Apache Lens
Cut data analytics silos in your enterprise

Sharad Agarwal & Amareshwari Sriramadasu
Agenda

- Evolution of data analytics in enterprise
- Introduction to Apache Lens
- Architecture
- OLAP Data model
- Demo
- Roadmap
Reporting ware house : Generation 1: RDBMS

- Reporting data in RDBMS
- Aggregations/materialized views in DB
  - \( \sim 1 \) TB
Generation 1: Challenges

- Data Scale: Loading of data taking ~ 24 hrs
  - Analysis only upto 3 dimensions
  - Heavy queries stalling other user queries
- Unable to move fast with new reporting requirements
Reporting warehouse: Generation 2: Columnar DB

- Small and summarized data in Columnar Database
  - Rich Dashboards
Generation 2: Challenges

- Scalability challenges with data growth
- Expensive to grow the capacity on columnar DB
- Data modelling and ETL cycles are long
  - Limited Analytical flexibility
Generation 3: Columnar DB + Hadoop

- Small and summarized data in Columnar Database (~10 TB)
  - Rich Dashboards
- Granular data in Hadoop (100s of TB)
  - Adhoc analysis
Generation 3: Challenges

- Maintaining two lines of independent data warehousing systems
  - Data discrepancies
  - Schema management
- Learning curve for Users
  - Duplicate datasets
- Inefficient Utilization
Apache Lens (formerly Grill)

- Platform to enable multi-dimensional queries in a unified way over datasets stored in multiple warehouses
  - OLAP Cube abstraction
  - Data discovery by providing single metadata layer
  - Unified access to data by integrating Hive with other traditional warehouses
Apache Lens

- Queries get pushed to where data resides
- Central Catalog management: All applications talk same language
- Query analytics for optimizing hot datasets
- Workload based experimentation with newer systems: AWS Redshift, Apache Spark, Apache Tez
Analytics Use cases

- Reporting queries
- Adhoc queries
  - Interactive/Batch queries
- Scheduled queries
- Infer insights through ML algorithms
Why both Hadoop and traditional warehouse?

Canned queries are mostly Interactive

Adhoc queries can be Interactive or batch depending on the data volumes and query complexity
Lens Architecture

Applications – Reporting, Ad Hoc Queries

CLI, Java Client, JDBC Client

REST API for querying and schema update

Lens Server

Driver API

OLAP Cube Metastore

Based on underlying Hive Metastore/HCatalog

Hive (MR), JDBC, Spark SQL/Shark

Pluggable Execution Engines

HDFS, HBase, Columnar DWH, S3, Redshift

Data Stores
OLAP Data model

- Storage
- Cube
  - Fact Table
    - Physical Fact tables
  - Derived Cube
- Dimension
  - Dimension Table
    - Physical Dimension tables
Data model - Relationships

Cube

Dimension

Fact table

Cube

Dimension Table

Dimension

Fact table

Storage

Dimension table

Storage
Demo: Example data model

- Sales
  - Product
  - Customer
    - City
    - City
Demo: Example physical data model

- **Sales**
  - Raw Fact: HDFS
  - Aggregate Fact1: HDFS and DB
- **Customer**
  - Customer table: HDFS and DB
- **Product**
  - Product table: HDFS and DB
- **City**
  - City table: HDFS
  - City subset: DB
  - Aggregate Fact2: HDFS and DB
Demo
Roadmap

Immediate
- Add support for querying streaming data stores
- Query submission throttling at drivers
- Ability to load multiple instances of same driver

Medium Term
- Estimate query execution times
- Authorization across all services and storages
- Add scheduler service
- Make it suitable to integrate with BI tools
- Enable machine learning through Lens

Long term
- Query caching
- Metastore UI
- Administrator console
- Automatic roll up suggestions on hot datasets
Explorations

• Enable HiveDriver on Tez/Spark
• Explore Zeppelin for web front end
• Newer drivers : Elastic search driver, Druid driver
## Stay Involved

### Web site


### Source repo

- [https://git-wip-us.apache.org/repos/asf/incubator-lens.git](https://git-wip-us.apache.org/repos/asf/incubator-lens.git)

### Source repo for Hive

- [https://github.com/InMobi/hive](https://github.com/InMobi/hive)

### Mailing lists

- dev@lens.incubator.apache.org
- user@lens.incubator.apache.org
Thank You!

• Questions?