Building Scalable Big Data Infrastructure Using Open Source Software

Sam William
sampd@stumbleupon.com
What is StumbleUpon?

The best way to discover **new** and **interesting** things from across the Web.
How StumbleUpon works

1. Register

2. Tell us your interests

3. Start Stumbling and rating web pages

We use your interests and behavior to recommend new content for you!
StumbleUpon

By the Numbers

Registered Users: +25 million
Advertisers: +80,000
Employees: 75
Average Stumbles a Month: 300
Percentage of Mobile Stumbles: 40%
Time Spent Stumbling: 7 hours a month
Offices:
- San Francisco
- New York
Size of Index: +100,000,000 Web Pages
The Data Challenge

1. Data collection
2. Real time metrics
3. Batch processing / ETL
4. Data warehousing & ad-hoc analysis
5. Business intelligence & Reporting
Challenges in data collection

• Different services deployed of different tech stacks

<table>
<thead>
<tr>
<th>Site</th>
<th>Rec / Ad Server</th>
<th>Other internal services</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apache / PHP</td>
<td>Scala / Finagle</td>
<td>Java / Scala / PHP</td>
</tr>
</tbody>
</table>

• Add minimal latency to the production services
• Application DBs for Analytics / batch processing
  – From HBase & MySQL
Data Processing and Warehousing

Challenges/Requirements:

• Scale over 100 TBs of data
• End product works with easy querying tools/languages
• Reliable and Scalable powers analytics and internal reporting.
Real-time analytics and metrics

• Atomic counters
• Tracking product launches
• Monitoring the health of the site
• Latency – live metrics makes sense
• A/B tests
Data Collection at SU
Activity Streams and Logs

All messages are Protocol Buffers

✓ Fast and Efficient
✓ Multiple Language Bindings (Java/ C++ / PHP )
✓ Compact
✓ Very well documented
✓ Extensible
Apache Kafka

• Distributed pub-sub system
• Developed @ LinkedIn
• Offers message persistence
• Very high throughput
  – ~300K messages/sec
• Horizontally scalable
• Multiple subscribers for topics
  – Easy to rewind
Kafka

- Near real time process can be taken offline and done at the consumer level
- Semantic partitioning through topics
- Partitions for parallel consumption
- High-level consumer API using ZK
- Simple to deploy—only requires Zookeeper
Kafka At SU

- 4 Broker nodes with RAID10 disks
- 25 topics
- Peak of 3500 msg/s
- 350 bytes avg. message size
- 30 days of data retention
Sutro

- Scala/Finagle
- Generic Kafka message producer
- Deployed on all prod servers
- Local http daemon
- Publishes to Kafka asynchronously
- Snowflake to generate unique Ids
Sutro - Kafka

Site - Apache/PHP
Ad Server - Scala/Finagle
Rec Server - Scala/Finagle
Other Services

Kafka

Broker
### Application Data for Analytics & Batch Processing

<table>
<thead>
<tr>
<th>HBase</th>
<th>MySQL</th>
</tr>
</thead>
<tbody>
<tr>
<td>• <strong>HBase</strong> inter-cluster replication (from production to batch cluster)</td>
<td>• <strong>MySQL</strong> replication to <strong>Batch DB Servers</strong></td>
</tr>
<tr>
<td>• Near <strong>real-time sync</strong> on batch cluster</td>
<td>• <strong>Sqoop</strong> incremental data transfer to <strong>HDFS</strong></td>
</tr>
<tr>
<td>• Readily available in <strong>Hive</strong> for analysis</td>
<td>• <strong>HDFS</strong> flat files mapped to <strong>Hive</strong> tables &amp; made available for analysis</td>
</tr>
</tbody>
</table>

**StumbleUpon**
Real-time metrics

1. HBase – Atomic Counters

2. Asynchbase - Coalesced counter inc++

3. OpenTSDB (developed at SU)
   - A distributed time-series DB on HBase
   - Collects over 2 Billion data points a day
   - Plotting time series graphs
   - Tagged data points
Real-time counters

Real time metrics from OpenTSDB
Kafka Consumer framework aka Postie

- Distributed system for consuming messages
- Scala/Akka - on top of Kafka’s consumer API
- Generic consumer - understands protobuf
- Predefined sinks HBase / HDFS (Text/Binary) / Redis
- Consumers configured with configuration files
- Distributed / uses ZK to co-ordinate
- Extensible
Akka

• Building concurrent applications made easy !!
• The distributed nodes are behind Remote Actors
• Load balancing through custom Routers
• The predefined sink and services are accessed through local actors
• Fault-tolerance through actor monitoring
**Batch processing / ETL**

**GOAL**: Create simplified data-sets from complex data

- Create highly denormalized data sets for faster querying
- Output structured data for specific analysis
- Power the reporting DB with daily stats
  
  e.g. Registration Flow analysis
Our favourite ETL tools:

- **Pig**
  - Optional Schema
  - Work on byte arrays
  - Many simple operations can be done without UDFs
  - Developing UDFs is simple (understand Bags/Tuples)
  - Concise scripts compared to the M/R equivalents

- **Scalding**
  - Functional programming in Scala over Hadoop
  - Built on top of Cascading
  - Operating over tuples is like operating over collections in Scala
  - No UDFs .. Your entire program is in a full-fledged general purpose language
Warehouse - Hive

- Uses SQL-like querying language
- All Analysts and Data Scientists versed in SQL
- Supports Hadoop Streaming (Python/R)
- UDFs and Serdes make it highly extensible
- Supports partitioning with many table properties configurable at the partition level
Hive at StumbleUpon

**HBaseSerde**
- Reads binary data from HBase
- Parses composite binary values into multiple columns in Hive (mainly on key)

**ProtobufSerde**
- For creating Hive tables on top of binary protobuf files stored in HDFS
- Serde uses Java reflection to parse and project columns
Data Infrastructure at SU
Data Consumption
Who uses this data?

• Data Scientists/Analysts
• Offline Rec pipeline
• Ads Team

All this work allows them to focus on querying and analysis which is critical to the business.
Business Analytics / Data Scientists

- Feature-rich set of data to work on
- Enriched/Denormalized tables reduce JOINs, simplifies and speeds queries – shortening path to analysis.
- R: our favorite tool for analysis post Hadoop/Hive.
Recommendation platform

• URL score pipeline
  – M/R and Hive on Oozie
  – Filter / Classify into buckets
  – Score / Loop
  – Load ES/HBase index

• Keyword generation pipeline
  – Parse URL data
  – Generate Tag mappings
URL score pipeline

- URL Data
- Clean URLs
- Create Scores
- Filter
- Update Scores

ES Index
ES Index
ES Index
Advertisement Platform

• **Billing Service**
  – RT Kafka consumer
  – Calculates skips
  – Bills customers

• **Audience Estimation tool**
  – Pre-crunch data into multiple dimensions
  – A UI tool for Advertisers to estimate target audience

• **Sales team tools**
  – Built with PHP leveraging Hive or pre-crunch ETL data in HBase
More stuff on the pipeline

• Storm from Twitter
  – Scope for lot more real time analytics.
  – Very high throughput and extensible
  – Applications in JVM language

• BI tools
  – Our current BI tools / dashboards are minimal
  – Google charts powered by our reporting DB (HBase primarily).
Open Source FTW!!

- Actively developed and maintained
- Community support
- Built with web-scale in mind
- Distributed systems – Easy with Akka/ZK/Finagle
- Inexpensive
- Only one major catch !!
  - Hire and retain good engineers !!
Thank You!

Questions

?