Scalable Search Engine Solution

A Case Study of BBS

Yifu Huang

School of Computer Science, Fudan University
huangyifu@fudan.edu.cn

COMP620028 Information Retrieval Project, 2013

Outline

Motivation

- Current BBS donot support the full text search, but we need it indeed
- Get familiar with the implementation details behind search engine
- Build a scalable search engine solution that can be easily reused

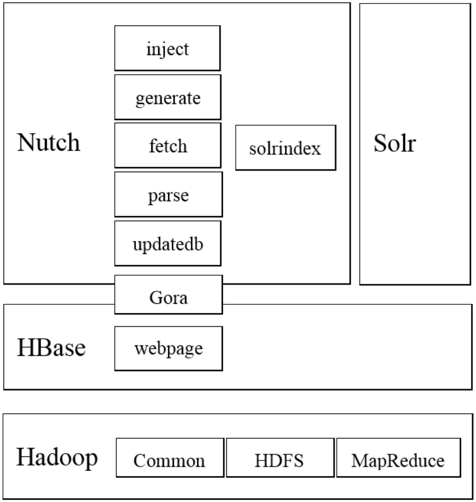
Motivation

- Current BBS donot support the full text search, but we need it indeed
- Get familiar with the implementation details behind search engine
- Build a scalable search engine solution that can be easily reused

Motivation

- Current BBS donot support the full text search, but we need it indeed
- Get familiar with the implementation details behind search engine
- Build a scalable search engine solution that can be easily reused

Architecture



Architecture (cont.)

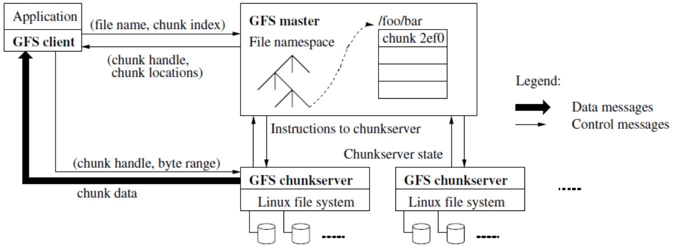
- Cluster
  - Intel(R) Core(TM)2 Duo CPU E7500 @ 2.93GHz
  - 4GB RAM
  - Hadoop 1.2.1
    - Namenode/Jobtracker: 1, datanode/tasktracker: 24
  - HBase 0.90.4
    - Master: 1, zookeeper: 3, regionserver: 24
- Data
  - Http://bbs.fudan.edu.cn/bbs/all
  - Board: 376
  - Post: 3111945

Architecture (cont.)

- Cluster
  - Intel(R) Core(TM)2 Duo CPU E7500 @ 2.93GHz
  - 4GB RAM
  - Hadoop 1.2.1
    - Namenode/Jobtracker: 1, datanode/tasktracker: 24
  - HBase 0.90.4
    - Master: 1, zookeeper: 3, regionserver: 24
- Data
  - Http://bbs.fudan.edu.cn/bbs/all
  - Board: 376
  - Post: 3111945

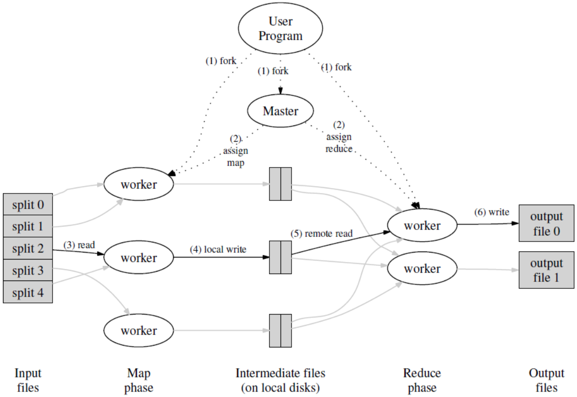
Hadoop

- Common
  - Configuration, serialization, compression, RPC, ...
- HDFS
  - Hdfs://namenode:9000/
  - Replication: 3



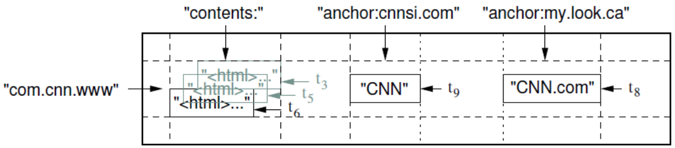
Hadoop (cont.)

- MapReduce
  - Map (k1, v1) -> (k2, v2)
  - Reduce (k2, v2) -> (k3, v3)



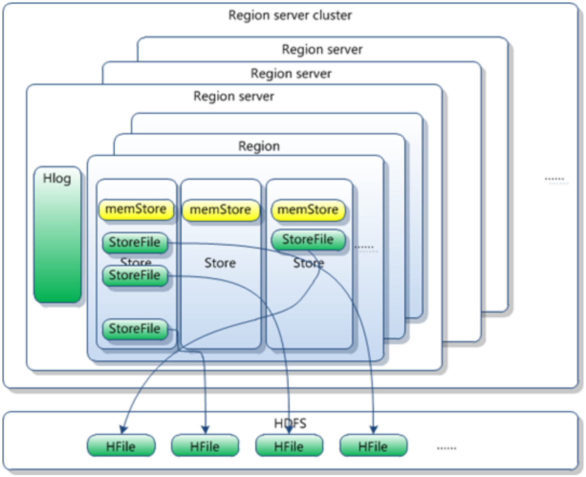
HBase

- Model
  - (row, column, time) -> cell



HBase (cont.)

- Storage



Nutch Inject

- Key idea
  - The number of URL is large, so we map them to different nodes and inject them into HBase webpage table
- Map (offset, line, reversedUrl, page)
  - Normalize
    - Regex, ...
  - Filter
    - Regex, prefix, suffix, ,domain, automaton, ...
  - ReversedUrl
    - E.g. "http://bar.foo.com:8983/to/index.html?a=b" becomes "com.foo.bar:8983:http/to/index.html?a=b"
  - Page
    - FetchTime, fetchInterval, metadata, score, marker, ...
- Reduce - default

Nutch Inject

- Key idea
  - The number of URL is large, so we map them to different nodes and inject them into HBase webpage table
- Map (offset, line, reversedUrl, page)
  - Normalize
    - Regex, ...
  - Filter
    - Regex, prefix, suffix, ,domain, automaton, ...
  - ReversedUrl
    - E.g. "http://bar.foo.com:8983/to/index.html?a=b" becomes "com.foo.bar:8983:http/to/index.html?a=b"
  - Page
    - FetchTime, fetchInterval, metadata, score, marker, ...
- Reduce - default

Nutch Inject

- Key idea
  - The number of URL is large, so we map them to different nodes and inject them into HBase webpage table
- Map (offset, line, reversedUrl, page)
  - Normalize
    - Regex, ...
  - Filter
    - Regex, prefix, suffix, ,domain, automaton, ...
  - ReversedUrl
    - E.g. "http://bar.foo.com:8983/to/index.html?a=b" becomes "com.foo.bar:8983:http/to/index.html?a=b"
  - Page
    - FetchTime, fetchInterval, metadata, score, marker, ...
- Reduce - default

Nutch Generate

- Key idea
  - We select some URL from HBase webpage table, map them to different nodes, and prepare to fech
- Map (reversedUrl, page, (url, scorce), page)
  - Check
    - Mark, distance, fetch schedule, score, ...
- Reduce ((url, scorce), page, reversedUrl, page)
  - Record
    - HostCount, domainCount, ...
  - Set
    - BatchId, marker, ...

Nutch Generate

- Key idea
  - We select some URL from HBase webpage table, map them to different nodes, and prepare to fech
- Map (reversedUrl, page, (url, scorce), page)
  - Check
    - Mark, distance, fetch schedule, score, ...
- Reduce ((url, scorce), page, reversedUrl, page)
  - Record
    - HostCount, domainCount, ...
  - Set
    - BatchId, marker, ...

Nutch Generate

- Key idea
  - We select some URL from HBase webpage table, map them to different nodes, and prepare to fech
- Map (reversedUrl, page, (url, scorce), page)
  - Check
    - Mark, distance, fetch schedule, score, ...
- Reduce ((url, scorce), page, reversedUrl, page)
  - Record
    - HostCount, domainCount, ...
  - Set
    - BatchId, marker, ...

Nutch Fetch

- Key idea
  - We map URL with random id to different nodes and fetch them with multi-threads in each node
- Map (reversedUrl, page, random_id, (conf, reversedUrl, page))
  - Check
    - BatchId, marker, resume, ...
- Reduce (random_id, (conf, reversedUrl, page), reversedUrl, page)
  - One producer to multiple consumers

Nutch Fetch

- Key idea
    - We map URL with random id to different nodes and fetch them with multi-threads in each node
- Map (reversedUrl, page, random_id, (conf, reversedUrl, page))
    - Check
        - BatchId, marker, resume, ...
- Reduce (random_id, (conf, reversedUrl, page), reversedUrl, page)
    - One producer to multiple consumers

Nutch Fetch

- Key idea
  - We map URL with random id to different nodes and fetch them with multi-threads in each node
- Map (reversedUrl, page, random_id, (conf, reversedUrl, page))
  - Check
    - BatchId, marker, resume, ...
- Reduce (random_id, (conf, reversedUrl, page), reversedUrl, page)
  - One producer to multiple consumers

Nutch Fetch (cont.)

- QueueFeeder
  - Feed the queues with input items, and re-fills them as items are consumed by FetcherThread-s
- FetcherThread
  - Pick items from queues and fetches the pages
    - Check robot rules
    - Check crawl delay schedule
    - Get page content
    - Check status code

Nutch Fetch (cont.)

- QueueFeeder
    - Feed the queues with input items, and re-fills them as items are consumed by FetcherThread-s
- FetcherThread
    - Pick items from queues and fetches the pages
        - Check robot rules
        - Check crawl delay schedule
        - Get page content
        - Check status code

Nutch Parse

- Key idea
  - We map URL to different nodes, extract field from them and save into HBase webpage table
- Map (reversedUrl, page, reversedUrl, page)
  - Set
    - Text, title, signature, outlinks, ...
- Reduce - default

Nutch Parse

- Key idea
    - We map URL to different nodes, extract field from them and save into HBase webpage table
- Map (reversedUrl, page, reversedUrl, page)
    - Set
        - Text, title, signature, outlinks, ...
- Reduce - default

Nutch Parse

- Key idea
    - We map URL to different nodes, extract field from them and save into HBase webpage table
- Map (reversedUrl, page, reversedUrl, page)
    - Set
        - Text, title, signature, outlinks, ...
- Reduce - default

Nutch Updatedb

- Key idea
  - We map URL to different nodes, extract their outlinks, and prepare to fetch these outlinks
- Map (reversedUrl, page, (reversedOut, score), pageOut)
  - Check outlink depth ...
- Reduce ((reversedOut, score), pageOut, reversedOut, pageOut)
  - Set inlinks ...

Nutch Updatedb

- Key idea
    - We map URL to different nodes, extract their outlinks, and prepare to fetch these outlinks
- Map (reversedUrl, page, (reversedOut, score), pageOut)
    - Check outlink depth ...
- Reduce ((reversedOut, score), pageOut, reversedOut, pageOut)
    - Set inlinks ...

Nutch Updatedb

- Key idea
  - We map URL to different nodes, extract their outlinks, and prepare to fetch these outlinks
- Map (reversedUrl, page, (reversedOut, score), pageOut)
  - Check outlink depth ...
- Reduce ((reversedOut, score), pageOut, reversedOut, pageOut)
  - Set inlinks ...

Nutch Solrindex

- Key idea
    - We map URL to different nodes, generate document and build index to solr server
- Map (reversedUrl, page, reversedUrl, doc)
    - Set
        - Id, digest, batchId, boost, ...
- Reduce - default

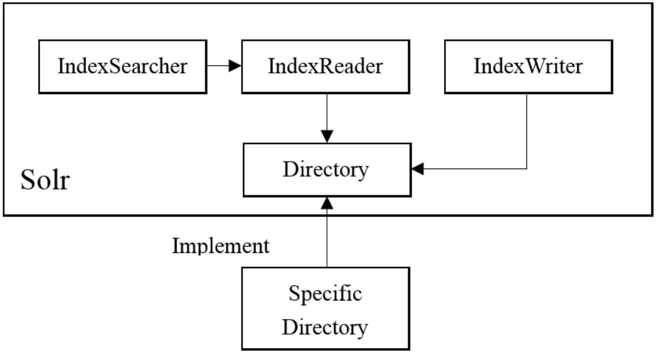
Nutch Solrindex

- Key idea
  - We map URL to different nodes, generate document and build index to solr server
- Map (reversedUrl, page, reversedUrl, doc)
  - Set
    - Id, digest, batchId, boost, ...
- Reduce - default

Nutch Solrindex

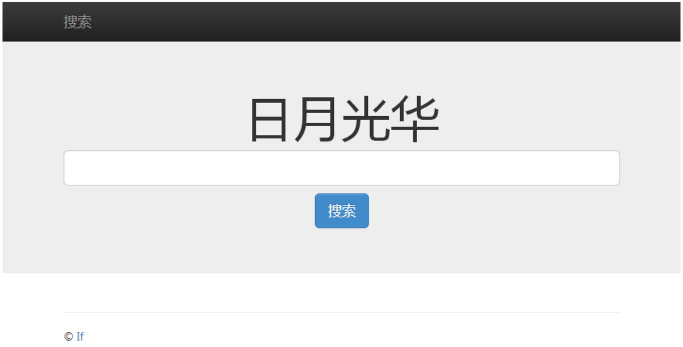
- Key idea
  - We map URL to different nodes, generate document and build index to solr server
- Map (reversedUrl, page, reversedUrl, doc)
  - Set
    - Id, digest, batchId, boost, ...
- Reduce - default

Solr


IndexSearcher
IndexReader
IndexWriter
Directory
Solr
Implement
Specific Directory

Demo

$$\boxed{10.171.5.222{:}3000}$$



Discussion

- Result
  - BBS search engine demo
  - A scalable search engine solution that can be easily reused
- Future Work
  - Incremental crawler
  - Field search
  - Personalized recommendation
  - ...

Discussion

- Result
  - BBS search engine demo
  - A scalable search engine solution that can be easily reused
- Future Work
  - Incremental crawler
  - Field search
  - Personalized recommendation
  - ...

References I

- [1] The Google File System. SOSP2003.
- [2] MapReduce: Simplified Data Processing on Large Clusters. OSDI2004.
- [3] Bigtable: A Distributed Storage System for Structured Data. OSDI2006.
- [4] Hadoop: The Definitive Guide. 2012.
- [5] Data-Intensive Text Processing with MapReduce. 2010.
- [6] The Hadoop Distributed File System. MSST2010.
- [7] Apache Hadoop YARN: Yet Another Resource Negotiator. SOCC2013.
- [8] HBase: The Definitive Guide. 2011.