CSC2231: DNS with DHTs

http://www.cs.toronto.edu/~stefan/courses/csc2231/05au

Stefan Saroiu

Department of Computer Science

University of Toronto

Administrivia

- **Next lecture:**
  - P2P churn
    - Understanding Availability Ranjita Bhagwan, Stefan Savage and Geoffrey Voelker. IPTPS 2003.
    - High Availability, Scalable Storage, Dynamic Peer Networks: Pick Two Charles Blake and Rodrigo Rodrigues. HOTOS 2003.

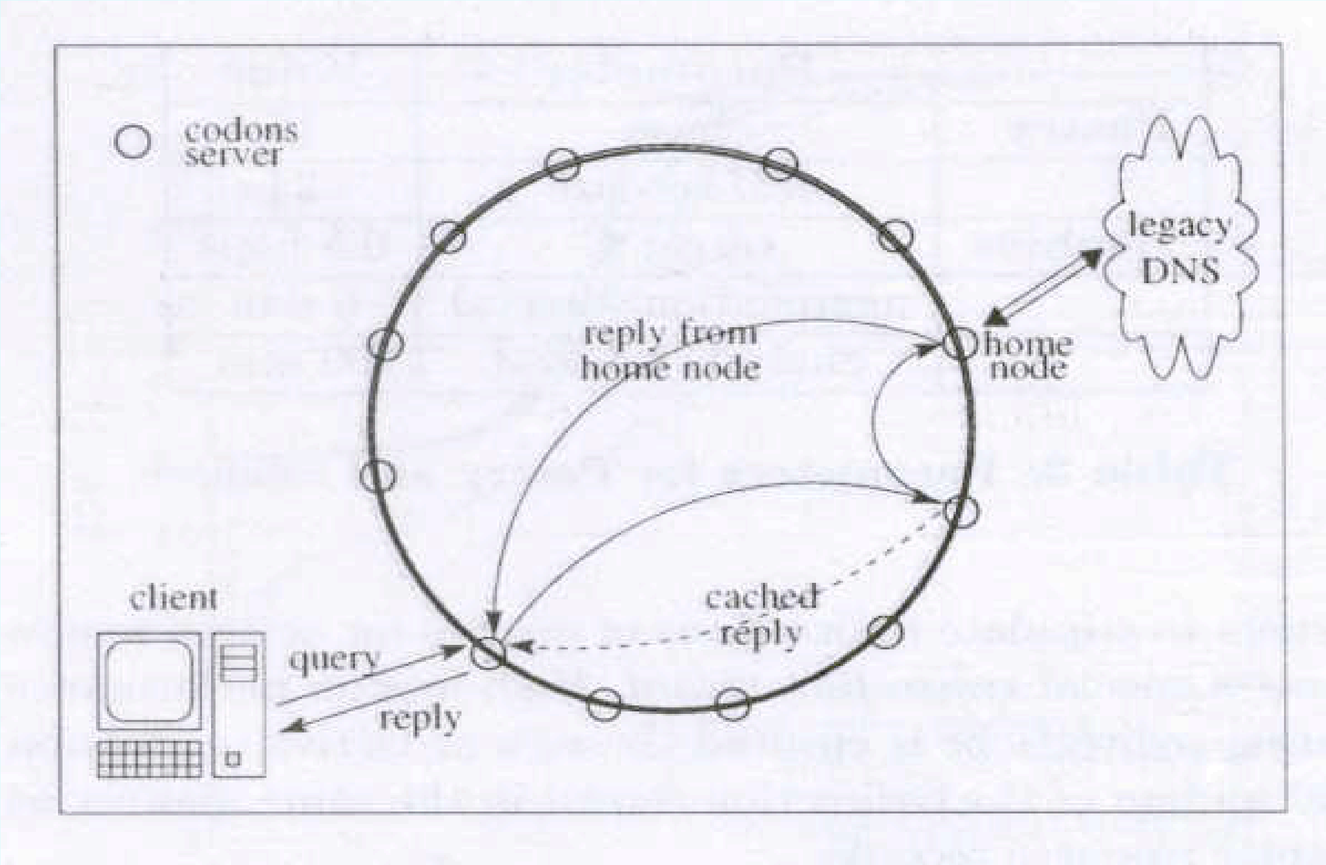
Limitations of current DNS

- **DNS problems:**
  - Every org must have DNS server:
    - 24/7 machine running with sysadmin
  - Hierarchical:
    - Poorly configured machine could affect entire sub-tree
    - Root DNS servers failures could be catastrophic
    - Root DNS servers are well-defined targets for attacks
  - Cache problems
    - Hard to propagate updates -- coherence problems
    - Short TTLs reduces hit rate

DHT-based DNS

- **DHTs:**
  - Scalable, self-organizing
  - Lack of hierarchy
    - Hard to attack a set of domain names
    - Handle flash-crowd effects well
    - No central points of failure
    - Network routes around failures
  - DNS servers --- mostly homogeneous
  - Can design backward-compatible DHT-based DNS

How would DHT-DNS work?



Where Beehive Improves

- **Uses controlled proactive caching**
- **Ex. Looking for 2101**
  - Takes 3 hops normally
- **Places copies of object at all nodes one prior to the home node.**
  - Reduces hops by one
  - Object is replicated on node 21
- **Can reduce to 1 hop by replicating it on node 2**

More Beehive

- **Important part is choosing what levels to replicate at**
- **Can set a constant to set average lookup performance (defined as C)**
- **Uses a function over Zipf-like distributions (similar to DNS traffic) to find C**
  - Must know the popularity distribution a priori

Security

- **Attack: Prevent spoofing of bindings**
- **Idea: use signatures**
  - www.cnn.com, A is signed with key K
  - www.cnn.com, K is signed with key K’
  - cnn.com, K’ is signed with key K’’
  - .com, K’’ is signed with master key M
- **If you trust M**
  - You trust K’’, then K’, then K, then A
- **This signature-based idea is orthogonal to whether DNS architecture is hierarchical or DHT-based**

Are we done?

Problems

- **Network outages are poorly handled**
- **Certain functionality is lost**
- **Solving the wrong problem**
- **Performance improvements are not due to DHTs**
  - But rather to heavy replication

Network Outages

- **Scenario: organization disconnects from the Internet**
  - Very common scenario in practice

- **Old DNS:**
  - Can still resolve local names
  - Can't resolve global names
  - External hosts can't resolve local names
- **P2P DNS:**
  - Cannot resolve local names
  - Can resolve some global names (but not connect)
  - External hosts can resolve local names (but not connect)

Functionality

- **Hard to support dynamically-generated records**
- **No support for "ANY" queries**
- **No server-side load balancing/proximity routing**
  - Akamai?

- **Possible solutions:**
  - Peers assume client-side functionality
    - Bad idea (+ ugly!)

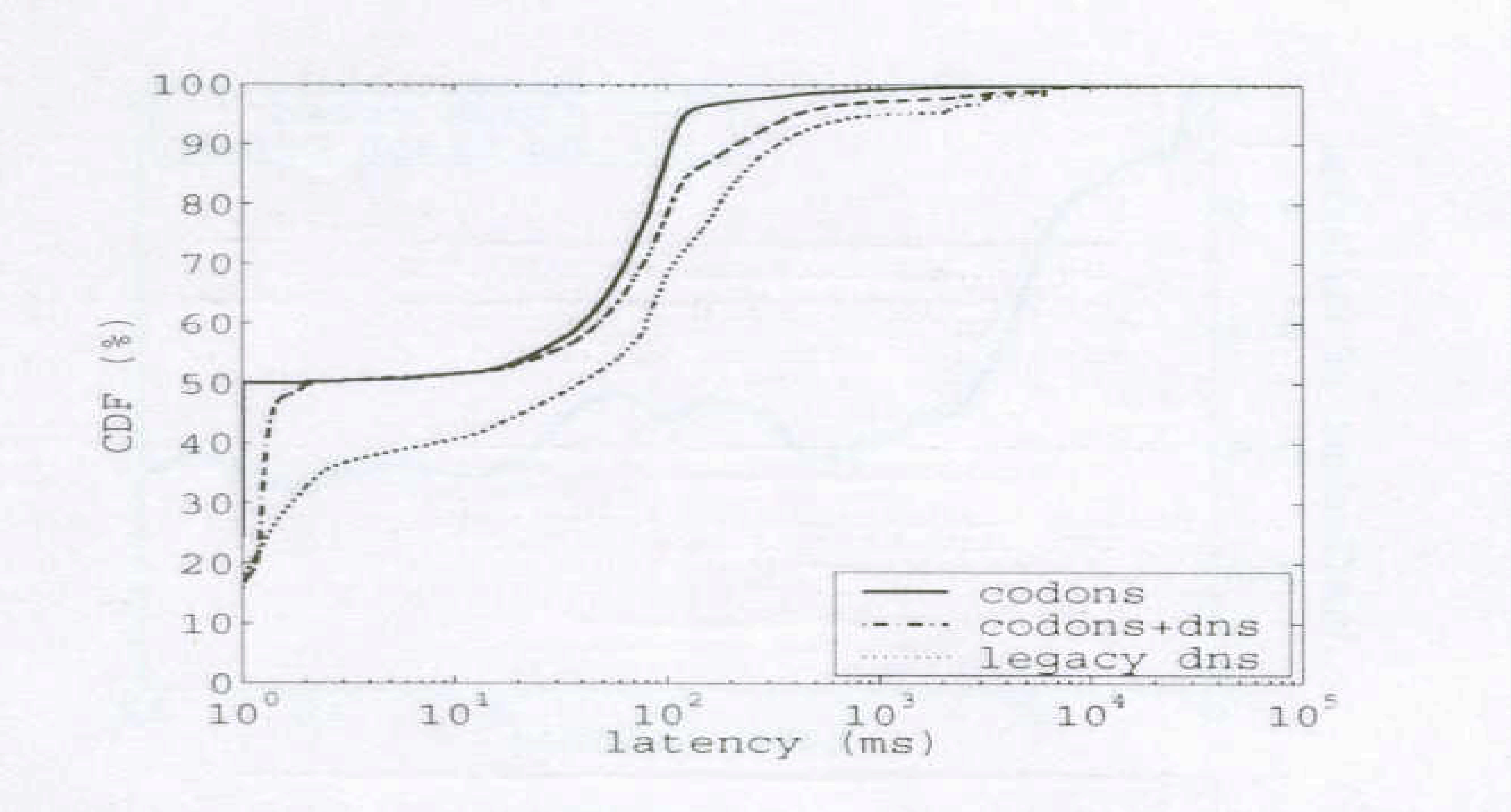
Administration

- **Common problem:**
  - Implementation errors
  - 9 out of 13 problems with DNS listed in O'Reilly are software deficiencies
- **Fixing software/configurations**
  - Sounds like an important problem
- **Changing system's architecture solves the wrong problem**

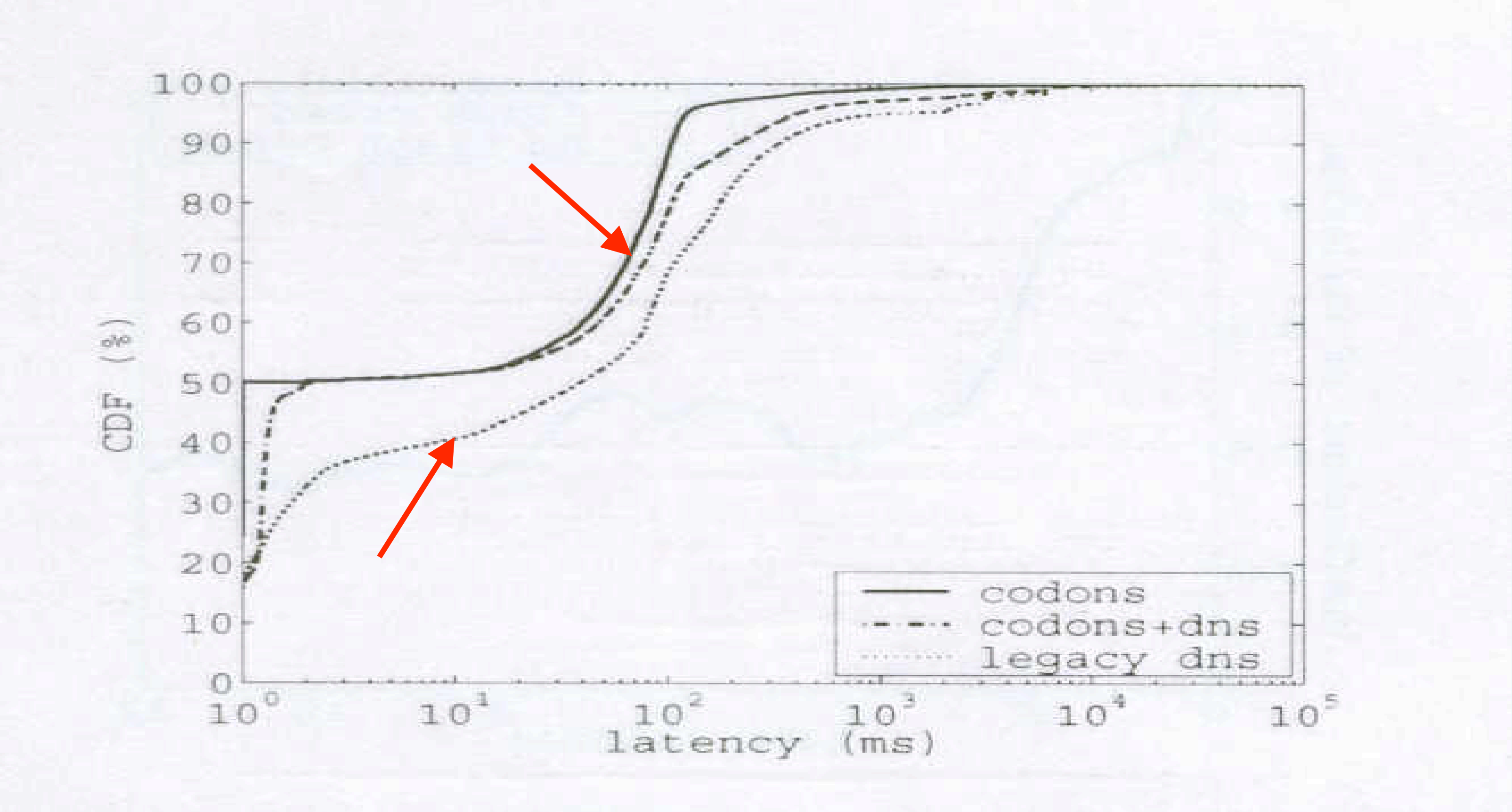
Administration (2)

- **Don’t have to run 24/7 servers**
  - But need to trust others for my own names
  - Where will we point the finger when something goes wrong?

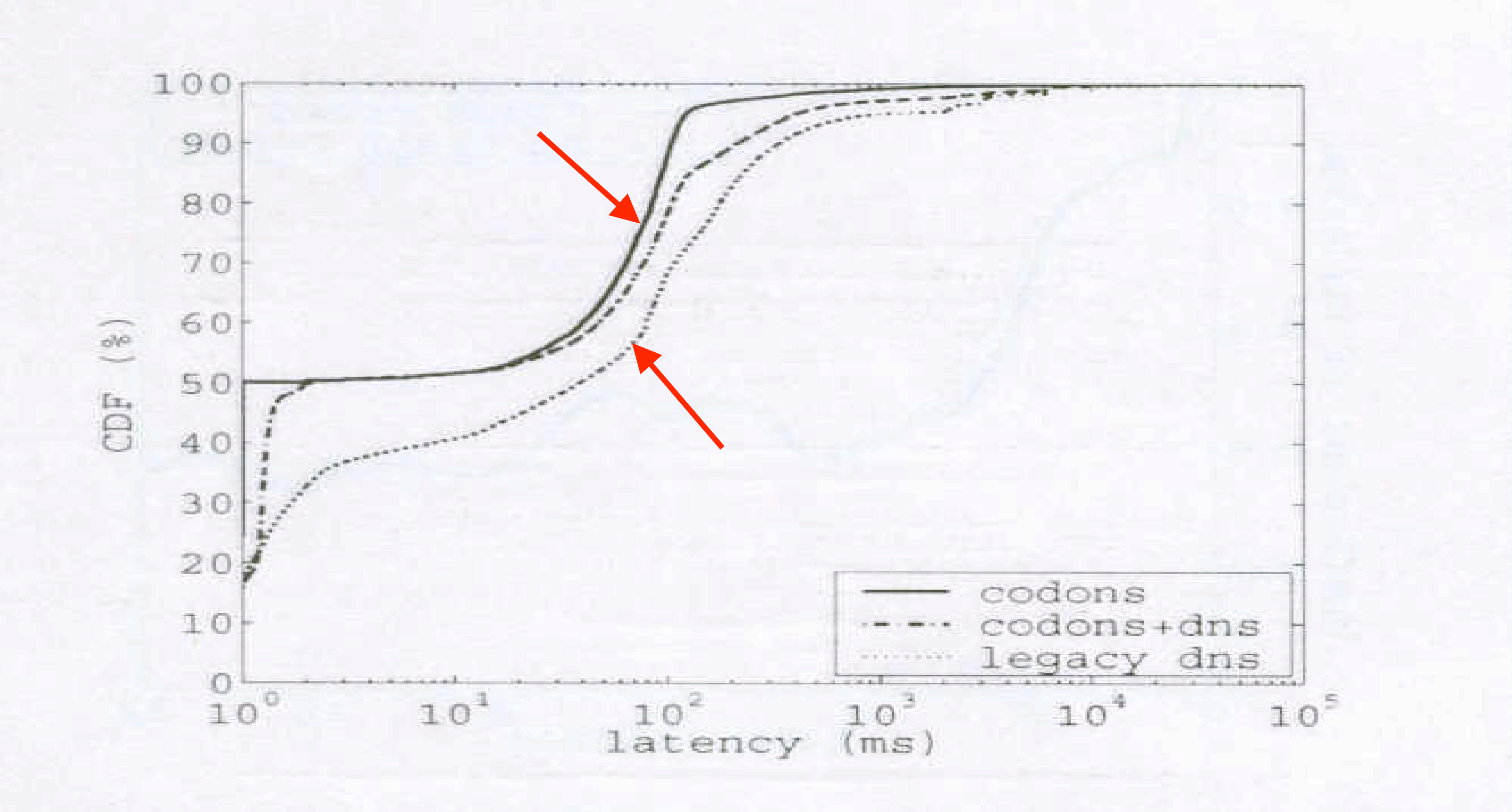
Performance



Performance



Performance



Alternate Design

- **Replication seems to have helped a lot!**
  - the case for pushing DNS!

Using the back of the envelope

- **There are 76.9 million domains registered**
  - Including generic TLDs and country-code TLDs
  - Compressed file with all info -- 7.5GB
- **About 20,000 AS's in the world**
  - Suppose each NS serves other 3 NS's (23 GB pushed)
  - Build delivery tree of depth 10 roughly
- **Push updates daily**
  - About 760 KBytes / hour
  - About 850 Kbps upload to three peers
- **A lot of changes are for the same bindings**
  - 87% of domains do not change at all

Advantages of pushing DNS

- **Great latency performance!**
- **Akamai still works**
- **Backward-compatible with old DNS**

- **We are only adding prefetching to DNS**
  - Improve performance with affecting the systems' architecture

- **Idea for M.Sc. project:**
  - build push-based DNS!

Discussion

- **Does it make sense to have so many different name systems?**
  - DNS names (DNS: names to IP addresses)
  - Peer IDs for P2P and DHTs (P2P system)
  - File names (FS: file names to i-nodes)
  - E-mail addresses (LDAP)
  - Chat Names (Chat Directory)
    - Dialing Skype names

Discussion

- **What if we had one large address space?**
  - 10^{81} atoms in the universe
  - 800 bits can identify any atom in the universe
- **Design name service to bind names to 1024 bit addresses**
  - Should we make it hierarchical?
    - e.g., decompose 1024 bits into:
      - IP address + disk # + partition # + block ID + …