Too Big to Eat: Boosting Analytics Data Ingestion from Object Stores with Scoop

Yosef Moatti1, Eran Rom2, Raúl Gracia-tinedo3, Dalit Naor1, Doron Chen1, Josep Sampé3, Marc Sánchez-Artigas3, Pedro García-López3, Filip Gluszak4, Eric Deschodt4, Francesco Pace5, Daniele Venzano5, Pietro Michiardi5

IBM Research1, OpenStack Storlets Project2, Universitat Rovira i Virgili3, GridPocket4, Eurecom5

Problem

• Large Volumes of Unstructured Data Objects
• Disaggregated Compute and Storage Clusters
• Costly Data Ingestion When Running SQL Queries

Goals

• Analytics Jobs Delegate SQL Tasks to The Storage
• The Storage Cluster Runs These Tasks To Cut Ingestion
• Framework To Orchestrate Cooperation between Analytics Jobs and Storage

Results

• Open-Source Prototype Implementation
• GridPocket Use-Case: Energy Data Management
• Evaluation on a 63-Machine Cluster with Zoe – Analytics on demand

Platform & Experiments

Conceptual Design of Scoop

- Analytics frameworks
- Swift Proxys
- Swift Storage Nodes
- Compute cluster
- Storage cluster
- Pushdown task
- Analytics delegator
- Filter framework
- Object requests
- Object streams

Implementation/Deployment

Spark SQL

New RDD & Data Source to Delegate SQL Predicates

25 Spark Nodes

10 Gbps

New Storlet to handle Selections & Projections

35 Swift Nodes

SQL [Swift + Storlets]

Query Speedup vs Data Selectivity

High Query Speedup: As Query Data Selectivity Grows, Job Speedup Exhibits a Non-linear Improvement

Results on GridPocket Queries

Solving GridPocket Problems: Typical GridPocket Data Intensive Queries

Show Speedups Up to x32

Inspecting Resource Usage

Bandwidth throughput of 10/100 Gbps streams (ITB dataset)

99% data selectivity

CPU usage of Swift storage nodes (ITB dataset)

Average storage node CPU usage

Average storage node CPU usage with Scoop

Nice trade-off: We Avoid Consuming 99% Transfer BW by Using 20% of (Mostly Idle) Storage Compute Power