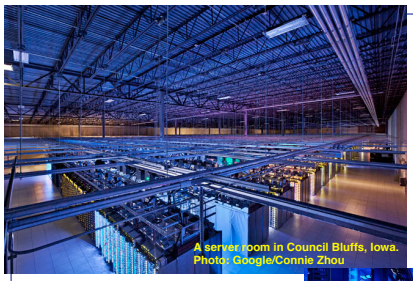
Υποδομές για Υπηρεσίες ΠΠΠ γιγαντιαίας κλίμακας

(Giant-scale infrastructures)

Παραδείγματα

- Web portals (Yahoo, CNN,…)
- e-Commerce (eBay, Amazon, AliBaba…)
- Search Engines (Google, Bing,…)
- Messaging and Communication (WhatsApp, iCQ, Slack…)
- Geoservices (Waze, GoogleMaps,…)
- Social Networks (Facebook, Twitter,…)



A server room in Council Bluffs, Iowa. Photo: Google/Connie Zhou

Clusters in Facebook

Clusters [συστοιχίες Η/Υ]

- Collections of commodity servers that work together on a single problem, offering as main advantages:

Γιατί συστοιχίες;

- Absolute scalability. A successful network service must scale to support a substantial fraction of the world's population.
- Cost and performance
  - no alternative to clusters can match the required scale
  - hardware cost is typically dwarfed by bandwidth and operational costs.
- Independent components. Users expect 24-hour service from systems that consist of thousands of hardware and software components. Transient hardware failures and software faults due to rapid system evolution are inevitable, but clusters simplify the problem by providing (largely) independent faults.
- Incremental scalability. Clusters should allow for scaling services as needed to account for the uncertainty and expense of growing a service.
  - three-year depreciation lifetime (χρόνος απόσβεσης) - should generally be replaced only when they no longer justify their rack space compared to new nodes.
  - A unit of rack space should quadruple in computing power over three years [Moore's law]. Actual increases appear to be faster due to improvements in packaging and disk size.

Βασικές υποθέσεις υπηρεσιών κλίμακας

- Service provider has limited control over the clients and the IP network
- Queries drive the service [e.g. HTTP get]
- Read-only queries greatly outnumber updates (queries that affect the persistent data store)

Αρχιτεκτονικό Μοντέλο

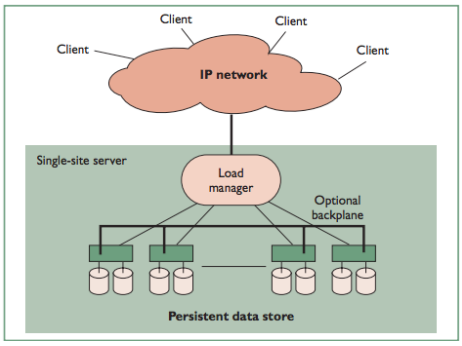


Figure 1. The basic model for giant-scale services. Clients connect via the Internet and then go through a load manager that hides down nodes and balances traffic.

Πηγή: E. Brewer, IC 2001

Πλεονεκτήματα Μοντέλου

- Access anywhere, anytime. A ubiquitous infrastructure facilitates access from home, work, airport, and so on.
- Availability via multiple devices. Infrastructure handles most of the processing => users can access services from "thin clients", which can offer far more functionality for a given cost and battery life.
- Groupware support. Centralizing data from many users allows service providers to offer group-based applications (calendars, teleconferencing systems, group-management systems).
- Lower overall cost. Infrastructure services have a fundamental cost advantage over designs based on stand-alone devices: can be multiplexed across active users; end-user devices have very low utilization (less than 4 percent), while infrastructure resources often reach 80 percent utilisation (moving anything from the device to the infrastructure effectively improves efficiency by a factor of 20); centralizing the administrative burden and simplifying end devices also reduce overall cost.
- Simplified service updates. Most powerful long-term advantage is the ability to upgrade existing services or offer new services without the physical distribution required by traditional applications and devices.

Βασικά Δομοστοιχεία Μοντέλου

- Clients (πελάτες), such as Web browsers, standalone email readers, or even programs that use XML and SOAP initiate the queries to the services.
- The best-effort IP network, whether the public Internet or a private network such as an intranet, provides access to the service.
- The load manager (εξισορροπητής φορτίου) provides a level of indirection between the service's external name and the servers' physical names (IP addresses) to preserve the external name's availability in the presence of server faults. The load manager balances load among active servers. Traffic might flow through proxies or firewalls before the load manager.
- Servers (εξυπηρετητές/διακομιστές/διαθέτες) are the system's workers, combining CPU, memory, and disks into an easy-to-replicate unit.
- The persistent data store (βάση δεδομένων) is a **replicated** or **partitioned** "database" that is spread across the servers' disks. It might also include network attached storage such as external DBMSs or systems that use RAID storage.
- Many services also use a backplane. This optional system-area-network handles inter server traffic such as redirecting client queries to the correct server or coherence traffic for the persistent data store.

Εξισορρόπηση φορτίου (load balancing)

- Στόχος: ισορροπημένος επιμερισμός εισερχόμενου φορτίου στους διαθέσιμους εξυπηρετητές.
- Προσεγγίσεις:
  - Have DNS distribute different IP addresses for a single domain name among clients in a rotating fashion ("round-robin DNS")
  - Combination of:
    - custom"layer-4" switches that understand TCP and port numbers, and can make decisions based on this information
    - "front-end" nodes that act as service-specific "layer-7" (application layer) switches, understand HTTP requests and parse URLs at wire speed
  - Include clients in the load-management process (clients know about alternative servers and can switch to them if primary server disappears)

Handling Failure (διαχείριση σφαλμάτων)

- Load-balancing switches:
  - Support hot failover to avoid the obvious single point of failure
    - Hot failover: the ability for one switch to take over for another automatically
  - Can handle very high throughputs
  - Detect down nodes automatically, usually by monitoring open TCP connections, and thus dynamically isolate down nodes from clients quite well

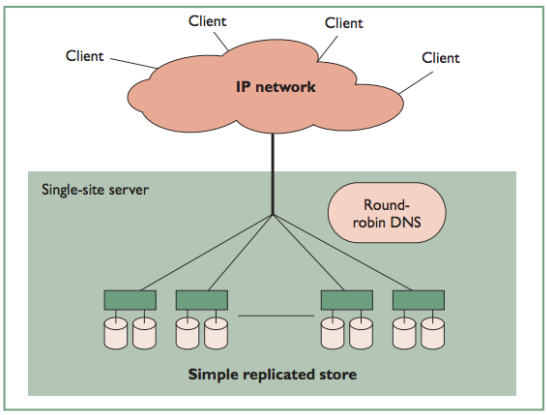
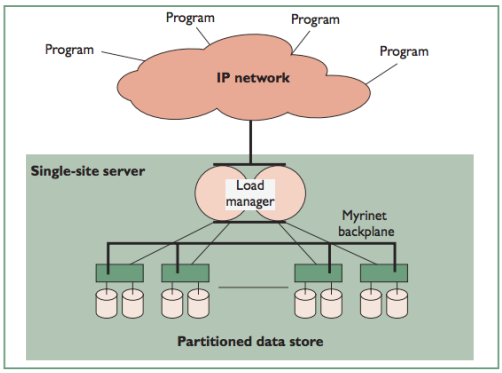


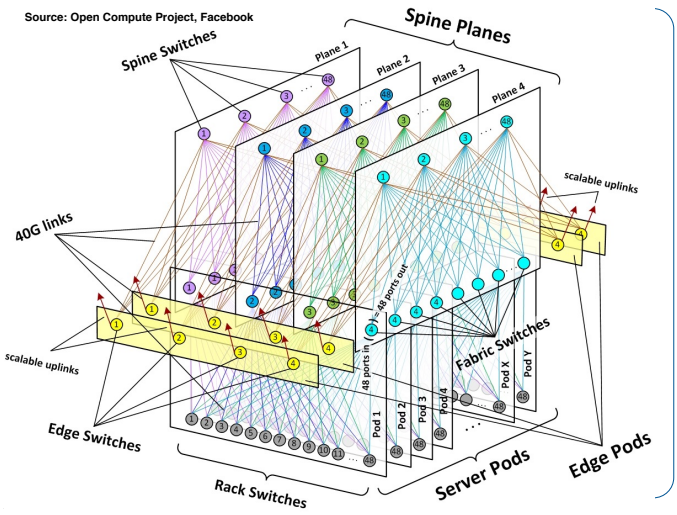
Figure 2. A simple Web farm. Round-robin DNS assigns different servers to different clients to achieve simple load balancing. Persistent data is fully replicated and thus all nodes are identical and can handle all queries.

Πηγή: E. Brewer, IEEE IC 2001



Πηγή: E. Brewer, IEEE IC 2001

Figure 3. Search engine cluster. The service provides support to other programs (Web servers) rather than directly to end users. These programs connect via layer-4 switches that balance load and hide faults. Persistent data is partitioned across the servers, which increases aggregate capacity but implies there is some data loss when a server is down. A backplane allows all nodes to access all data.



High Availability (υψηλή διαθεσιμότητα)

- Major driving requirement behind giant-scale system design, in the presence of component failures, natural disasters, and also constantly evolving features and unpredictable growth.
- Availability Metrics (μετρικές):
- **uptime** (λειτουργικός χρόνος) = (MTBF – MTTR)/MTBF
  - Fraction of time a site is handling traffic
  - Typically measured in nines - traditional infrastructure systems aim for 4 to 5 nines (0.9999 to 0.99999)
- **yield** (απόδοση) = queries completed/queries offered
  - Fraction of queries that are completed successfully
- **harvest** (συγκομιδή) = data available/complete data
  - in systems based on queries, we can measure *query completeness*— how much of the database is reflected in the answer
  - this can be extended to features supported by a service

DQ (data per query) Principle

Data per query x **queries per second -> constant**

- Principle rather than a literal truth: the system's overall capacity tends to have a particular physical bottleneck (στενωπός), such as total I/O bandwidth or total seeks per second
- The DQ value is the *total amount of data that has to be moved per second on average*
  - it is thus bounded by the underlying physical limitation
  - at the high utilization level typical of giant-scale systems, the DQ value approaches this limitation
- The DQ value is measurable and tunable

Measuring and Tuning DQ

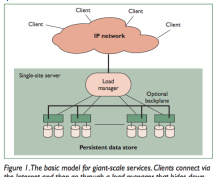
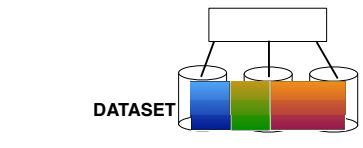


Figure 1. The basic model for giant-scale services. Clients connect via the Internet and then go through a load manager that hides down nodes and balances traffic.

http://www.seleniumhq.org/

- Πώς μετράμε το DQ μιας υποδομής;
  - Define target workload (φορτίο)
  - Use a load generator to measure a given combination of hardware, software and db size against this workload
  - Given the metric and the load generator, it is easy to measure relative impact of faults
- Πώς βελτιώνουμε το DQ;
  - DQ scales linearly with the number of nodes
  - We can translate future traffic predictions into future DQ requirements and this into hardware and software target - convert traffic predictions into capacity planning decisions

Partitioning (διαμελισμός-διαχωρισμός)



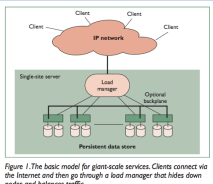
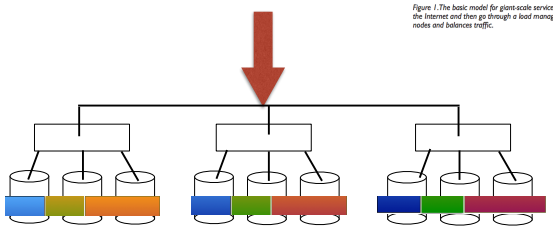
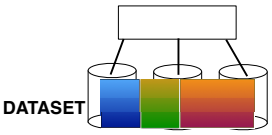


Figure 1. The basic model for giant-scale services. Clients connect via the Internet and then go through a load manager that hides down nodes and balances traffic.



Partitioning (διαμελισμός-διαχωρισμός)



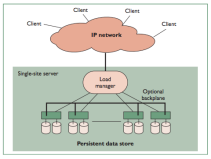
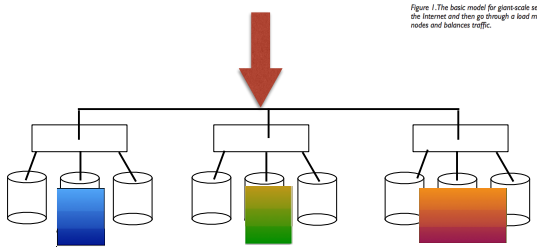


Figure 1.The basic model for giant-scale services. Clients connect via the Internet and then go through a load manager that hides down nodes and balances traffic.



Partitioning (διαμελισμός-διαχωρισμός)

- Persistent data is partitioned across the servers, which increases aggregate capacity

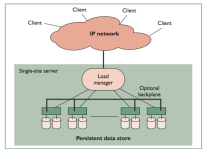
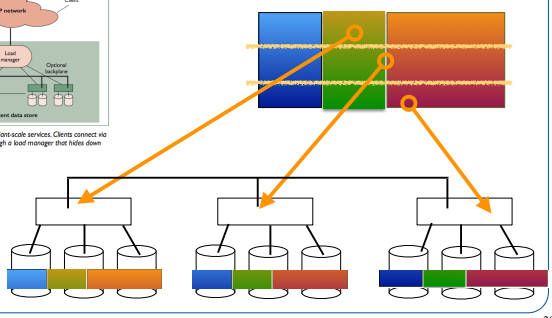


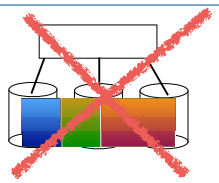
Figure 1.The basic model for giant-scale services. Clients connect via the Internet and then go through a load manager that hides down nodes and balances traffic.

DATASET



Partitioning and Faults

- What is the effect of failure on:
  - Yield? (απόδοση)
  - Harvest? (συγκομιδή)



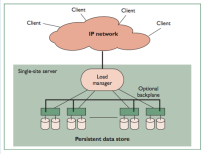
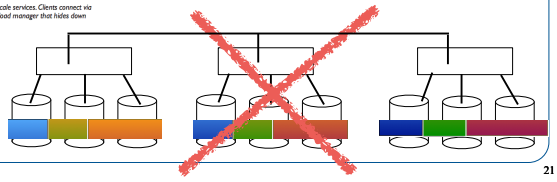
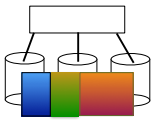


Figure 1.The basic model for giant-scale services. Clients connect via the Internet and then go through a load manager that hides down nodes and balances traffic.



Replication (αναπαραγωγή)



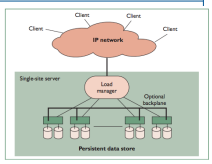
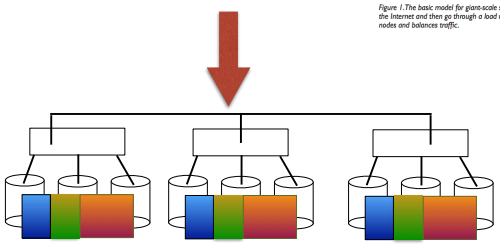


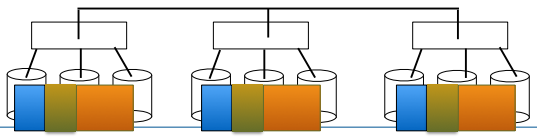
Figure 1.The basic model for giant-scale services. Clients connect via the Internet and then go through a load manager that hides down nodes and balances traffic.



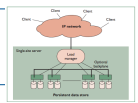
Replication (αντιγραφή-αναπαραγωγή)

- Used to increase performance and availability and to improve fault tolerance – provides multiple consistent copies of data in processes running in different computers.
- The traditional view of replication silently assumes that there is enough excess capacity to prevent faults from affecting yield.

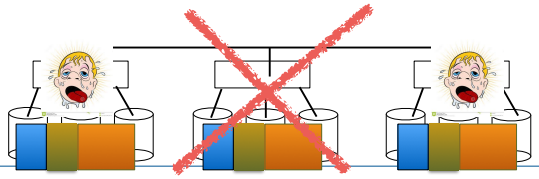
DATASET



Replication and faults

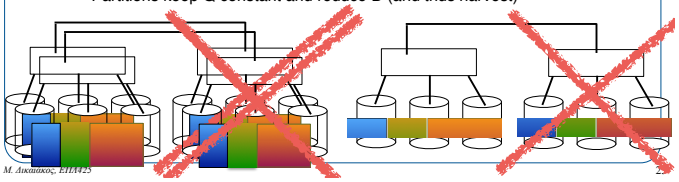


- What is the effect of failure on:
  - Yield? (απόδοση)
  - Harvest? (συγκομιδή)
- The traditional view of replication silently assumes that there is enough excess capacity to prevent faults from affecting yield.
- Load redirection problem: under faults, the remaining replicas have to handle the queries formerly handled by the failed nodes.
- Under high utilization, this is unrealistic.



Replication vs Partitioning

- Replication is a traditional technique for increasing availability
- Consider a *two-node cluster* that faces a fault *in one node*:
  - The replicated version maintains 100 percent harvest but drops to 50 percent yield
  - The partitioned version drops to 50 percent harvest but remains at 100 percent yield
  - Both versions have the same initial DQ value and lose 50 percent of it under one fault:
    - Replicas maintain D (data per query) and reduce Q (queries per sec - yield)
    - Partitions keep Q constant and reduce D (and thus harvest)



Replication vs Portioning

- We can influence whether faults impact yield, harvest, or both:
- Replicated systems tend to map faults to reduced capacity (and to yield at high utilizations)
- Partitioned systems tend to map faults to reduced harvest, as parts of the database temporarily disappear, but the capacity in queries per second remains the same

Key insights

- The DQ constant is independent of whether the database is replicated or partitioned. WHY?
  - Exception: replication requires more DQ points than partitioning for heavy write traffic, which was rare in giant-scale systems (not anymore - see Facebook). WHY?
- Easier to grow systems via replication than by repartitioning onto more nodes
- We can vary the replication according to the data's importance and control which data is lost in the presence of a fault.
- We can exploit randomisation to make our lost harvest a random subset of the data.

Τα ονόματα στο ΠΠΠ

Domain Name System (DNS)

- a set of servers that map written names to IP addresses
  - Example: www.cs.washington.edu → 128.208.3.88
- many systems maintain a local cache called a hosts file
  - Windows: C:\Windows\system32\drivers\etc\hosts
  - Mac: /private/etc/hosts
  - Linux: /etc/hosts

DNS: Outline

- Computer science concepts underlying DNS
  - Indirection: names in place of addresses
  - Hierarchy: in names, addresses, and servers
  - Caching: of mappings from names to/from addresses
- Inner-workings of DNS
  - DNS resolvers and servers
  - Iterative and recursive queries
  - TTL-based caching
- Web and DNS
  - Influence of DNS queries on Web performance
  - Server selection and load balancing

Host Names vs. IP addresses

- Host names
  - Mnemonic name appreciated by humans
  - Variable length, alpha-numeric characters
  - Provide little (if any) information about location
  - Examples: www.cnn.com and ftp.eurocom.fr
- IP addresses
  - Numerical address appreciated by routers
  - Fixed length, binary number
  - Hierarchical, related to host location
  - Examples: 64.236.16.20 and 193.30.227.161

Separating Naming and Addressing

- Names are easier to remember
  - www.cnn.com vs. 64.236.16.20
- Addresses can change underneath
  - Move www.cnn.com to 64.236.16.20
  - E.g., renumbering when changing providers
- Name could map to multiple IP addresses
  - www.cnn.com to multiple replicas of the Web site
- Map to different addresses in different places
  - Address of a nearby copy of the Web site
  - E.g., to reduce latency, or return different content
- Multiple names for the same address
  - E.g., aliases like ee.mit.edu and cs.mit.edu

Strawman Solution: Local File

- Original name to address mapping
  - Flat namespace
  - /etc/hosts
  - SRI kept main copy
  - Downloaded regularly
- Count of hosts was increasing: moving from a machine per domain to machine per user
  - Many more downloads
  - Many more updates

Strawman Solution #2: Central Server

- Central server
  - One place where all mappings are stored
  - All queries go to the central server
- Many practical problems
  - Single point of failure
  - High traffic volume
  - Distant centralized database
  - Single point of update
  - Does not scale

Need a distributed, hierarchical collection of servers

Domain Name System (DNS)

- Properties of DNS
  - Hierarchical name space divided into **zones**
  - Distributed over a collection of **DNS servers**
- Hierarchy of DNS servers
  - Root servers
  - Top-level domain (TLD) servers
  - Authoritative DNS servers
- Performing the translations
  - Local DNS servers
  - Resolver software

DNS - The Internet Domain Name System

- A distributed naming database
- Name structure reflects administrative structure of the Internet
- Rapidly resolves domain names to IP addresses
  - exploits caching heavily
  - typical query time ~100 milliseconds
- Scales to millions of computers
  - partitioned database
  - caching
- Resilient to failure of a server
  - replication

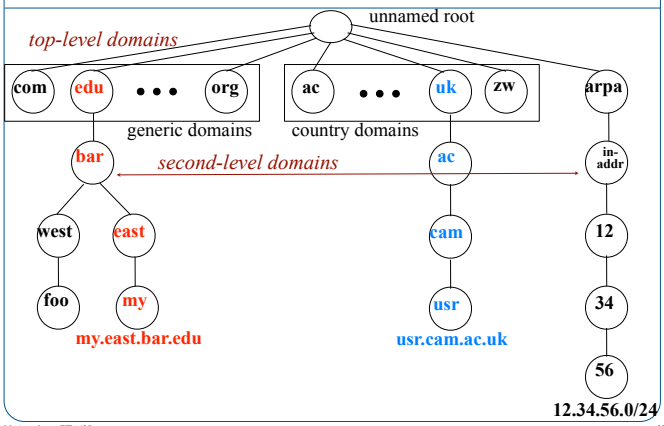
Αρχιτεκτονική DNS

- **Ζώνες** (zones): μη επικαλυπτόμενα τμήματα ενός χώρου ονομάτων, το καθένα εκ των οποίων υποστηρίζεται από διαφορετικό εξυπηρετητή
  - υποδένδρα της ιεραρχίας του DNS, τα οποία ανήκουν σε διαφορετική διοικητική αρχή.
  - Κάθε ζώνη έχει συνήθως έναν πρωταρχικό εξυπηρετητή και περισσότερους δευτερεύοντες εξυπηρετητές, οι οποίοι μπορούν να υποκαταστήσουν τον πρωταρχικό σε περίπτωση βλάβης.
  - Μεγάλοι οργανισμοί μπορούν να οργανώνουν τα πεδία τους σε περισσότερες της μιας ζώνες.
- Η αποδοτική απεικόνιση διευθύνσεων IP σε ονόματα κόμβων (hostnames) προϋποθέτει μια διαφορετική ιεραρχία, βασιζόμενη σε διευθύνσεις IP.
  - Η ανάθεση των διευθύνσεων IP υποστηρίζεται από μητρώα (registries): APNIC (Ασία), ARIN (Β. Αμερική), RIPE NCCC (Ευρώπη), LACNIC, AFRINIC
- Η ανάθεση διευθύνσεων IP στα τρία μητρώα γίνεται από το **Internet Assigned Numbers Authority (IANA)**, που είναι τμήμα του μη κερδοσκοπικού οργανισμού **ICANN (Internet Corporation for Assigned Names and Numbers)**.
  - Με την δέσμευση ενός συνόλου διευθύνσεων IP από έναν οργανισμό, ο οργανισμός αυτός γίνεται υπεύθυνος για ένα τμήμα του ονοματοχώρου in-addr.arpa. Πρόκειται για μια ιεραρχία που βασίζεται στις οκτάδες των 32-μπιτων διευθύνσεων IP.

TLD and Authoritative DNS Servers

- Top-level domain (TLD) servers
  - Generic domains (e.g., com, org, edu)
  - Country domains (e.g., uk, fr, ca, jp)
  - Typically managed professionally
    - Network Solutions maintains servers for “com”
    - Educause maintains servers for “edu”
- Authoritative DNS servers
  - Provide public records for hosts at an organization
  - For the organization’s servers (e.g., Web and mail)
  - Can be maintained locally or by a service provider

Distributed Hierarchical Database

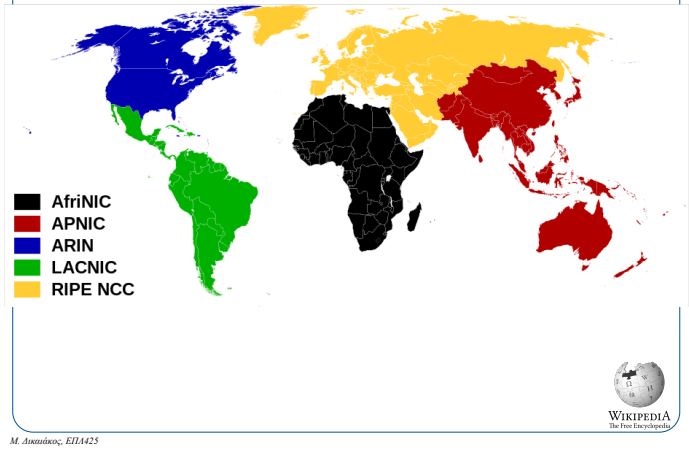


The role of root servers

Notice that:

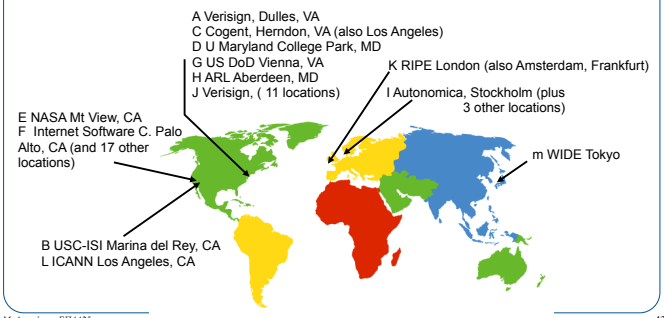
- root servers only know who you need to ask next.
  - .com -> list of servers
  - .net -> list of servers
  - .ch -> list of servers
  - .ug -> list of servers
  - .br -> list of servers
- Caching of previous answers means there is less need to query the root servers after the first question.

http://www.root-servers.org/presentations/wsis.pdf



DNS Root Servers

- 13 root servers (see http://www.root-servers.org/)
- Labeled A through M



Uniform Resource Locator (URL)

- RFC1738
- an identifier (αναγνωριστικό / όνομα) for the location of a document on a web site
- a basic URL:

http:// www.aw-bc.com / info/regesstepp/index.html

protocol host path

- upon entering this URL into the browser, it would:
  - **ask the DNS server** for the IP address of www.aw-bc.com
  - connect to that IP address at port 80 (**open socket**)
  - ask the server to **GET** /info/regesstepp/index.html
  - **display** the resulting page on the screen

Πιο σύνθετα URL

- **anchor (άγκυρα)**: jumps to a given section of a web page
  http://www.textpad.com/download/index.html**#downloads**
  - fetches index.html then jumps down to part of the page labeled downloads
- **port (θύρα)**: for web servers on ports other than default 80
  http://www.cs.washington.edu**:8080**/secret/money.txt
- **query string (αλφαριθμητικό επερώτησης)**: a set of parameters passed to a web program
  http://www.google.com/search**?q=miserable+failure&start=10**
  - parameter q is set to "miserable+failure"
  - parameter start is set to 10

Στοιχεία Συντακτικού URL

- Σχήμα (scheme, followed by a colon)
  - http:, ftp:, mailto:, telnet:
- // (Double slash (only for http, ftp, wais, gopher)
- Όνομα πεδίου (Internet domain name) ή διεύθυνση IP
- Αριθμός θύρας προαιρετικά (port number) - eg. www.cs.ucy.ac.cy:8080
  - Default ports:
    - HTTP is 80
    - FTP is 21
    - SMTP is 25
    - IMAP is 143
- Μονοπάτι (path) - e.g. /users/mdd