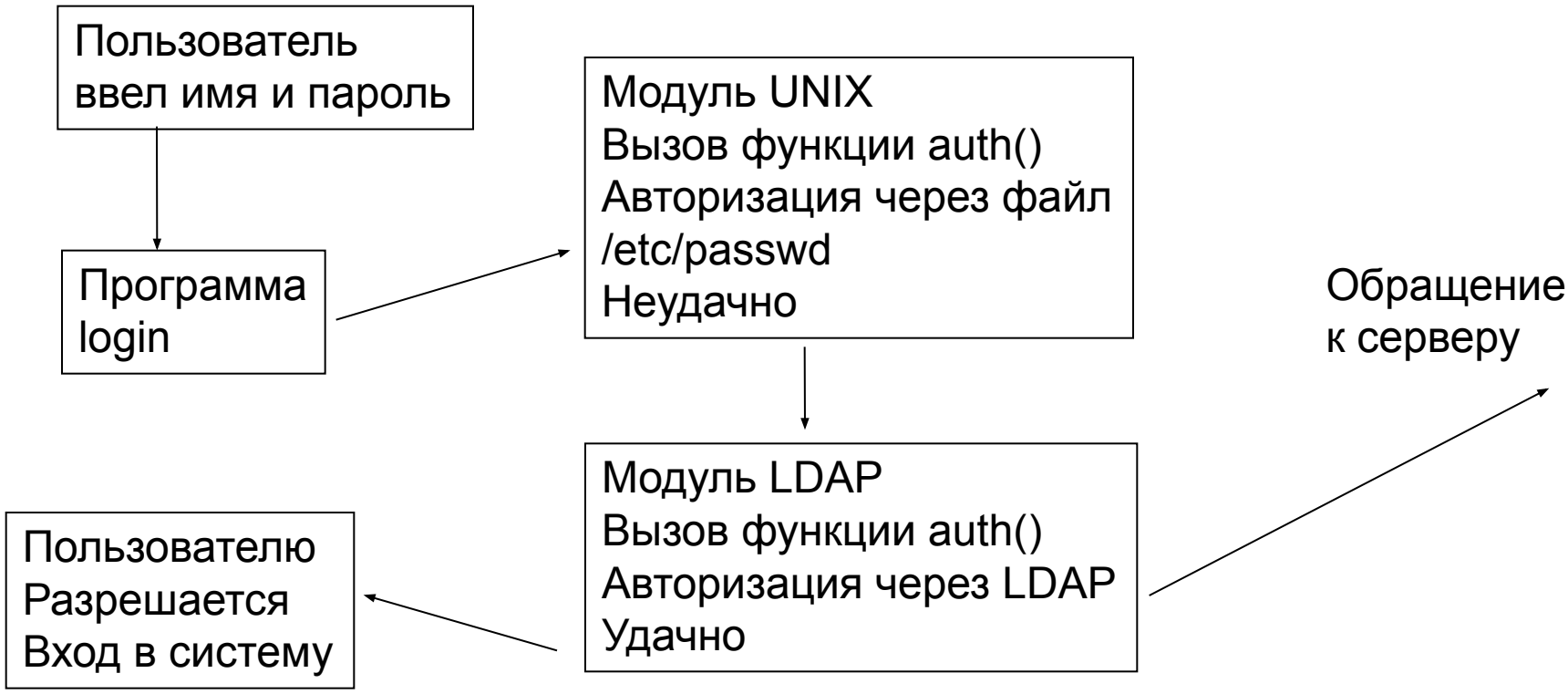
AAA (Authentication, Authorization, Accounting)

- Аутентификация
  - Кто такой?
- Авторизация
  - Что можно делать
- Учет ресурсов
  - Что делалось в процессе работы

PAM – pluggable authentication modules

- По аналогии с nss использование нескольких систем авторизации, аутентификации, учета
  - Данные из локальных файлов
  - LDAP
  - NIS
  - RADIUS
- Реализована в виде модулей, из которых последовательно вызываются необходимые функции

Пример работы PAM



Пример общего конфигурационного файла

```
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth        required      /lib/security/$ISA/pam_env.so
auth        sufficient    /lib/security/$ISA/pam_unix.so likeauth nullok
auth        sufficient    /lib/security/$ISA/pam_ldap.so use_first_pass
auth        required      /lib/security/$ISA/pam_deny.so

account     sufficient    /lib/security/$ISA/pam_succeed_if.so uid < 100
#account     required      /lib/security/$ISA/pam_listfile.so item=user sense
=deny file=/etc/security/nologin_users onerr=succeed
#account     [default=bad success=ok user_unknown=ignore service_err=ignore s
ystem_err=ignore] /lib/security/$ISA/pam_ldap.so
account     required      /lib/security/$ISA/pam_access.so
account     sufficient    /lib/security/$ISA/pam_unix.so
account     [default=bad success=ok user_unknown=ignore service_err=ignore sy
stem_err=ignore] /lib/security/$ISA/pam_ldap.so

password    required      /lib/security/$ISA/pam_cracklib.so retry=3 type=
password    sufficient    /lib/security/$ISA/pam_unix.so nullok use_authtok m
d5 shadow
password    sufficient    /lib/security/$ISA/pam_ldap.so use_authtok
password    required      /lib/security/$ISA/pam_deny.so
```

Вопросы ?