Search and Real-Time Analytics on Big Data

Sewook Wee, Ryan Tabora, Jason Rutherglen
Accenture & Think Big Analytics

Strata New York
October, 2012
Big Data: **data** becomes your core asset. It realizes its value when you know **how to do what**

**Data**
- Volume, Velocity, Variety
- Database, Web, Blogs, Social media, Sensor data, smart meter data, log files

**Architecture**
- Scalable, Agile, Resilient, Affordable
- Hadoop, NoSQL, NewSQL, MPP

**Analytics**
- Know your customers better, Know your business better, Know your competitors better
- Recommender system, Operational intelligence, Competitive intelligence
What does that mean? Let’s go with an example: Vehicle Telematics

Big Data and Search

You often find it useful to have search capability when building service layer down the road.
Search basics

Search engine responds to queries NOT from raw documents but from pre-built index.

**Batch**

1. Crawl/import raw documents
2. Extract terms by tokenizing/filtering/stemming
3. Build index structure

**Search**

1. Parse the query
2. Look up index
3. Retrieve documents
4. Rank, cluster, and present

**Raw data**

<table>
<thead>
<tr>
<th>Doc ID</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>doc0</td>
<td>... the brief for big data will be given to those who...</td>
</tr>
<tr>
<td>doc1</td>
<td>...for writing massively parallel big data analytics, and ...</td>
</tr>
<tr>
<td>doc2</td>
<td>... Apache Solr for real-time Big Data analytics, ...</td>
</tr>
<tr>
<td>doc3</td>
<td>...conferences on various big data and other programming ...</td>
</tr>
</tbody>
</table>

**Index**

<table>
<thead>
<tr>
<th>Term</th>
<th>Termdocs</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Apache</td>
<td>doc2</td>
</tr>
<tr>
<td>big data</td>
<td>doc0, doc1, doc2, doc3</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

* text body from StrataNY 2012 tutorial abstracts
Search applications on unstructured data

Text Search
- Extract terms from documents
- Build an index that maps a term to a list of term documents
- Present the search results (PageRank, clustering)

Faceted Search
- Popular in online retailers
- Extract metadata (e.g., price, color, brand) from documents
- Use the metadata to interactively filter out search results
Other search applications

• Geo location search (near-by points of interest)
• Geo-aware (context-aware) text search
• Google image search
Big Data makes search interesting in two ways.

Data volume calls for distributed search
• Index size grows with data
• Sorted index is expensive to update after certain size
• Distributed search enables index sharding
• It comes with distributed system complexity

Data velocity calls for real time search
• Real time contents have a short life cycle
• They have less terms to index
• Their context is important but hard to capture
• It calls for alternatives to traditional batch index building and result ranking techniques
What are available distributed search technologies?

**Lucene-based**

- Apache Solr
- LucidWorks
- elasticsearch
- DataStax
- ZettaSearch
- Katta

**Custom engine**

- Amazon CloudSearch
- Riak
- Vertascale
- Lily
Apache Lucene and Solr

Solr: Lucene = Car : its engine – Kevin Tan (Supermind Consulting)

**Lucene**
- Open source information retrieval library
- Created in Java, ported to many other languages
- Maintains document-text indices and provides search capability

**Solr**
- Open source enterprise search platform
- Full-text search server, written in Java
- Provides distributed search and index replication

**SolrCloud**
- a set of new distributed capabilities in Solr
- Maintains shard and replication states using Zookeeper

Source: Grant Ingersoll (LucidWorks)

Commercial offerings based on Lucene/Solr

Focus on core search capability
- Extensive portfolio of data connectors
- Inherit Solr Cloud’s distributed system architecture
- Convenience layer (e.g., GUI for cluster configuration)

Making search capability setup simple
- Free search schema
- Use JSON over HTTP interface
- Use Lucene, but NOT Solr. It built its own distributed system
- Automatic shard allocation and migration
- Facets, highlighting, custom scripts features

One integrated system for both DB and search
- Use Cassandra in place of Solr Cloud
- Store index segment locally keeping LuceneIndexFormat ➔ leverage Lucene performance
- Store raw field data in Cassandra ➔ sync with raw data, durability + availability of Solr

* Different from Solandra
Other offerings based on custom search engines

**Fully-managed search service in the cloud**
- Leverages A9.com search technology
- Users 1) Create a search domain 2) Configure your search fields 3) Upload your data for indexing, and 4) Submit search requests from your web site or application
- Auto scaling & fault handling behind the scene

**A distributed, full-text search engine that is built on Riak Core**
- Index Riak KV objects as they're written using a precommit hook.
- Support various MIME types (e.g., JSON, XML, text)
- Facets, highlighting, custom scripts features
- Focus on real-time aspect
- Term-based partitioning (a.k.a, global indexing)

**Makes Hadoop, S3 & HBase searchable**
- Builds indexes using Hadoop MapReduce
- Stores indexes on HDFS, S3 or HBase
- Query with a thin client runtime
- Cloud (AWS) or On-Prem deployment options
- Beta release stage

---

Other offerings based on custom search engines