Data Warehousing and Analytics Infrastructure at Facebook

Ashish Thusoo & Dhruba Borthakur
athusoo,dhruba@facebook.com
Overview

- Challenges in a Fast Growing & Dynamic Environment
- Data Flow Architecture, Applications & Middleware
- Backend Infrastructure Challenges & Usage Statistics
- Conclusion
Accessibility

Simplify data access and use prevalent and familiar tools
Make data available as soon as possible
Flexibility

Support for different data formats

Support for user defined computations and extensions

Ability to evolve with data
Technologies

- Hadoop - Scalable Storage and Compute Grid
- Hive, coHive & HiPal - Extensible & Flexible Data Warehousing Framework that provides SQL-like Query Language on Hadoop
- Scribe-HDFS - Scalable Log Collection Infrastructure
- Nectar - Flexible Instrumentation Framework
- Databee & Chronos - Dependency-based Scheduling Infrastructure
**Usage**

- **Types of Applications:**
  - **Reporting**
    - Eg: Daily/Weekly aggregations of impression/click counts
    - Measures of user engagement
    - Microstrategy reports
  - **Ad hoc Analysis**
    - Eg: how many group admins broken down by state/country
  - **Machine Learning (Assembling training data)**
    - Ad Optimization
    - Eg: User Engagement as a function of user attributes
  - **Index Generation etc.**
Analysis and Data Organization

- > 99% of analysis through Hive on Hadoop

Hive:
- Easy to use: Familiar SQL interface with Data as Tables and Columns)
- Easy to extend:
  - Can embed map/reduce user programs in the data flow
  - Support for user defined functions
- Flexible:
  - Supports user defined data formats and storage formats
  - Support user defined types
- Interoperable:
  - JDBC, ODBC and thrift interfaces for integration with BI tools
Class MyEvent : NectarAppEvent

MyEvent->log()

1. Setup new scribe category
2. Copiers to copy data into warehouse
3. Automatic creation of tables in Hive _current and _r tables
4. Automatic detection of schema changes
5. Near real time event statistics dashboard

Nectar Framework Services
Data Discovery (10,000 feet)

1. Tag based search on Table Metadata
2. Crowd sourcing to generate tag information
3. Ability to ask questions from expert users that are identified by analyzing usage logs
4. Lineage information of data shown for browsing and data discovery

coHive Services in HiPal
Batch Jobs (10,000 feet)

- Chaining Transformations
  - Built-in Operators for Common Tasks
  - Specification for DAG of Transforms
  - Smart retries and checkpointing on failures
  - Chronos Scheduler
  - Data Availability Statistics and Alerts
  - Resource Utilization History
- Monitoring, Alerting & Statistics
- DataBee & Chronos Services
Hadoop & Hive Cluster @ Facebook

- **Hadoop/Hive cluster**
  - 16000 cores
  - Raw Storage capacity ~ 21PB
  - 8 cores + 12 TB per node
  - 32 GB RAM per node
  - Two level network topology
    - 1 Gbit/sec from node to rack switch
    - 10 Gbit/sec to top level rack switch

- **2 clusters**
  - One for adhoc users (SILVER cluster)
  - One for strict SLA jobs (PLATINUM cluster)
Hive & Hadoop Usage @ Facebook

- **Statistics per day:**
  - 800TB of I/O per day
  - 10K – 25K Hadoop jobs per day

- **Hive simplifies Hadoop:**
  - New engineers go though a Hive training session
  - Analysts (non-engineers) use Hadoop through Hive
  - Most of jobs are Hive Jobs
Data Collection and Real time Availability

- **Challenge: Large Volume of data to be queryable as soon as it is produced:**
  - Logs produce 100 TB per day uncompressed
  - Store in Scribe-HDFS clusters co-located with web-tier
    - Query-able within 15 seconds (via tail -f <filename>)
    - Overlapping HDFS clusters to avoid Single Point of Failure
    - Two separate NameNodes
    - Two DataNode instances on same physical machine
  - Gzip compressed, uploaded into central warehouse
    - Query-able within 15 minutes (via Hive external tables)
**Scribe-HDFS: Near Real Time Log Collection**

- **Scribe**
  - An Open Source Technology created at Facebook for Log Collection
  - Routes log messages from source (web machines) to destination storage
  - Persistently buffers data on intermediate nodes to deal with failures

- **Scribe-HDFS**
  - A highly scalable routing service + a highly scalable storage service
  - Enables us to scale with commodity nodes - no more expensive filers
Scribe-HDFS: 101

Scribed

Scribed

Scribed

HDFS Data Node

HDFS Data Node

HDFS Data Node

<category, msgs>

Append to
/staging/<category>/<file>

Scribe-HDFS
Challenge: Remove all single points of failure (SPOF)

- Hadoop NameNode was a SPOF
- Hadoop AvatarNode
  - Active-passive Hot Standby pair of NameNodes
  - Failover time for 20 PB file system having 65 million files is 10 seconds
  - Work-in progress to support active-active AvatarNode
Warehouse Utilization and Workload

- **Compute Map-Reduce cluster is CPU bound**
  - Peak usage of 95% CPU utilization
  - Peak network usage is 70% (network is not a bottleneck)
  - 70% tasks find data on local rack
  - We do not compress map outputs

- **Storage HDFS cluster is capacity bound**
  - 75% storage full (current size is 21 PB, 2000 nodes)
  - Disk bandwidth used is 20% of capacity (IO is not a bottleneck)
  - All Hive tables compressed using gzip
Hive: Optimizing Resource Utilization

- **Joins:**
  - Joins try to reduce the number of map/reduce jobs needed.
  - Memory efficient joins by streaming largest tables.
  - **Map Joins**
    - User specified small tables stored in hash tables on the mapper
    - No reducer needed

- **Aggregations:**
  - Map side partial aggregations
    - Hash-based aggregates
    - Serialized key/values in hash tables
  - 90% speed improvement on Query
    - `SELECT count(1) FROM t;`
  - Load balancing for data skew
Storage Costs

- **Challenge:** disk is cheap, but the cost of PetaBytes of disk is huge!

- **HDFS-RAID**
  - Default policy for HDFS is to replicate data 3 times
  - Generate parity block from a stripe of blocks from a file
  - HDFS-RAID replicates data 2 times and keeps parity blocks
  - Saves around 25% disk space

- **Columnar Compression**
  - Hive RCFile format organizes content by column
  - Saves around 15-25% of disk space
Single Warehouse for Adhoc and Periodic jobs

- **Challenge:** support adhoc and periodic jobs in a single instance of the warehouse
  - New job arrives every 10 seconds
  - 90% jobs are small and finish within a few minutes
  - Users run lots of experiments to mine data
  - Batch processing for large hourly/daily/weekly reporting
  - 99% uptime

- **Hadoop Map-Reduce FairShare Scheduler**
  - Each user gets an equal share of the cluster
  - Optimize latency for small jobs
  - Optimize cluster utilization and fair-share for larger jobs
  - Resource aware scheduling (CPU and memory resources only)
Isolation for Business Critical Pipelines

- **Challenge:** 100% uptime for a select set of pipeline
- **Separate a small storage & compute cluster**
  - PLATINUM cluster
  - Small subset of data from production warehouse
  - No adhoc queries
  - High bar for graduating a pipeline from production warehouse to PLATINUM warehouse
  - Poor man’s Disaster Recovery Solution (DR)
  - 99.9% uptime SLA
- **Hive replication**
  - Scribe copies data into PLATINUM
  - Copier looks at Hive query logs to replicate data from PLATINUM to SILVER cluster
Conclusion

- Scalable, Accessible & Flexible infrastructure a MUST

- Use & Contribute to Open Source to keep the costs low
  - Hadoop
  - Hive
  - Scribe-HDFS

- Always a challenge to keep up with a dynamically growing & changing environment