ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Information-Centric Networks

Section # 5.3: Content Distribution

Instructor: George Xylomenos

Department: Informatics

European Union
European Social Fund

OPERATIONAL PROGRAMME
EDUCATION AND LIFELONG LEARNING
investing in knowledge society
MINISTRY OF EDUCATION, LIFELONG LEARNING AND RELIGIOUS AFFAIRS
MANAGING AUTHORITY

NSRF 2007-2013
programme for development
EUROPEAN SOCIAL FUND

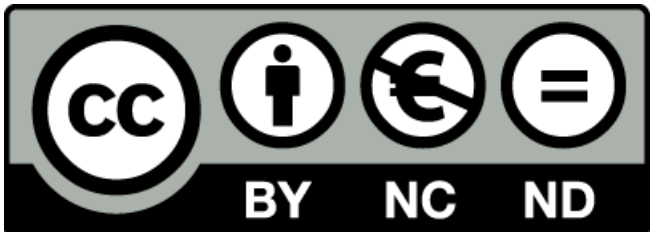
Co- financed by Greece and the European Union

Funding

- These educational materials have been developed as part of the instructors educational tasks.
- The **“Athens University of Economics and Business Open Courses”** project only funded the reformatting of these educational materials.
- The project is being implemented as part of the Operational Program “Instruction and Lifelong Learning” and is co-financed by the European Union (European Social Fund) and national funds.

Licencing

- These educational materials are subject to a Creative Commons License.



Week 5 / Paper 3

- Democratizing content publication with Coral
  - Michael J. Freedman, Eric Freudenthal, David Mazières
  - NSDI, USENIX, 2004
- Main point
  - CDNs are good if you can afford them
  - CoralCDN is a P2P based variant of CDNs
  - Anyone can offer a proxy, anyone can be a client or server
  - A simple change in the URL is all that is required
  - Incorporates DNS and proxy selection
  - Uses Sloppy DHTs to cluster content

Introduction

- Being popular is a mixed blessing on the Internet
  - Getting mentioned on slashdot can bring your site down
  - Popular commercial sites rely on server farms or CDNs
  - Other sites rely on mirroring or P2P file sharing
  - CoralCDN turns P2P into a CDN
- CoralCDN operation
  - Simply append ".nyud.net:8090" to a URL
  - CoralDNS transparently redirects to a participating server
  - Uses the Coral key/value indexing system
    - Tries to keep queries in the local area
    - Prevents traffic hot spots
  - A DHT based on clusters of well-connected machines
    - Reminiscent of hierarchical DHT systems

The Coral CDN

- CoralCDN components
  - The participating HTTP proxies that serve users
  - The DNS servers for “nyucd.net”
  - The Coral indexing and clustering middleware
- Usage models
  - Publisher: “coralize” selected URLs in web pages
  - Third-parties: “coralize” posted URLs
  - Users: “coralize” URLs for any sites
  - Coralize means add “nyud.net:8090” in the URL
    - Queries are redirected to the Coral name servers
  - All relative links are automatically coralized

System overview

- What happens when a client accesses a coralized URL?
  - The client sends www.x.com.nyud.net to resolver
  - The resolver asks one of the nyud.net nameservers
  - The Coral DNS server probes the client's location
  - The DNS server asks Coral for a nearby NS and/or proxy
  - The server replies with nearby (or random) servers
  - The resolver returns the address of an HTTP proxy
  - The client asks the proxy for the coralized URL
  - The proxy looks up the URL in Coral
  - The proxy fetches the object from the other proxy or the origin
  - The proxy stores the object and returns it to the client
  - The proxy stores a reference to itself in Coral

The Coral indexing abstraction

- Coral is a Sloppy DHT (DSHT)
  - Each key can be associated with multiple values
  - Each Coral node belongs to several clusters
  - Each cluster is characterized by a maximum desired RTT
    - This is the cluster diameter
  - The clusters are organized into levels
    - Normally three with RTTs of 20 msec, 60 msec and infinity
  - Each node participates in a cluster at each level
  - Queries start from local clusters and proceed to global ones
  - Coral interface
    - Put(key,value,ttl,[levels]): can limit TTL and levels to insert
    - Get(key,[levels]): can limit levels to search
    - Nodes(level,count,[target],[services]): return nodes at desired level
    - Levels(): number of levels and their RTTs

The Coral DNS server

- The Coral DNS server is process dnssrv
  - Returns proxies close to a client
  - Serves domain names ending http.L2.L1.L0.nyucd.net
    - Redirected from nyud.net with a DNAME record
  - Uses resolver's address to determine client location
    - Tries to match resolver with a cluster of the highest level possible
    - Uses the nodes function with the target parameter for this
  - Returns two sets of data
    - Nearby DNS servers (to be used for further name queries)
    - Nearby HTTP proxies (to be used for actual data downloads)
  - The DNS servers returned are as specific as possible
    - A faraway DNS server will say it is authoritative for L0
    - A closer one will say it is authoritative for L1 and even L2

The Coral HTTP proxy

- A proxy using either CoralCDN or origin servers
  - Assumes origin servers that are not well provisioned
  - Always tries to use local proxies instead
  - Caches pages fetched from other servers
  - Looks up unavailable web pages with Coral
  - Then the proxy inserts itself in the DSHT
    - It becomes a source for the page

Key-based routing

- Coral uses 160 bit keys (from hashing)
  - Keys are stored at nodes with keys close to themselves
  - Based on Kademlia which uses the XOR metric
  - Tries to approach the goal slowly
    - Unlike other DHTs which try to go as fast as possible
    - This way it allows flash crowd traffic to be spread
  - Keys are not always stored at the closest node
    - This is the sloppy part
  - A heuristic is used to stop a bit earlier than the closest node
    - Nodes that already store the key are preferred
    - Nodes that have already been asked a lot for the key are preferred
    - The closest node is always sure to also store the key
  - Retrieval returns multiple nodes (if they exist)
    - These can be used for parallel downloads

Hierarchical operations

- Coral uses levels of clusters to achieve locality
  - Retrievals indicate the starting and stopping levels
    - By default only the highest level cluster is used
    - If the key is not found, we search larger clusters
  - Insertions also always start at the highest level
    - The key is then also entered at larger clusters in some cases
- Nodes join the right clusters
  - A new node makes several queries to seed its tables
  - Nodes store hints towards their addresses in the DSHT
    - They store pointers to themselves under the keys of nearby routers
    - They also store pointers under their subnets and gateways
  - New nodes discover nearby nodes using these hints
  - Nodes periodically check whether they should change clusters

Performance evaluation

- CoralCDN deployment on 166 PlanetLab machines
  - Plus a web server behind a DSL link as the origin server
  - Each machine is also a client making requests
  - 2 orders of magnitude more load than what the DSL can take!
- Server load: a tiny amount of requests reaches the DSL
  - After the first few minutes, content is replicated
- Client latency: multi-level clustering really works
  - Much better than with a single level
- Clustering: proxies are automatically grouped
  - Excellent matching with topology
- Load balancing: requests are spread to nearby nodes
  - Even flash crowds can be easily handled

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

End of Section # 5.3

Course: Information-Centric Networks, **Section # 5.3: Content Distribution**

Instructor: George Xylomenos, **Department:** Informatics

European Union
European Social Fund
OPERATIONAL PROGRAMME EDUCATION AND LIFELONG LEARNING
investing in knowledge society
MINISTRY OF EDUCATION, LIFELONG LEARNING AND RELIGIOUS AFFAIRS
MANAGING AUTHORITY
NSRF 2007-2013
programme for development
EUROPEAN SOCIAL FUND
Co- financed by Greece and the European Union