Online Applications and Services

Kai Shen
Dept. of Computer Science, University of Rochester

Classification of Online Applications and Services

- Goal:
  - understand the application/service properties that might affect the design of the server framework

- In terms of resource consumption from the server's perspective
  - I/O-intensive: Web search engines
  - CPU-intensive: scientific computing centers
  - Network-intensive: video-on-demand servers
  - A combination: multi-user high-resolution-simulation interactive gaming
  - Nothing: Web servers serving a moderate set of static Web pages

Classification of Online Applications and Services (cont.)

- In terms of the management of persistent state
  - read-only: most of the former mentioned
  - read/write: discussion groups "slashdot", online photo album, database-driven e-commerce apps
  - read/write with difference consistency semantics
    - loss tolerant
    - stale reading

- Any other important applications/classifications?
Functional Partitioning

- partition the service into smaller pieces based on the functionality
- dedicate a software server module (e.g., a request dispatcher/multiplexer and thread pool) for each piece
- optionally they can be put on separate hardware servers

MSN Groups - http://groups.msn.com

authenticate the user
request cache
discussion
groups management
online photo
album management
relational database service

Why Functional Partitioning?

- modularity
- service reuse
- targeted resource provisioning
- targeted system support

MSN Groups - http://groups.msn.com

authenticate the user
request cache
discussion
groups management
online photo
album management
relational database service

Web Search Engine

- HTTP service (Web servers)
- keyword translation service
- query cache service
- index searching service
- aggregate analysis/ranking service
- reverse document translation service

HTTP

- HTTP service
  - perform server-side HTTP functionalities
  - handle the response page layout

Classification

- resource consumption?
- management of persistent state?
Keyword Translation

- Keyword translation
  - translate the keywords into internal IDs
  - Why? save space, quick lookup
- Classification
  - resource consumption?
  - management of persistent state?

Query Cache

- Query cache service
  - cache old query results
  - return directly from cache if new queries hit the cache
- Classification
  - resource consumption?
  - management of persistent state?

Index Searching

- Index searching
  - find matching documents from the persistent index
  - perform intersection for multiple word search
- Classification
  - resource consumption?
  - management of persistent state?

Ranking

- Ranking
  - find the most relevant out of all matches, possibly use hyperlinks (yesterday's talk by Jon Kleinberg)
- Classification
  - resource consumption?
  - management of persistent state?
Reverse Document Translation

- Reverse document translation
  - translate the internal document IDs back to the original document URL

- Classification
  - resource consumption?
  - management of persistent state?

Putting It All Together

- HTTP service (Web servers)
  - little resource consumption; almost no persistent state
- keyword translation service
  - little resource consumption; read-only persistent state
- query cache service
  - moderately I/O-intensive; read/write persistent state, no much consistency requirement
- index searching service
  - highly I/O-intensive; read-only persistent state
- aggregate analysis/ranking service
  - CPU-intensive; read-only persistent state
- reverse document translation service
  - little resource consumption; read-only persistent state