Youtube Data Warehouse

Biswa Chatterjee
biswapesh@google.com
XLDB 2011
YTDW - Motivation & History

- Consolidated warehouse of Youtube data
- Videos, playbacks, summarized logs, etc.
- Very large (X PB uncompressed, Trillion row tables)
- High volume ETL (XXX TB processed / day)
- 100% Google Tech Stack:
  - Query: Oracle -> MySQL -> ColumnIO
  - ETL: Python -> Sawzall + Tenzing + Python
  - Reporting: Microstrategy -> ABI
- Key technologies: Sawzall, Tenzing, Dremel, ABI
YTDW - About Sawzall

- Scripting language on Google MR framework
- Sawzall vs MR-Saw
- Built-in security for accessing sensitive logs data
- Strong support for aggregation and complex computations
- Read/write various formats
- Procedural language
- Open sourced!

YTDW Usage:
- ETL of Youtube logs
- Complex one-off logs analysis
YTDW - About Tenzing

- **SQL on MR - Think HIVE, HadoopSQL**
- **Key strengths:**
  - Strong SQL support
  - Highly scalable - built on Google MR
  - Read / write many formats
- **Weaknesses:**
  - Not ideal for complex procedural code
  - Higher latency than Dremel
  - Limited support for nested-repeated structures
- **YTDW Usage:**
  - ETL for non-logs data, denormalizations
  - Medium complexity analysis on YTDW data
YTDW - About Dremel

- Current use in YTDW:
  - Reporting query engine
  - Interactive simple logs analysis
- Key Strengths
  - Very low latency
  - SQL support
  - Strong nested-relational support
  - Access to logs
- Limitations
  - More complex SQL constructs (joins, setops, ...)
  - Limited library of functions
  - Doesn't scale as much as MR
<table>
<thead>
<tr>
<th></th>
<th>Sawzall</th>
<th>Tenzing</th>
<th>Dremel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latency</td>
<td>High</td>
<td>Medium</td>
<td>Low</td>
</tr>
<tr>
<td>Scalability</td>
<td>High</td>
<td>High</td>
<td>Medium</td>
</tr>
<tr>
<td>SQL</td>
<td>None</td>
<td>High</td>
<td>Medium</td>
</tr>
<tr>
<td>Power</td>
<td>High</td>
<td>Medium</td>
<td>Low</td>
</tr>
</tbody>
</table>
YTDW Future: Query Engines

- Adding MR capabilities to Dremel
  - Scalable reliable shuffle
  - Materializing large result sets
  - Read / write multiple data formats
- Easier / more powerful analysis in Dremel
  - User defined scalar and table values functions
  - More SQL features:
    - Better support for joins
    - Analytic functions, set operators, etc.
- Long term for Dremel:
  - Completely replace Tenzing MR backend
  - Extend BigQuery service capabilities
YTDW - About ABI

● Complete reporting and dashboarding solution
● Built on Google stack
● Tight integration with Dremel and ColumnIO
● Google Visualizations, some Flash
● Current use in YTDW:
  ○ Most reports and dashboards
YTDW - Misc Technologies

- **Python**
  - Glue code - drivers, wrappers, etc.
  - Simple small scale extracts

- **Scheduler**
  - In-house scheduling framework
  - Built internally by YTDW engineers

- **C++ MapReduce**
  - Used sparingly for complex cases not possible using Sawzall / Tenzing

- **Query Rewriter**
  - Sits between ABI and Dremel
  - Rewrites queries to be faster / cheaper
YTDW - Performance

Performance improvement strategies:

● Data model:
  ○ De-normalized aggregated materialized views
  ○ Range partitioning

● Query rewrite layer:
  ○ Use the right aggregated materialized view
  ○ Prune partitions based on data knowledge

● Reporting front end
  ○ Aggressive result caching (memcache)
YTDW - Q & A