Architecture of The Internet Archive

Elliot Jaffe, Scott Kirkpatrick
SYSTOR ‘09
May 6, 2009 -- Haifa, Israel
Outline

• 10 years of the Internet Archive
• Architecture
• Performance
• Adding a Cache
• Conclusions
Disclaimer

• The authors have never worked in a technical capacity at the Internet Archive and are not privy to any internal operational decisions or policies.

• We are remote researchers with very limited observational capabilities.
Internet Archive


- (One of) The largest public online digital library

- > 1 Petabyte of content
  - 500 TB in Internet Crawls from 1996
  - 500 TB in non-encumbered Media (audio, video, books)

- Sustained > 3Gb/sec traffic -- 24/7.
Basic Architecture

- Front End HTTP Service
- Lookup Service
- Content Index
- Back End HTTP Service and Storage
- Monitoring Service

Internet Requests
Basic Architecture

Internet Requests

Front End HTTP Service

Content Index

Lookup Service

Back End HTTP Service and Storage

Monitoring Service
Basic Architecture

- Content Index
- Back End HTTP Service and Storage
- Front End HTTP Service
- Lookup Service
- Monitoring Service

Redirect
Basic Architecture

Content Index

Front End HTTP Service

Lookup Service

Response

Back End HTTP Service and Storage

Monitoring Service
Lookup Service

• Broadcast object ID to all storage servers

• Storage servers maintain catalog in RAM

• First to reply has the data
Lookup Service

- Scalable
- Simple Replication Algorithm
- Automatic load balancing
Performance

- 6 Gb/sec download to the Internet
- 100Mb/sec internal network
- > 2500 commodity nodes
- > 6000 hard disks
Throughput

- 2.3 - 48 terabytes/day
- 7.3 - 42.5 million hits/day
Adding a Cache

• Motivation: Reduce load on the back-end so that many nodes can be turned off on a regular basis.

• Don’t care about better performance
• Must be robust!
Empirical Data

- 7 days of logs: Nov 1 to Nov 7, 2008
- 270 million requests
- 240 terabytes of downloads
- 15 gigabytes of log files
Optimal Caching

- Use a Stack-Distance Algorithm to calculate optimal caching for LRU replacement strategies [Mattson, 1970]
Optimal Caching

- Use a Stack-Distance Algorithm to calculate optimal caching for LRU replacement strategies [Mattson, 1970]

30TB, 91.8% bytes, 95%
I/O ops per second

• Each file requires multiple IOPS
• Number of IOPS is related to size

• Assuming downloads happen instantaneously
• Calculate IOPS using Unix prefetch algorithm
I/O per second (01/11 - 07/11)
Sizing components

- Max IOPS: 4021411
- Average IOPS: 7734

- IOPS per hard disk: 250
- Number of hard disks for avg: 31
- Number of hard disks for max: 16086
Sizing components

• Max IOPS: 4021411
• Average IOPS: 7734

• IOPS per hard disk: 250
• Number of hard disks for avg: 31
• Number of hard disks for max: 16086

There are only 6000 hard disks in the system!
Sizing components

• The actual requirements is somewhere in the middle.
• We would prefer not to use hundreds of hard disks for the cache

• Solution: Solid State Disks
  – IOPS per solid state disk: 20000!
Conclusions

• The Internet Archive is a very robust, simple system
• Its simplicity is its greatest strength
  – No fancy algorithms
  – No high maintenance employees
  – No special hardware requirements
• The IA is one solid data point for live large scale systems architecture
Conclusions

• We have access logs to the IA from Jan 08 to the present: 700GB of logs
• These logs represent a treasure trove for empirical systems research