

Именование и поиск

Сухорослов Олег Викторович

Распределенные системы

Факультет компьютерных наук НИУ ВШЭ

17.10.2020

Адрес

- Для доступа к объектам в распределенной системе (узлы, процессы, сервисы, данные, пользователи...) требуется знать их **адрес**
 - Особый вид имени, связанный с "местом" в сети (access point, endpoint)
 - Примеры: MAC-адрес, IP-адрес и порт
- Свойства адресов
 - У объекта может быть несколько адресов
 - Адрес объекта может изменяться
 - Адрес может быть в будущем назначен другому объекту
 - Часто имеют фиксированную длину
 - Неудобны для людей (human-unfriendly)

Имена

- Идентификатор
 - Ссылается не более чем на один объект
 - На каждый объект ссылается не более одного идентификатора
 - Идентификатор всегда ссылается на один и тот же объект
 - Не зависит от текущего местоположения (адреса) объекта
 - Примеры: hash(data), UUID, URN
- Удобное (*human-friendly*) имя
 - Стока символов, часто включающая слова и имена из языка
 - Примеры: имя файла, доменное имя, URL

Схемы именования

- Плоская (flat)
 - У имени нет компонентов, плоское пространство имен
 - Пример: простое имя, ключ данных в DHT
- Структурированная
 - Иерархическое пространство имен (дерево)
 - Пример: доменное имя в DNS
- На основе атрибутов
 - С объектом связаны пары (атрибут, значение)
 - Пример: атрибуты объектов в LDAP

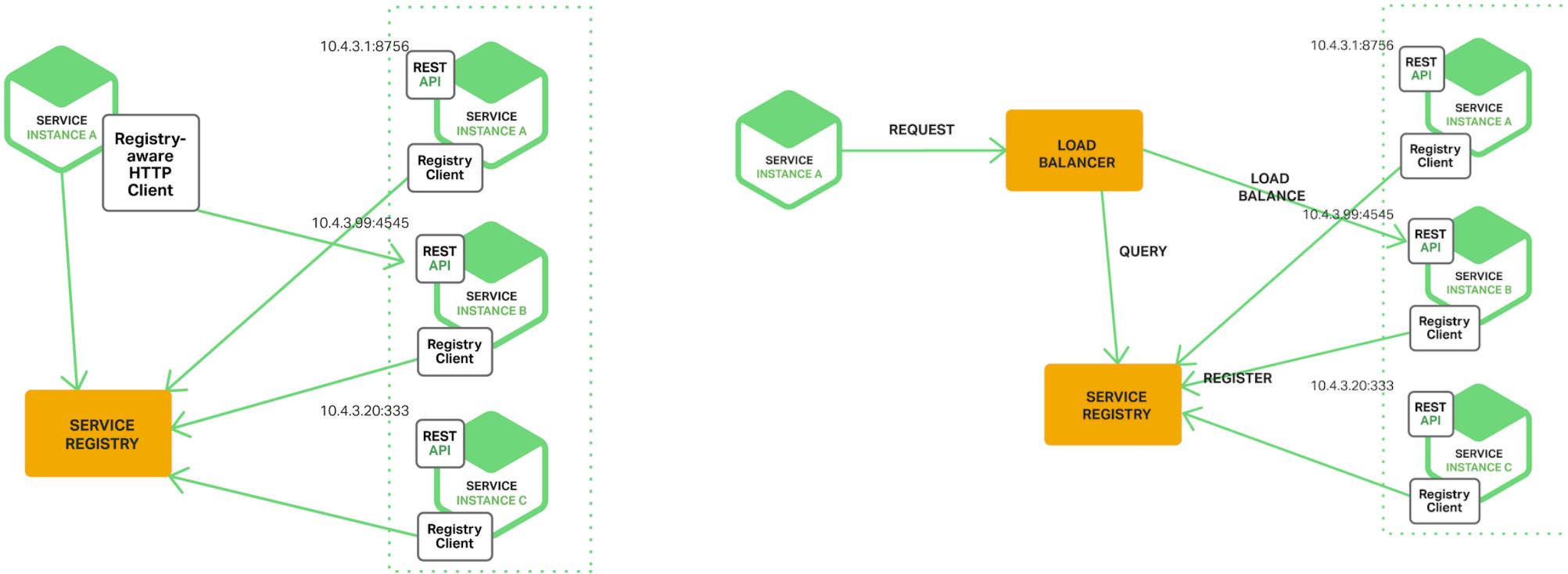
Проблемы

- Как назначать имена?
 - Избежание коллизий
 - Разделение пространства имен на отдельно управляемые части
- Как из имени получить адрес?
 - Разрешение имени: $\text{resolve}(\text{name}) \rightarrow \text{address}$
 - Поиск данных: $\text{search}(\text{name}) \rightarrow \text{value}$
- Как проверить что с объектом действительно связано это имя?
 - Хэш от содержимого объекта является self-certifying name

Реализация разрешения имен

- Широковещательная рассылка или мультикаст
 - ARP, ZeroConf (mDNS, Bonjour, Avahi)
- Хранить таблицу вида (*name, address*)
 - Размещение копии таблицы на каждом узле
 - Централизованный сервис (хранит всю таблицу)
 - Распределенный сервис
 - каждый узел хранит копию всей таблицы
 - каждый узел хранит часть таблицы

Промышленные решения



Примеры: Eureka, Consul, etcd, Zookeeper...

HOSTS.txt

ARPANET DIRECTORY
NIC 19275
Jan. 1974

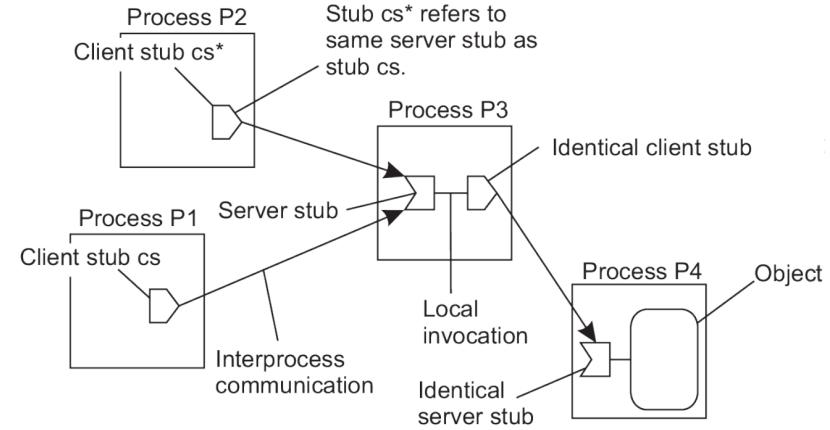
HOST NAMES

HOST NAMES

HOSTNAME	HOST ADDR (Dec)	LIAISON	STATUS
<hr/>			
AFWL-TIP	176	D Hyde (505)247-1711 x3803	TIP, Up 3-74
ALOHA-TIP	164	R Binder (808)948-7066	TIP
AMES-11	208	J Hart (415)965-5935	USER, up 12-73
AMES-67	16	W Hathaway (415)965-6033	SERVER
AMES-TIP	144	W Hathaway (415)965-6033	TIP
ANL	?	L Amiot (312)739-7711 x4309	SERVER, up 2-74
ARPA-DMS	28	S Crocker (202)694-5037	USER, Agency use only
ARPA-TIP	156	S Crocker (202)694-5037	TIP
BBN-11X	5	R Thomas (617)491-1850 x483	Peripheral processor for #69, up 12-73
BBN-1D	232	A McKenzie (617)491-1850 x441	USER
BBN-NCC	40	A McKenzie (617)491-1850 x441	USER
BBN-TENEX	69	R Thomas (617)491-1850 x483	SERVER
BBN-TENEXB	133	R Thomas (617)491-1850 x483	SERVER, Limited
BBN-TESTIP	158	A McKenzie (617)491-1850 x441	TIP (magtape)
BELVOIR	27	W Andrews (703)664-5511	USER, up 6-74
BRL	29	M Romanelli (301)278-4574	USER
CASE-10	13	J Calvin (216)368-2984	SERVER
CCA-TENEX	31	R Winter (617)491-3670	SERVER
CCA-TIP	159	R Winter (617)491-3670	TIP
CMU-10A	78	H Van Zoeren (412)621-2600 x160	SERVER
CMU-10B	14	H Van Zoeren (412)621-2600,x160	SERVER
CMU-11	142	C Pierson (412)621-2600 x130	USER, up Spring 74
CMU-CC	206	D King (412)621-2600 x2683	USER, up Spring 74
DOCB-TIP	153	S Stevenson (303)499-1000 x3138	TIP
EGLIN	?	E Blackwell (904)882-3734	Up 3/74
ETAC	20	G Petregal (202)433-3911	USER, up Spring 74
ETAC-TIP	148	G Petregal (202)433-3911	TIP (magtape)
FNWC	33	M Reese (408)646-2817	USER, up 2-74
FNWC-TIP	161	M Reese (408)646-2817	TIP
GWC-TIP	152	A Wells (402)294-2968	TIP (magtape)
HARV-1	73	B Reussow (617)495-4147	USER
HARV-10	2	B Reussow (617)495-4147	SERVED

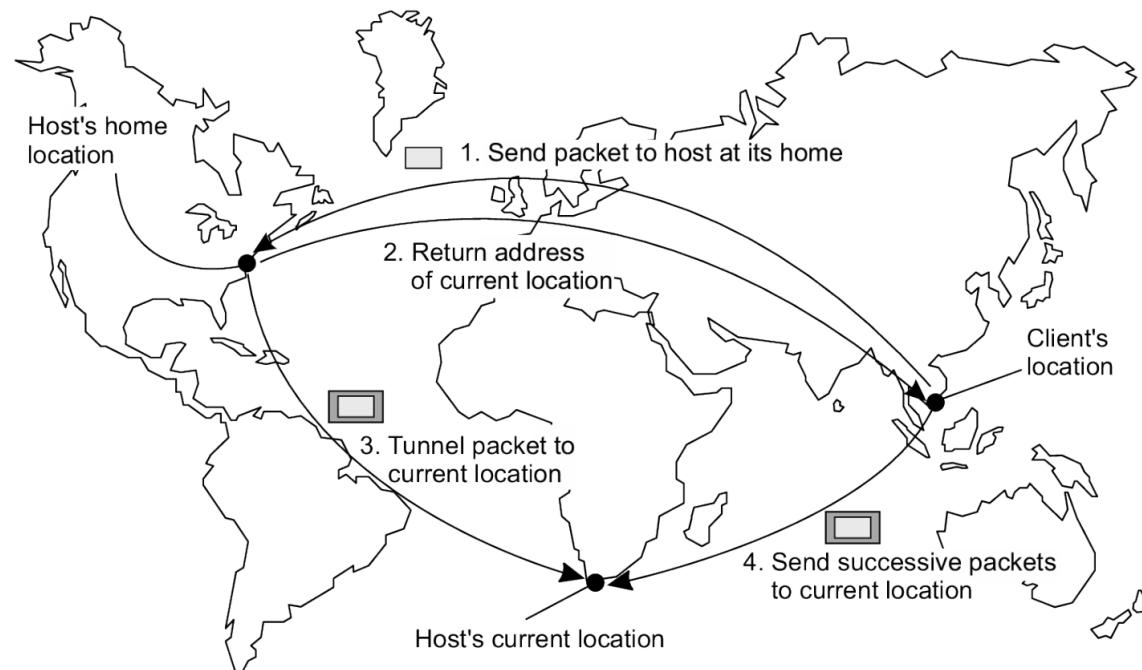
Forwarding Pointers

- При перемещении объект оставляет на старом узле ссылку на новый узел
- Запросы пересыпаются по цепи ссылок
- Ответ (и текущий адрес) передается отправителю напрямую или по цепи
- Проблемы: рост длины цепи, накладные расходы на поддержку, потеря звена

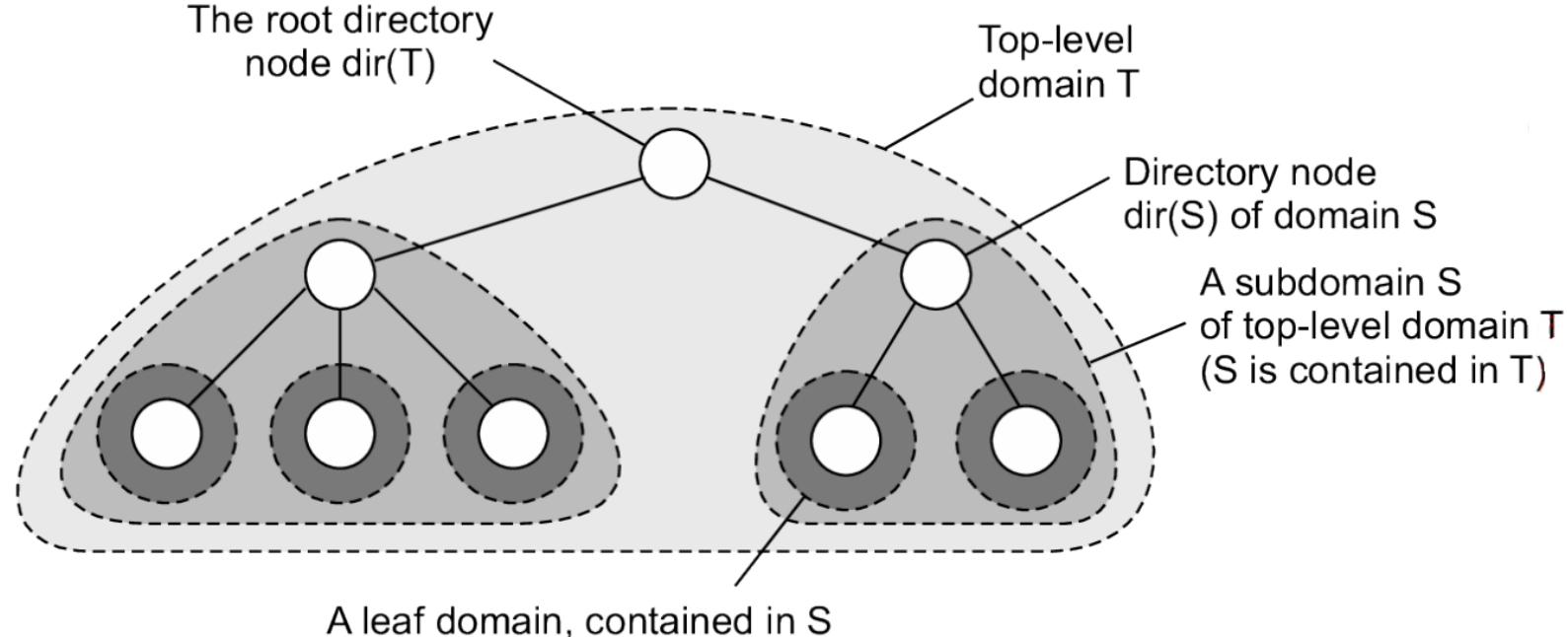


Home Location

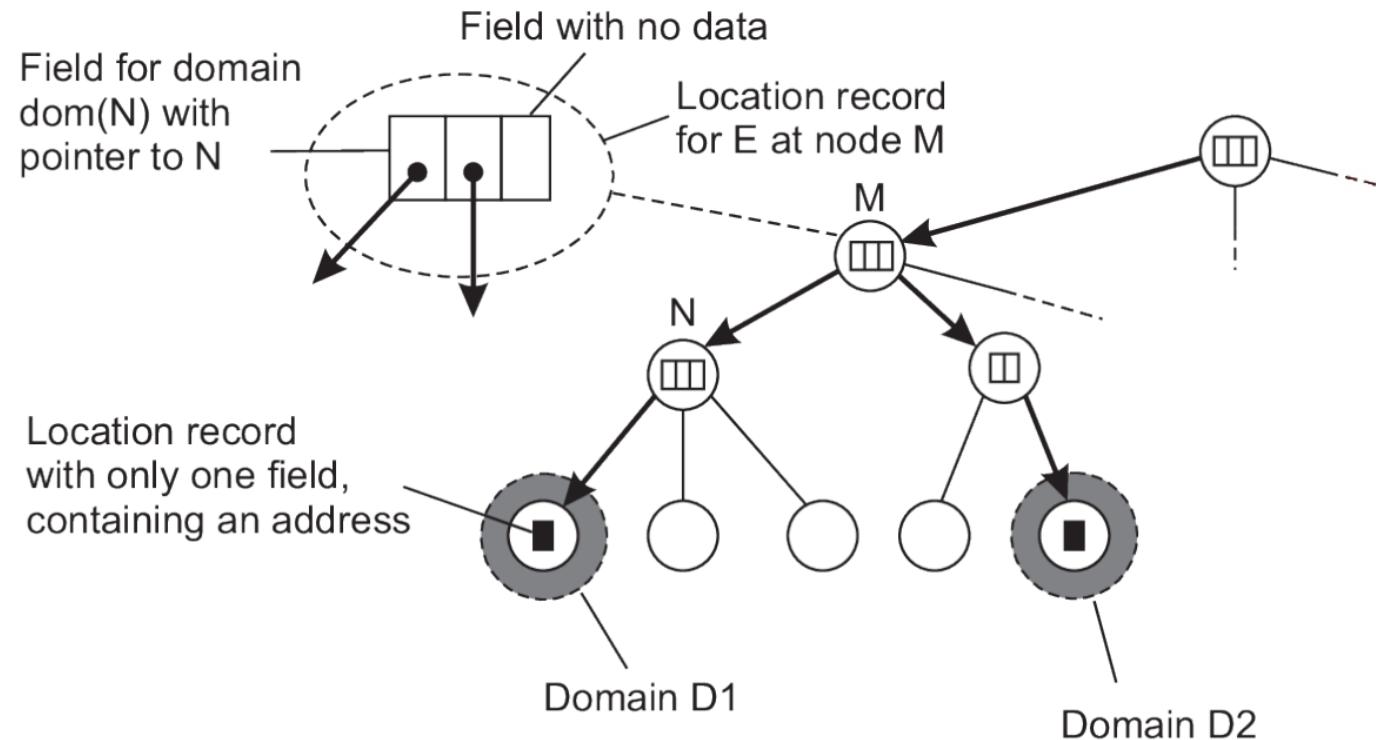
- Для каждого объекта есть стабильный адрес, по которому можно узнать текущий адрес объекта
- Проблемы: рост задержки при пересылке запроса, отказ или смена home



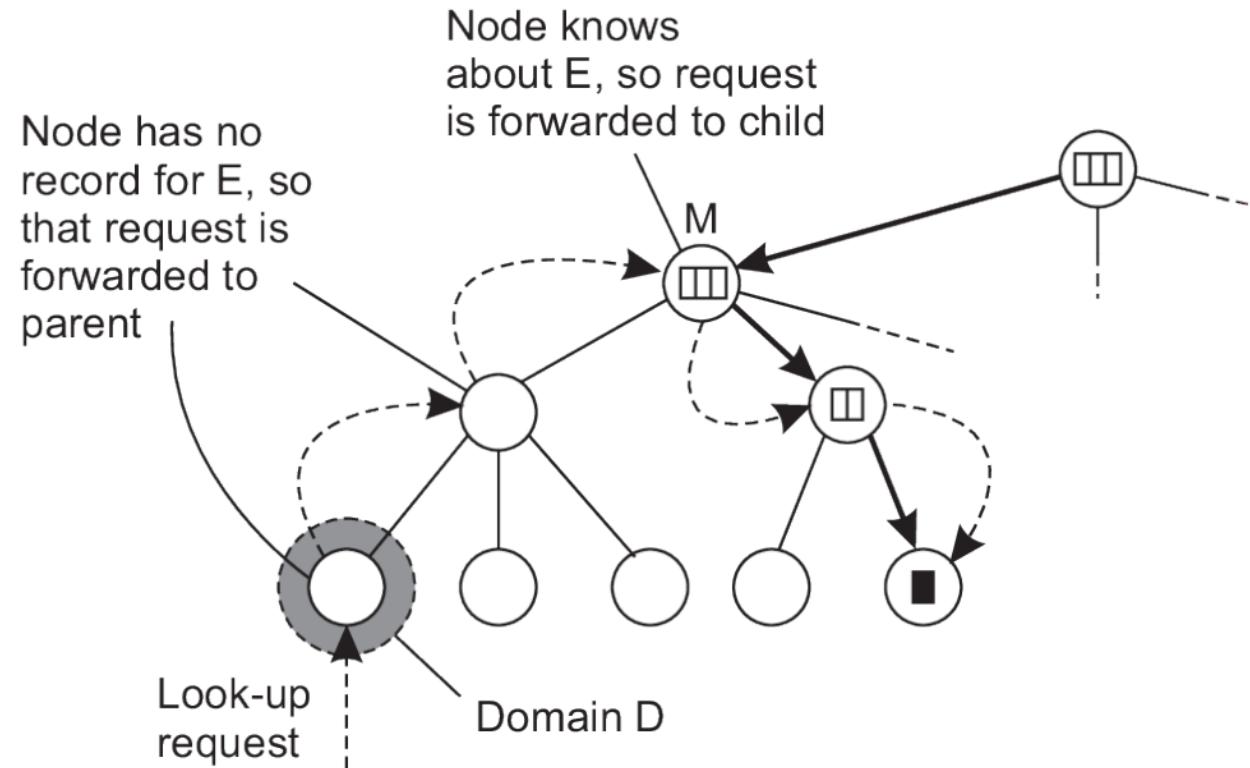
Иерархический подход



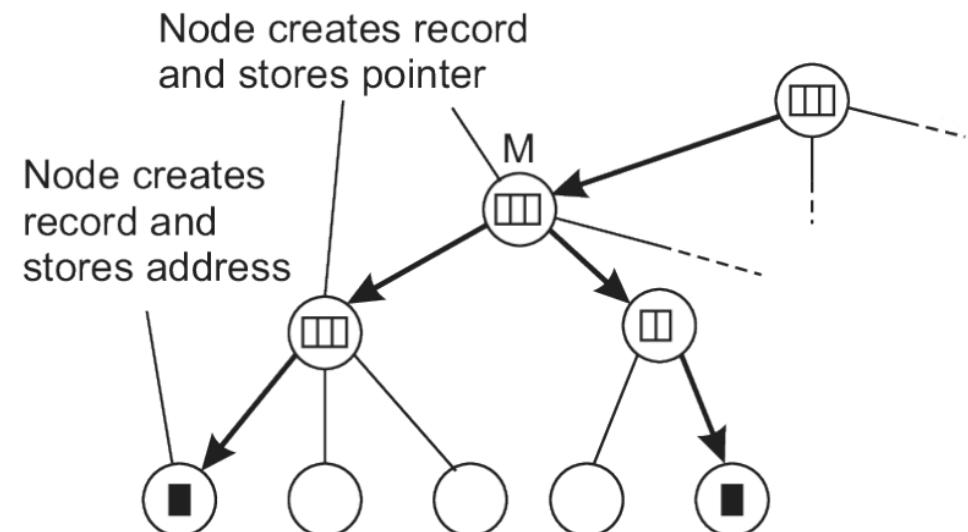
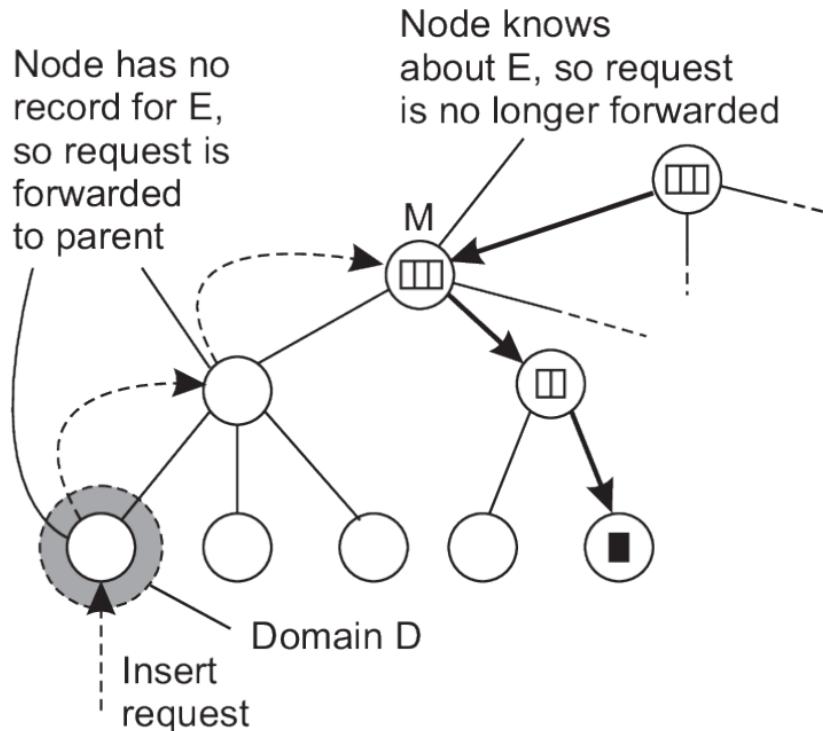
Иерархический подход: хранимые данные



Иерархический подход: поиск

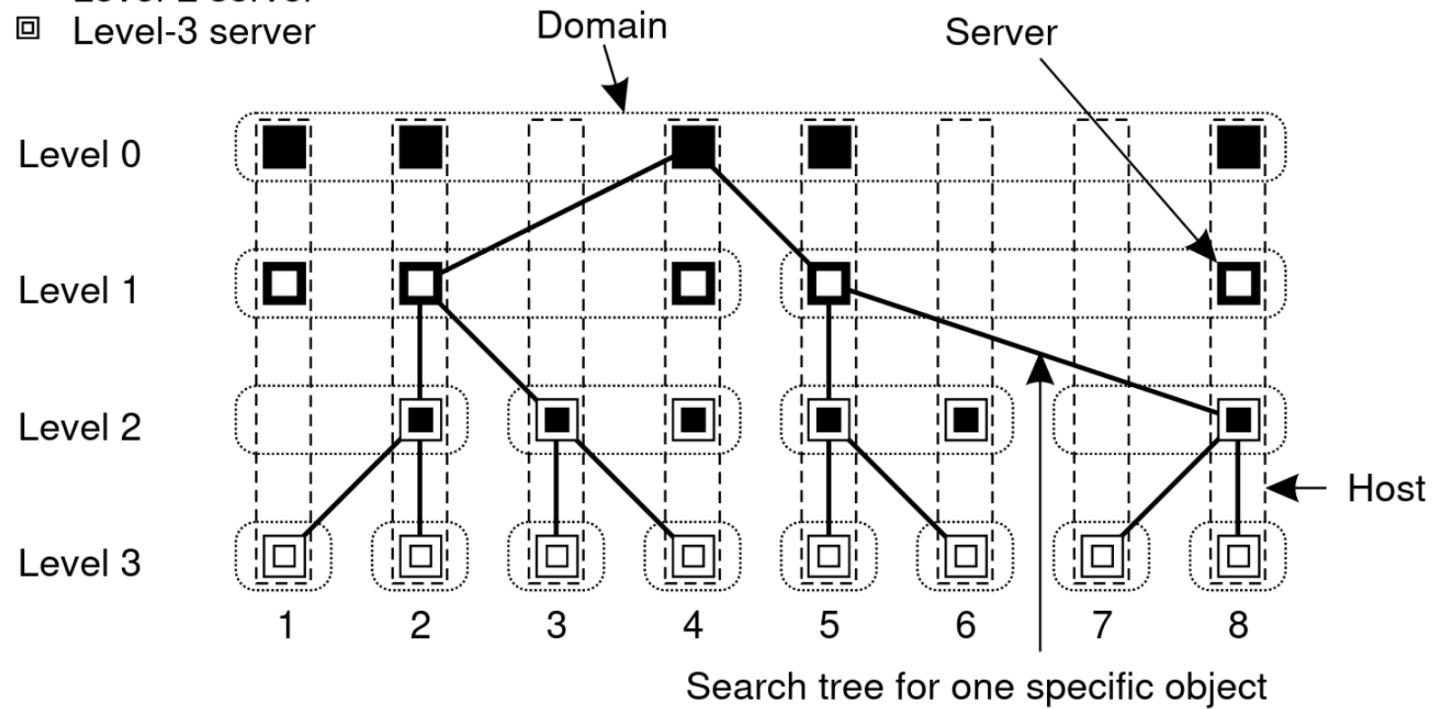


Иерархический подход: вставка



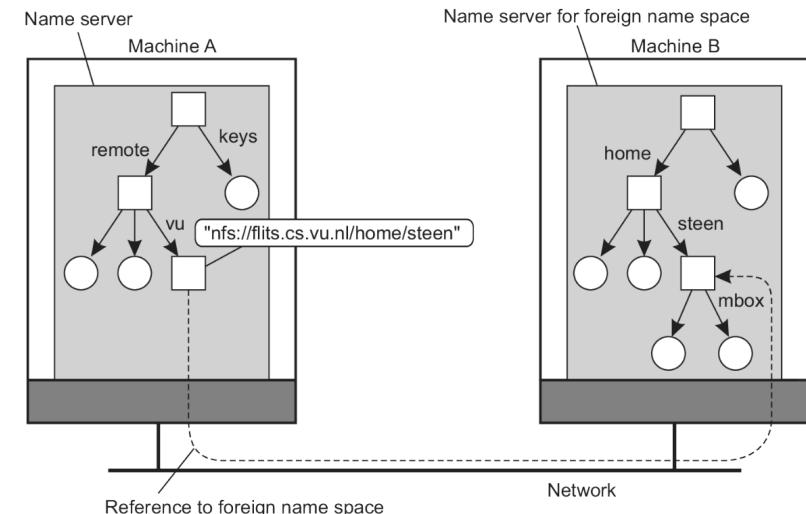
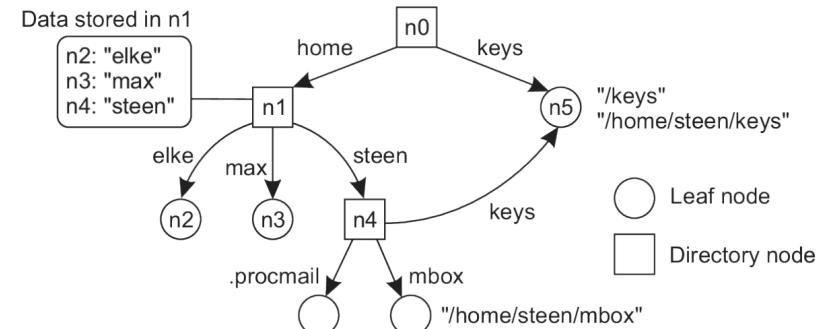
Иерархический подход: масштабирование

- Level-0 server
- Level-1 server
- Level-2 server
- Level-3 server



Структурированные имена

- Организованы в пространство имен
 - Один или несколько корневых вершин
 - Строго иерархическая структура (дерево)
 - Направленный ациклический граф
- Особенности
 - Имена образуют путь в графе
 - Относительные и абсолютные имена
 - Алиасы и точки монтирования
- Примеры использования
 - Файловые системы, ОС, DNS



Domain Name System (DNS)

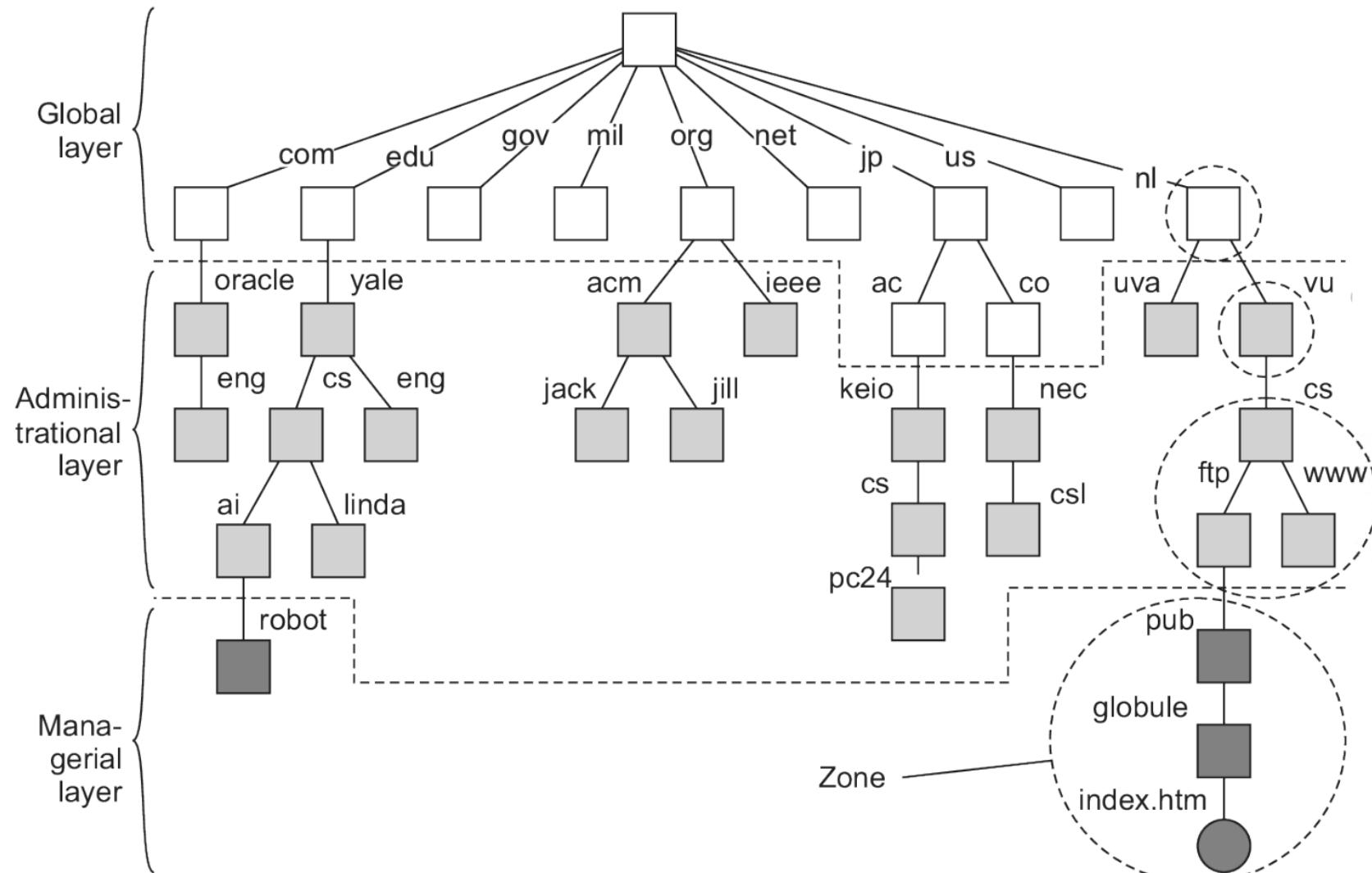
...one can draw the somewhat surprising conclusion that even after more than 30 years, DNS gives no indication that it needs to be replaced. We would argue that the main cause lies in the designer's deep understanding of how to keep matters simple. Practice in other fields of distributed systems indicates that not many are gifted with such an understanding.

van Steen M., Tanenbaum A.S. Distributed Systems: Principles and Paradigms.

DNS: централизованный подход?

- Единая точка отказа
 - Многие транзакции в Интернете зависят от DNS
- Высокая нагрузка и трафик
 - Миллионы записей, триллионы запросов в день
- Неравномерная задержка
 - Требуется низкая задержка вне зависимости от расположения клиента
- Обслуживание и поддержка
 - Миллионы организаций, отвечающих за свои записи
- Такой подход не будет масштабироваться

DNS: пространство имен



DNS: основные понятия

- **Домен:** узел в пространстве имён вместе со всеми дочерними узлами (поддерево)
- **Зона:** часть пространства имён, размещаемая как единое целое на некотором сервере
- **DNS-сервер:** может отвечать за некоторые зоны и/или перенаправлять запросы вышестоящим серверам
- **Ресурсная запись:** единица хранения и передачи информации в DNS
- **Делегирование:** передача ответственности за часть пространства имён другому лицу или организации
- **Авторитетность:** ответ на запрос поступил от сервера, отвечающего за зону

Ресурсные записи

Формат: (*Name*, *Value*, *Type*, *Class*, *TTL*)

Type=A

- Name: hostname
- Value: IP address

Type=NS

- Name: domain
- Value: hostname of authoritative name server for domain

Type=CNAME

- Name: alias name
- Value: canonical name

Type=MX

- Name: domain
- Value: hostname of mail server for domain

Пример

root:

```
(edu, a3.nstld.com, NS, IN)
(a3.nstld.com, 192.5.6.32, A, IN)
(com, a.gtld-servers.net, NS, IN)
(a.gtld-servers.net, 192.5.6.30, A, IN)
...
...
```

a3.nstld.com:

```
(princeton.edu, dns.princeton.edu, NS, IN)
(dns.princeton.edu, 128.112.129.15, A, IN)
...
...
```

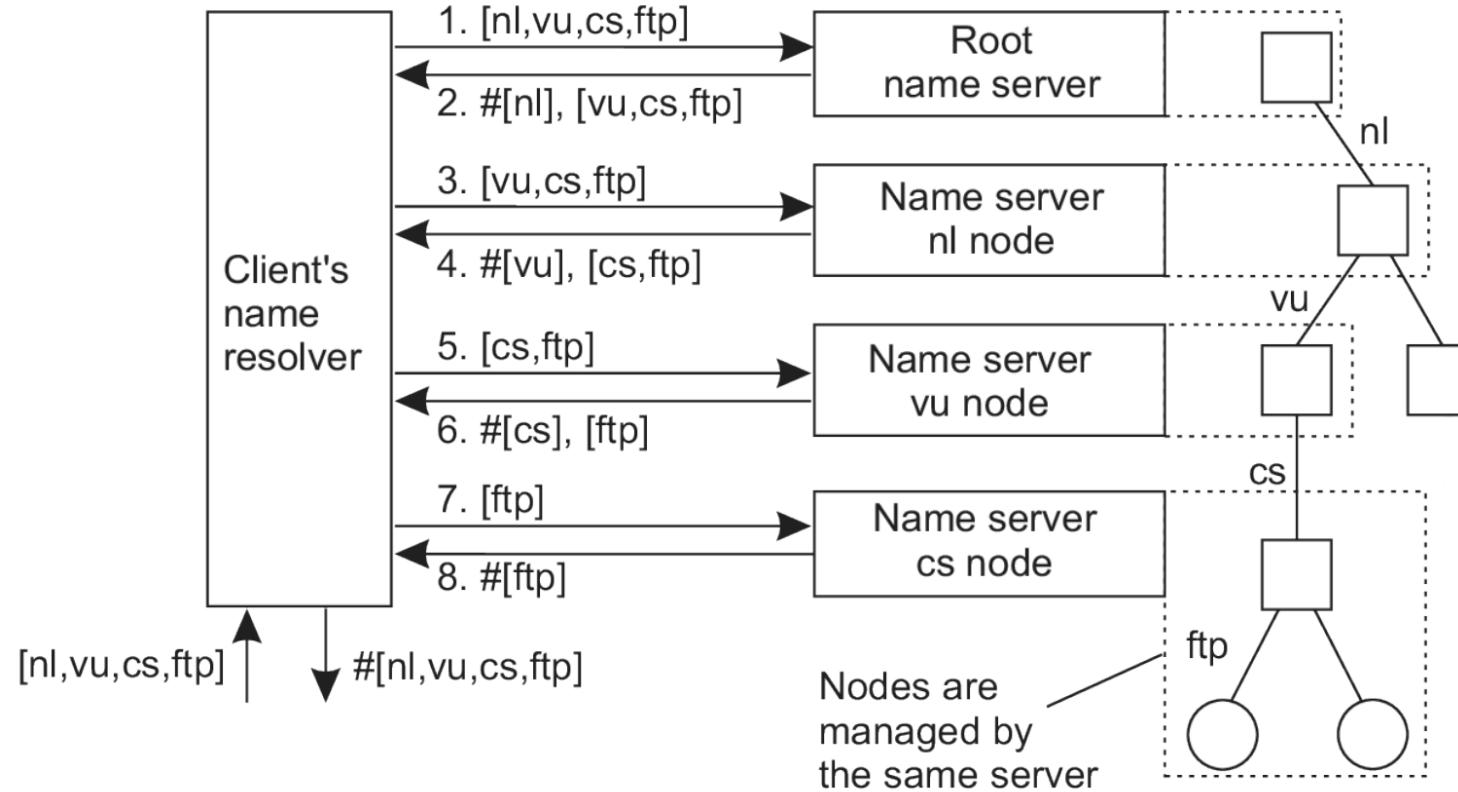
dns.princeton.edu:

```
(email.princeton.edu, 128.112.198.35, A, IN)
(cs.princeton.edu, dns1.cs.princeton.edu, NS, IN)
(dns1.cs.princeton.edu, 128.112.136.10, A, IN)
...
...
```

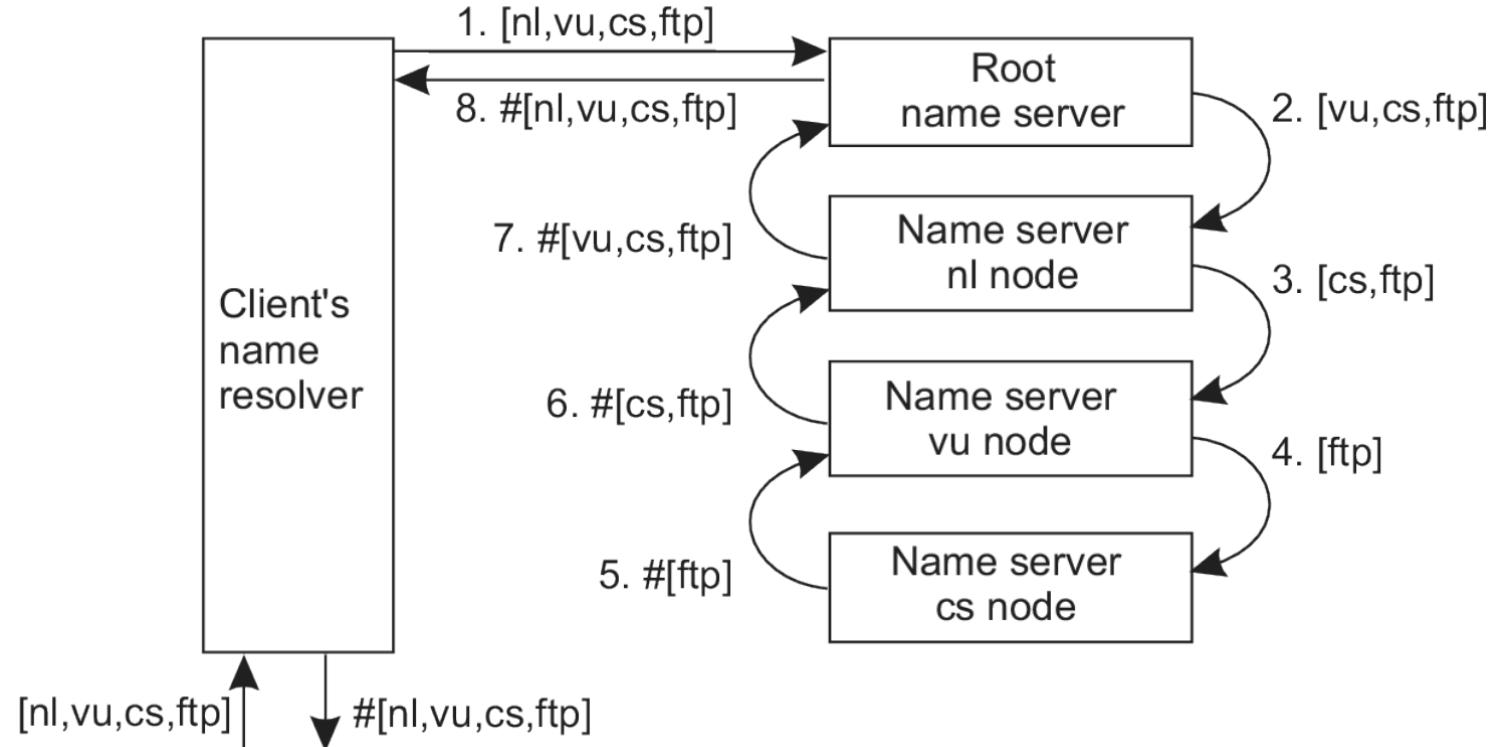
dns1.cs.princeton.edu:

```
(penguins.cs.princeton.edu, 128.112.155.166, A, IN)
(www.cs.princeton.edu,
 coreweb.cs.princeton.edu, CNAME, IN)
(coreweb.cs.princeton.edu, 128.112.136.35, A, IN)
(cs.princeton.edu, mail.cs.princeton.edu, MX, IN)
(mail.cs.princeton.edu, 128.112.136.72, A, IN)
...
...
```

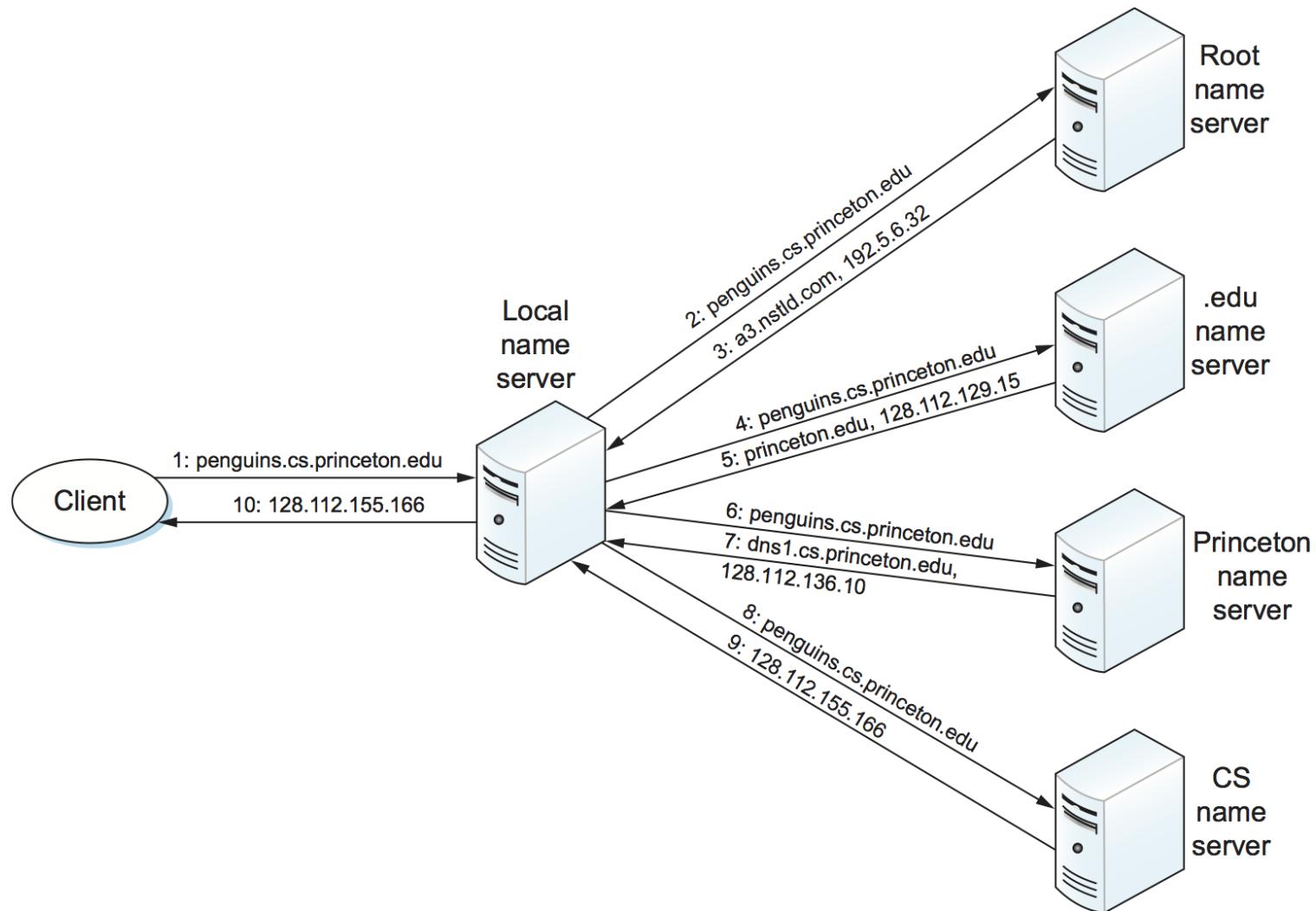
Итеративное разрешение имени



Рекурсивное разрешение имен

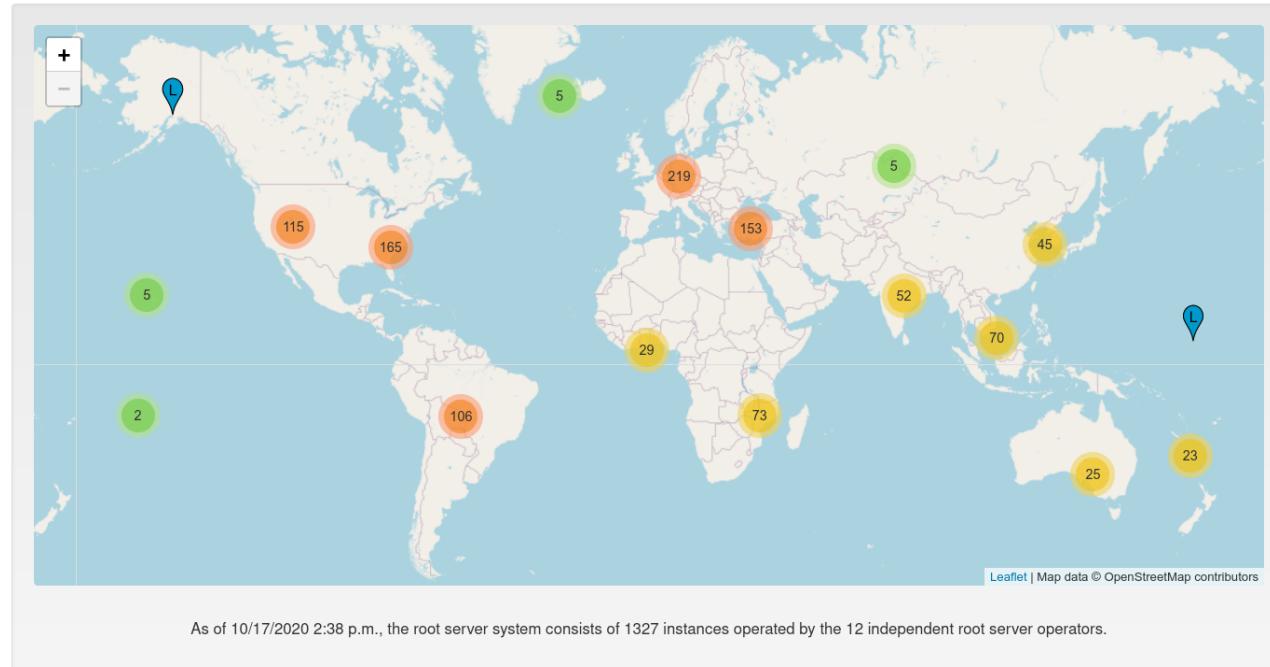


Разрешение имен в DNS



DNS: корневые серверы

- Корневой зоной управляет ICANN
- 13 логических корневых серверов (один обслуживается самой ICANN)
- Каждый логический сервер имеет множество реплик (~1K серверов)

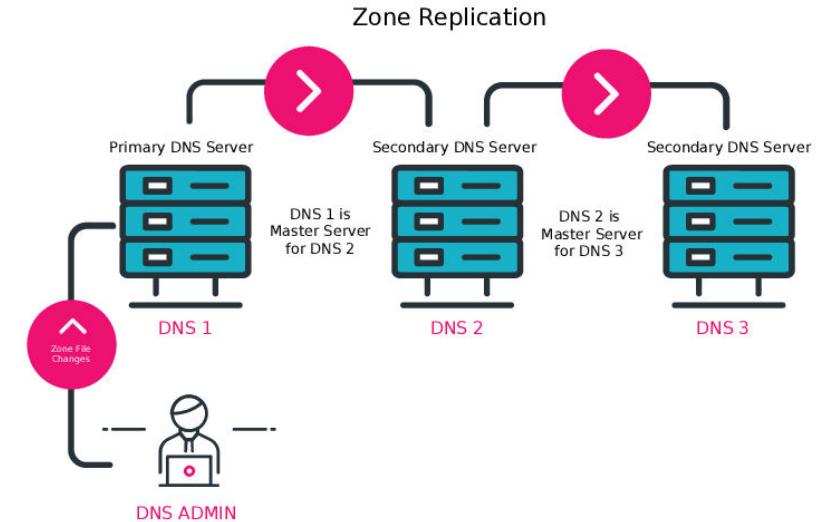


DNS: кэширование

- Сервер сохраняет полученные от других серверов записи в локальном кэше
 - Последующие аналогичные запросы используют записи из кэша
 - Записи в кэше удаляются по истечении времени жизни (TTL)
- Локальные серверы обычно хранят в кэше адреса TLD-серверов
 - Позволяет снизить нагрузку на корневые серверы
- Записи в кэше могут быть неактуальными
 - Изменения дойдут до всех серверов только после истечения TTL
 - Не требуется "дорогая" синхронизация реплик

DNS: репликация

- С зоной может быть связан набор серверов
 - один primary, несколько secondary
- Обновления применяются на primary
- Изменения передаются на secondary по запросу (zone transfer)



Отличия между уровнями

Корневые и TLD сервера:

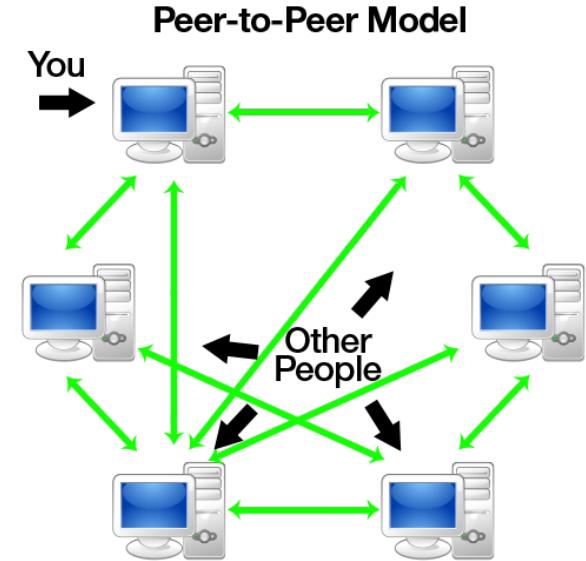
- Широкая география
- Мало логических серверов
- Скорость ответа менее критична
- Медленное распространение изменений
- Большое число реплик

Низлежащие уровни:

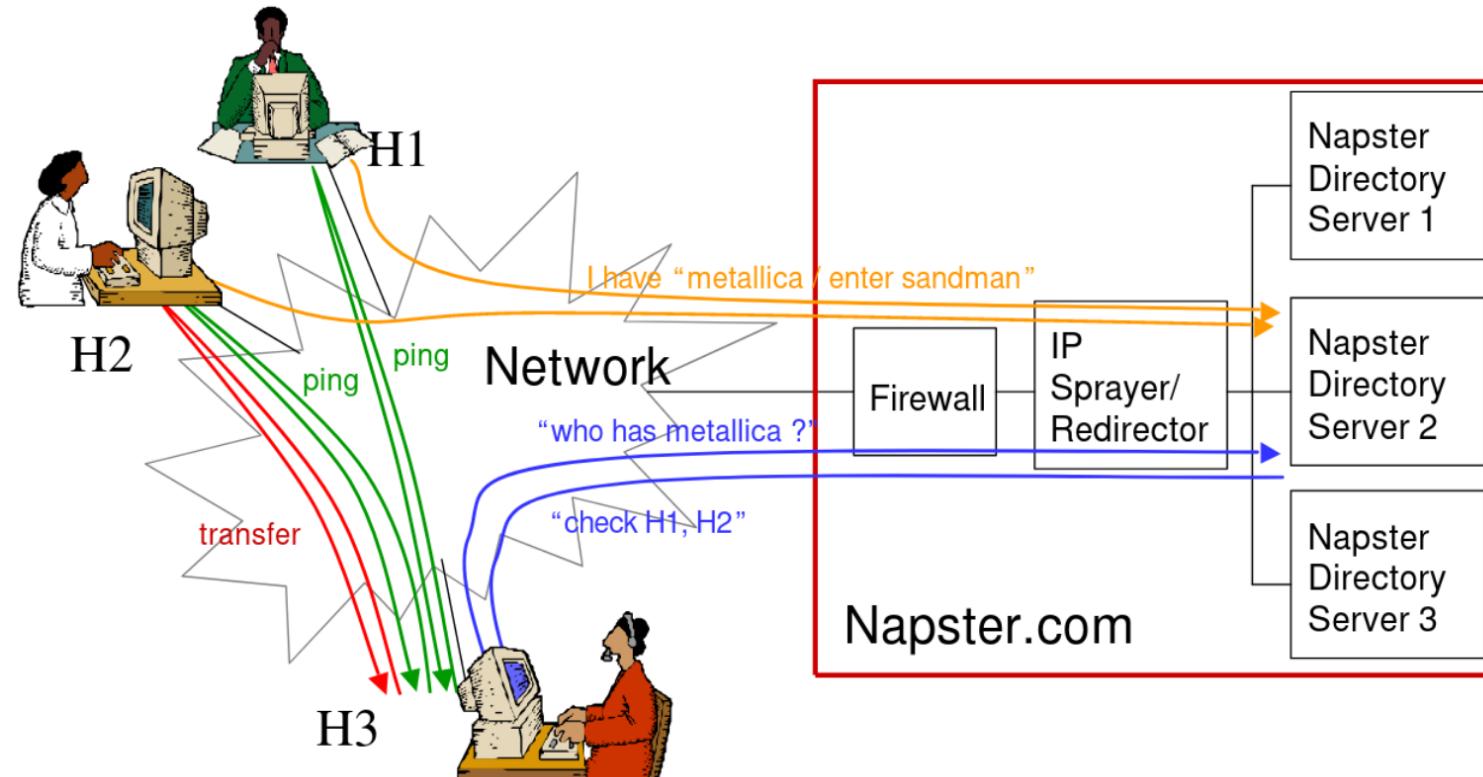
- Обычно организация
- Много серверов (суммарно)
- Скорость ответа критична
- Быстрое распространение изменений
- Небольшое число реплик

Peer-to-Peer (P2P) архитектура

- Отсутствуют выделенные, постоянно работающие серверы
- Участники (peers) взаимодействуют напрямую, играя как роли клиентов, так и серверов
 - Предоставление сервиса другому участнику
 - Получение сервиса от другого участника
- Участники могут отключаться и менять адреса
- Как разрешать имена в таких системах?
 - Например, определить адрес файла по его имени

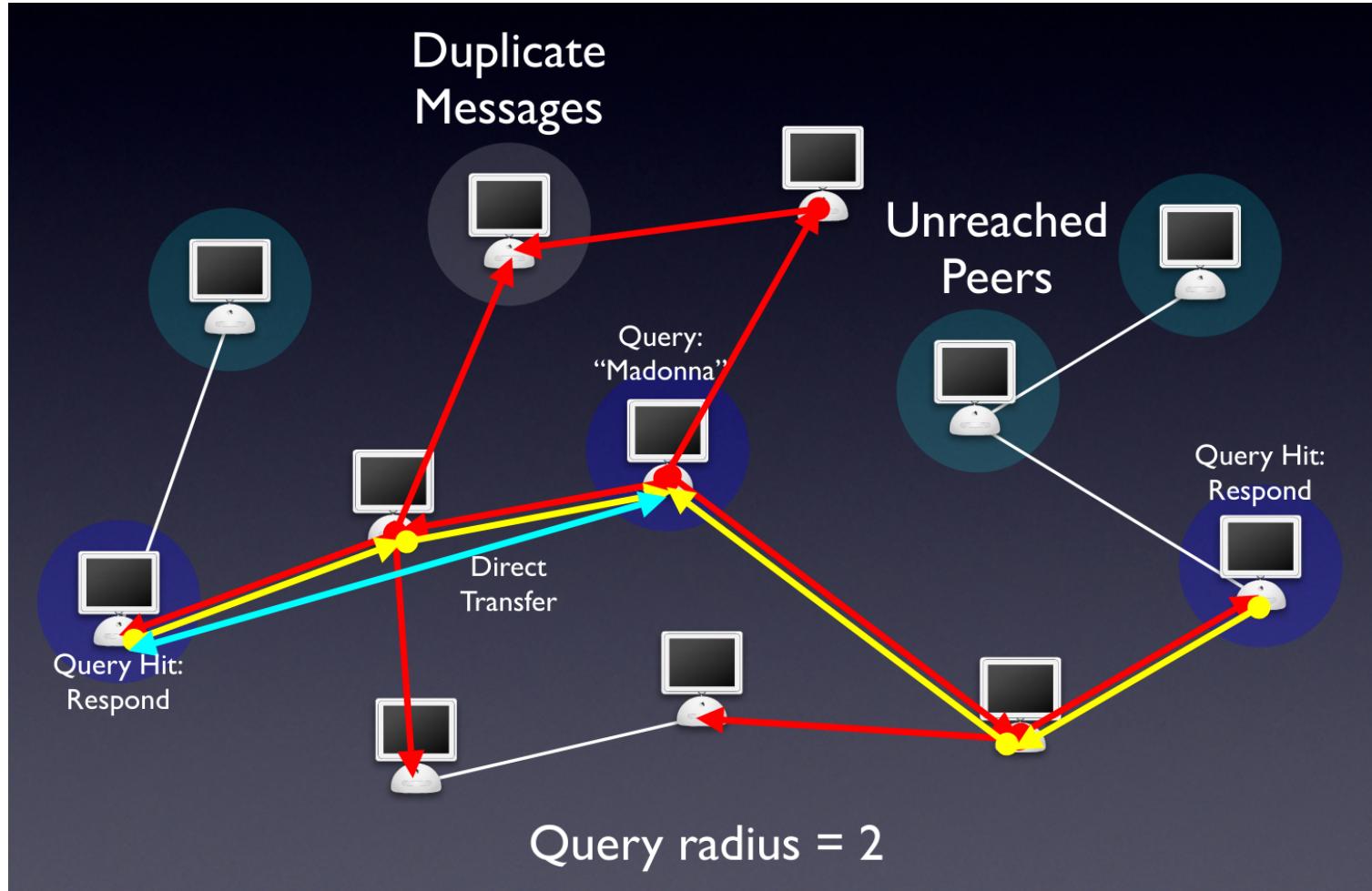


Napster (1999)



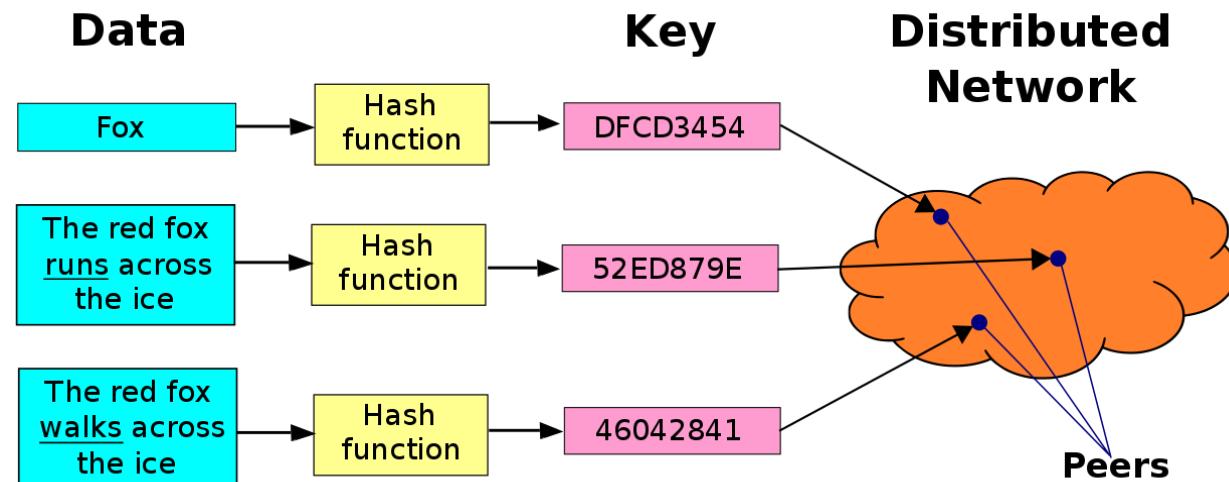
https://galaxis.com/posts/napster_protocol_overview/

Gnutella (2000)

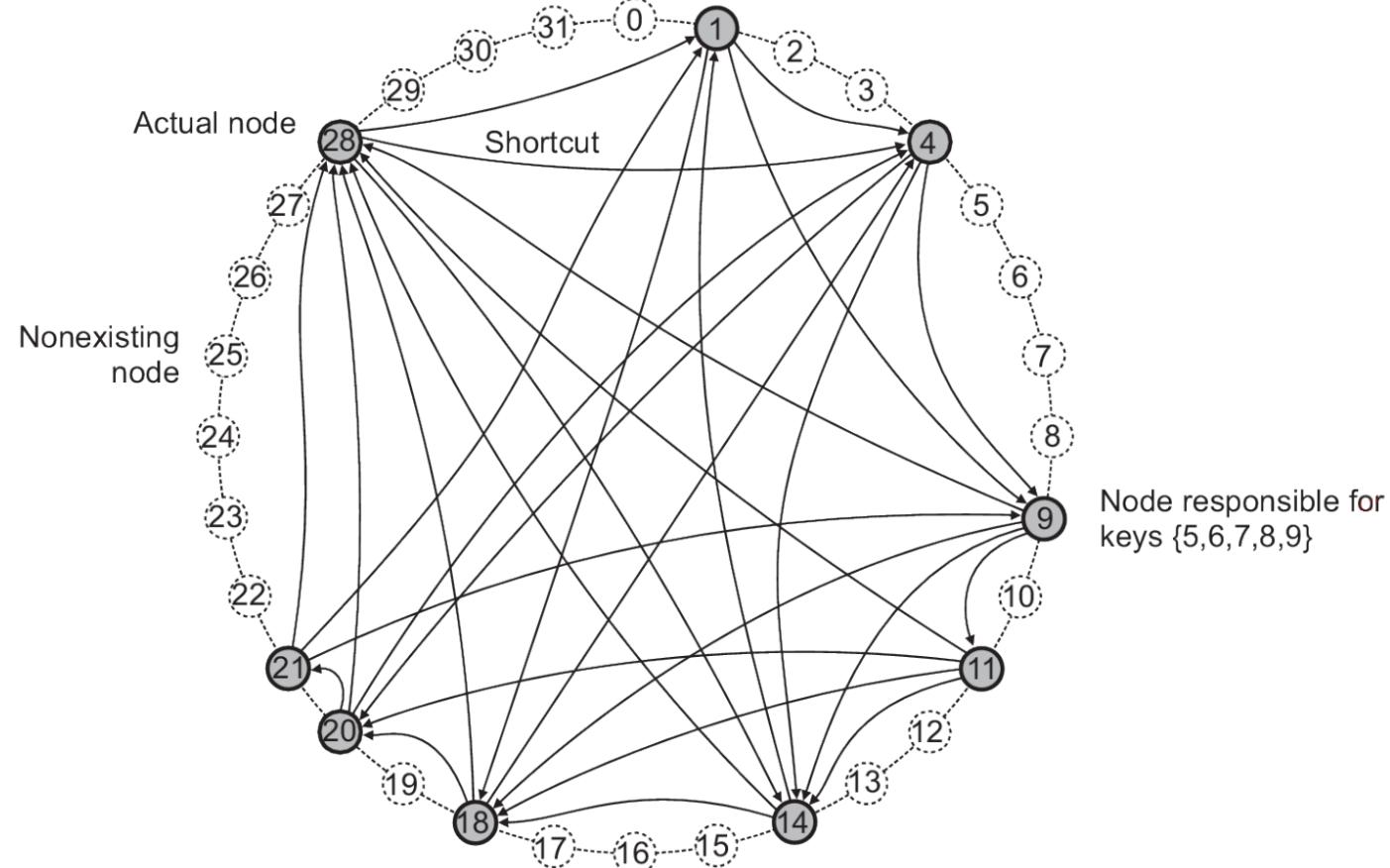


Distributed Hash Table (DHT)

- Объекты отображаются в плоское пространство имен (ключи фиксированной длины)
- Узел отвечает за некоторый фрагмент пространства имен (хранит часть данных)
- Узел хранит адреса других узлов для маршрутизации запросов (оверлейная сеть)
- Примеры использования: BitTorrent, Coral CDN, Tox, InterPlanetary File System

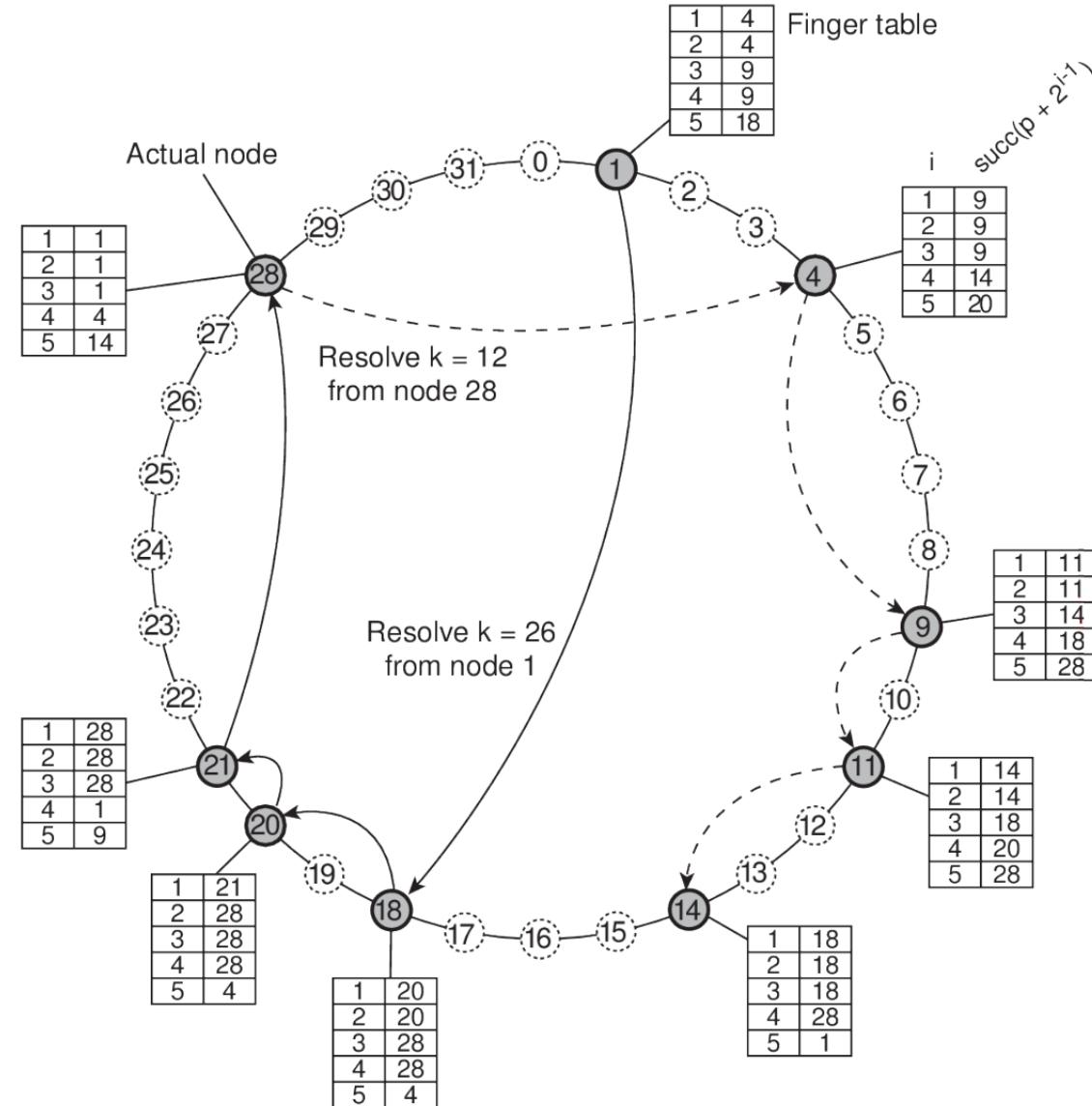


Протокол Chord



Stoica I. et al. Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications (2001)

Chord: поиск



Chord: другие функции

- Добавление узла
 - Генерация ключа для нового узла
 - Инициализация predecessor и finger table
 - Уведомление других узлов, перенос данных
- Поддержание данных об узлах в актуальном состоянии (стабилизация)
 - Опрос successor о его predecessor, уведомление successor о новом predecessor
 - Обновление finger table путем отправки запросов
 - Проверка доступности predecessor
- Обработка отказов узлов
 - Поддержание списка r ближайших successor
 - Репликация данных на k следующих узлах

Chord: свойства

- Каждый узел хранит информацию о $O(\log N)$ других узлах
- Поиск по ключу требует $O(\log N)$ сообщений
- Перестройка при добавлении/удалении узла требует $O(\log^2 N)$ сообщений

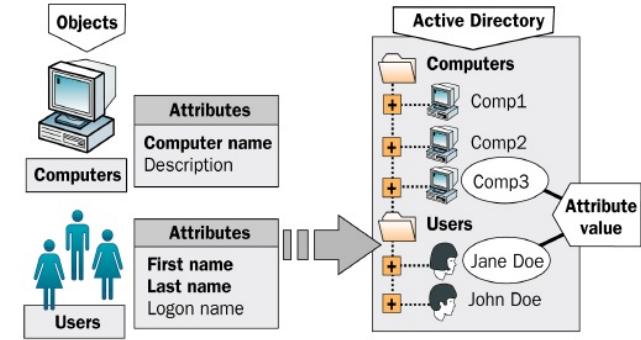
DNS vs DHT?

- В чем преимущества и недостатки каждого подхода?
- Для каких целей какой подход лучше?

Pappas V. et al. A comparative study of the DNS design with DHT-based alternatives
(2006)

Именование на основе атрибутов

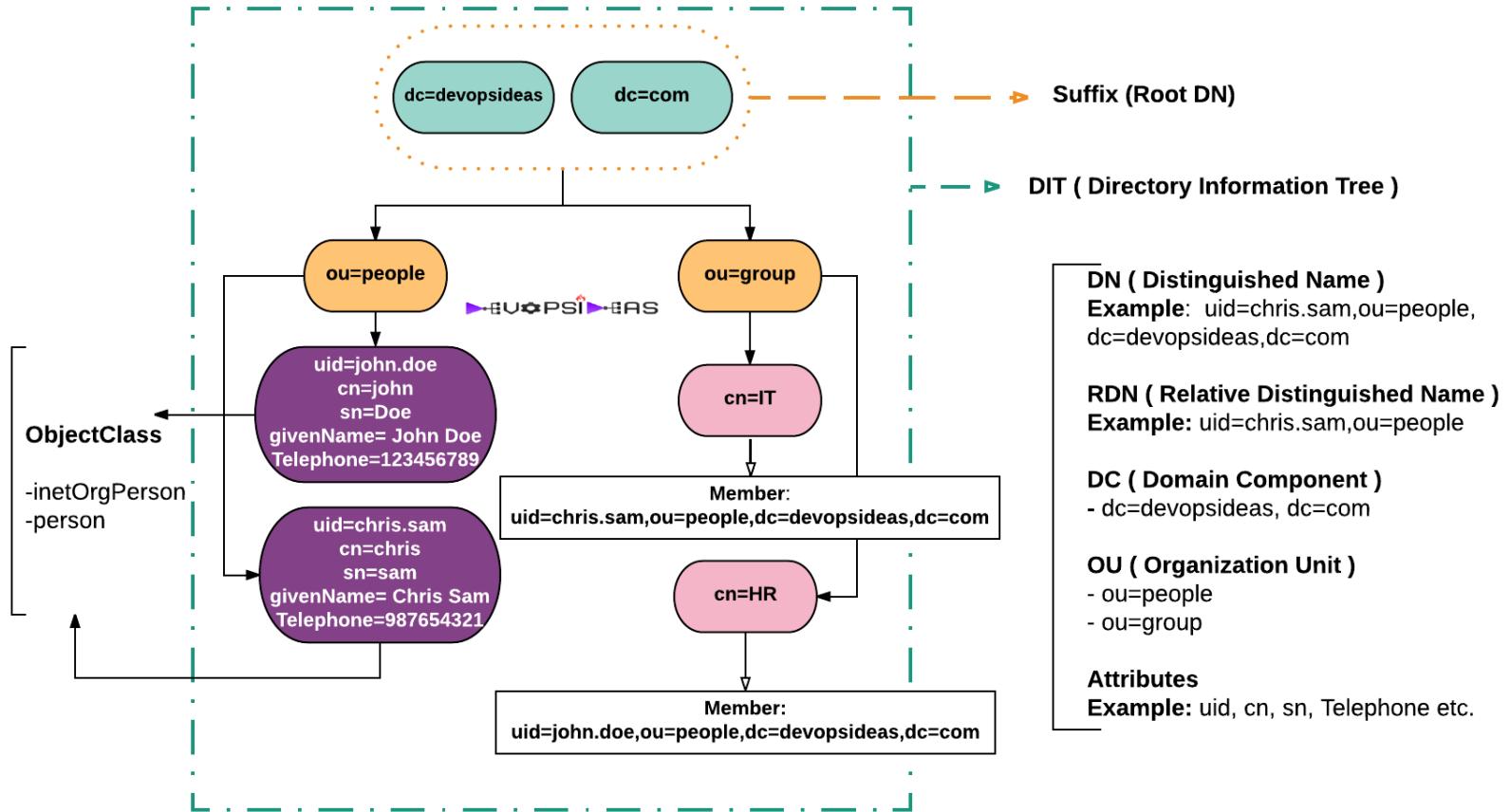
- С объектом связаны пары (*атрибут, значение*)
- Позволяет организовать гибкий поиск на основе значений атрибутов
- Реализацию подобного сервиса часто называют *служба каталогов (directory service)*
- Как реализовать поиск по атрибутам, если данные распределены?



Screenshot of a music player interface showing search filters for a song by Bob Dylan. The filters include:

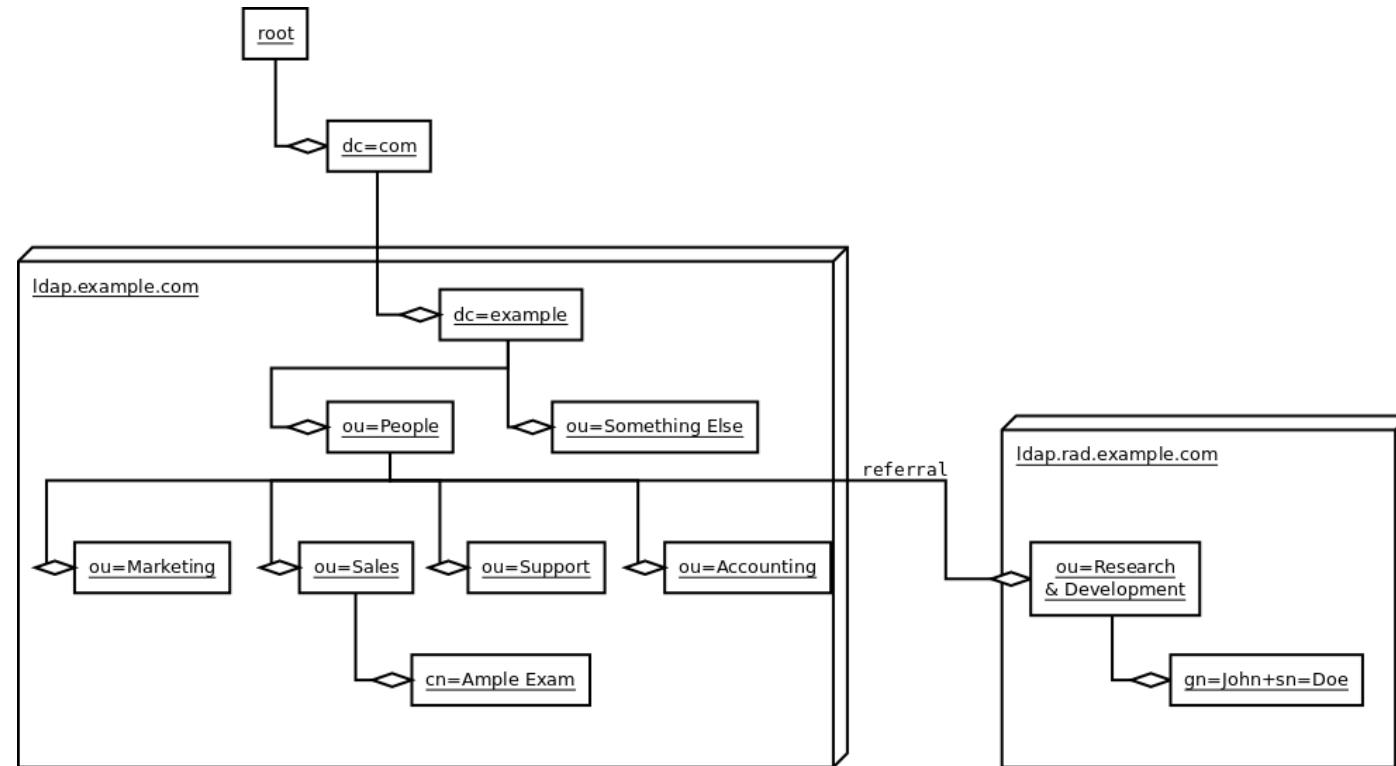
- song: Knockin' on Heaven's Door
- artist: Bob Dylan
- album: Bob Dylan's Greatest Hits, Vol. 3
- album artist: Bob Dylan
- composer: Bob Dylan
- Show composer in all views
- grouping: (empty)
- genre: Rock
- year: 1989

Иерархическая реализация



Lightweight Directory Access Protocol (LDAP)

Распределение дерева между серверами



Децентрализованные реализации?

- Поиск файла по атрибутам в peer-to-peer системе
- См. литературу

Литература

- van Steen M., Tanenbaum A.S. Distributed Systems: Principles and Paradigms. Pearson, 2017. (глава 5)
- Kurose J., Ross K. Computer Networking: A Top-Down Approach (раздел 2.4)
- Peterson L., Davie B. Computer Networks: A Systems Approach (про DNS и P2P)

Дополнительно

- Couloris G.F. et al. Distributed Systems: Concepts and Design. Pearson, 2011
(главы 13 и 10)
- Упомянутые статьи