Bringing SQL to the OpenStack World

Apache Tajo on Swift

Jihoon Son
Apache Tajo PMC member

Who am I

- Jihoon Son
  - Ph.D candidate (Computer Science & Engineering, 2010.3 ~)
  - Tajo project co-founder (2010)
  - Apache Tajo PMC and Committer (2014.5.1 ~)
- Contacts
  - Email: jihoonson AT apache.org
  - LinkedIn: https://www.linkedin.com/in/jihoonson

Outline

- OpenStack Swift
- Apache Tajo
- Tajo on Swift
- Location-aware Computing of Tajo
- Brief Evaluation Results
- Roadmap

OpenStack Swift

- Popular object storage
  - Images, videos, logs, ...
- Enterprises store objects on Swift to provide their services
  - Usually private clusters

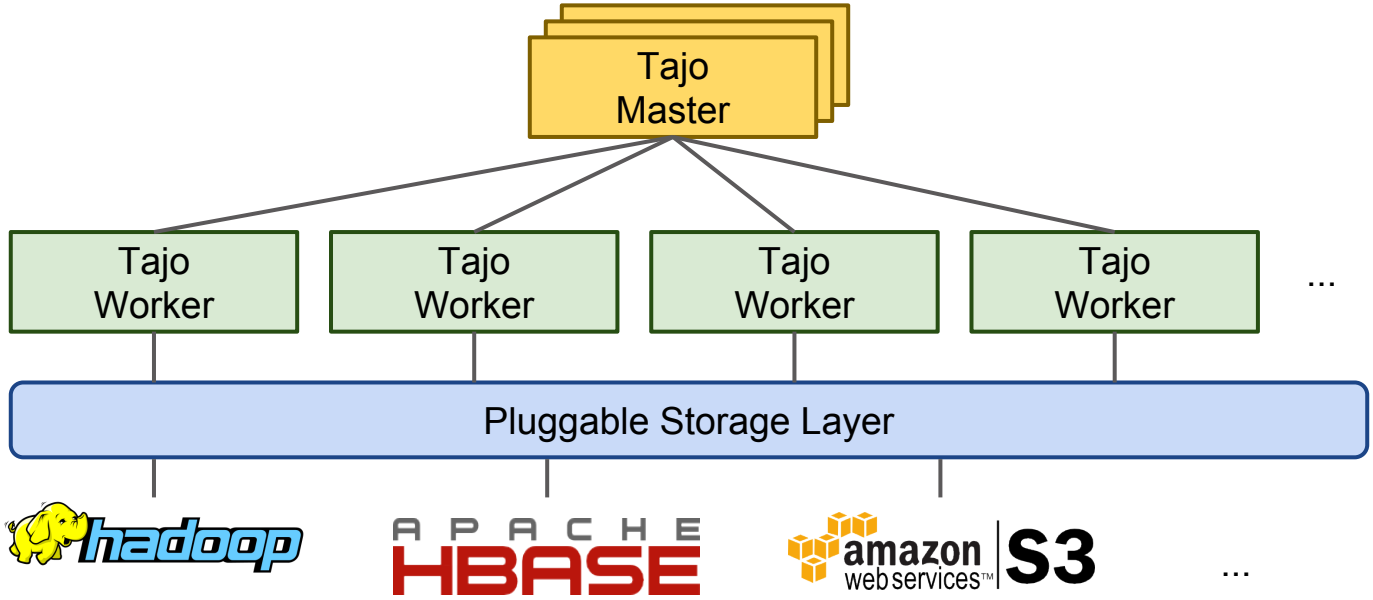
SQL on Swift

- Data analysis is important to improve the quality of their services
  - SQL is one of the most powerful and popular query languages
- Many enterprise data analysis tools relying on SQL
  - OLAP, visualization, data mining, …
- Need for using SQL on Swift

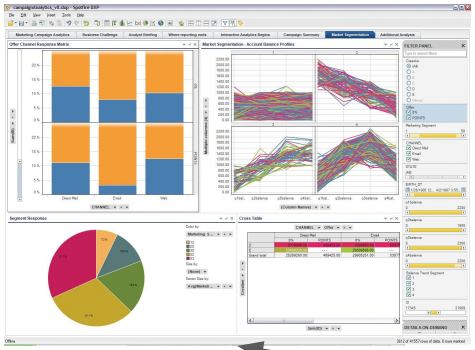
Apache Tajo

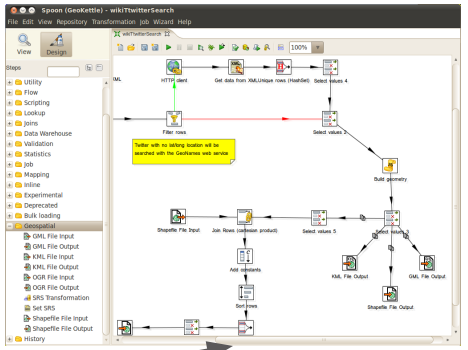
- Scalable, efficient, and fault-tolerant data warehouse system
  - Support SQL standards compliance
  - Efficient batch execution and interactive ad-hoc analysis
    - Low latency and high throughput
    - No use of MapReduce
  - No single point of failure

Apache Tajo


Tajo
Master
Tajo
Worker
Tajo
Worker
Tajo
Worker
Tajo
Worker
...
Pluggable Storage Layer
hadoop
APACHE
HBASE
amazon
web services
S3
...

Apache Tajo





Tajo cluster

hadoop

APACHE HBASE

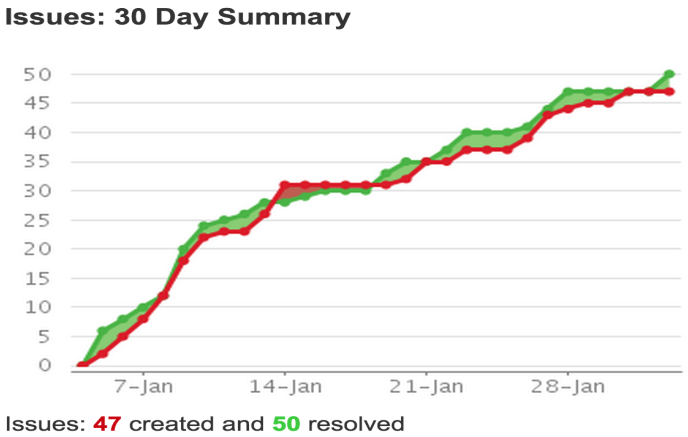
amazon web services™ S3

...

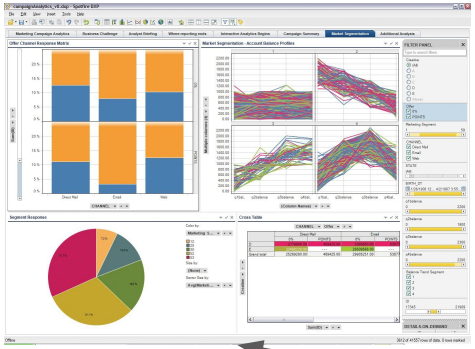
Apache Tajo

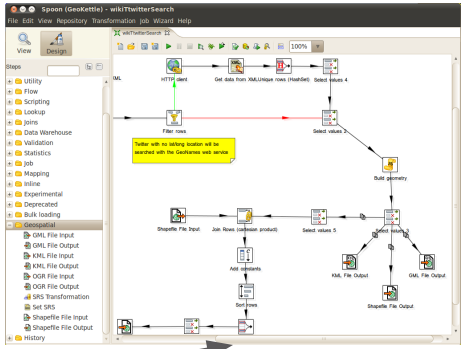
- Active open source project
  - Apache incubating project (2013)
  - Apache Top-Level Project (2014)
  - 18 committers + 16 contributors

- Activity summary



Tajo on Swift





Tajo cluster

hadoop

APACHE HBASE

openstack Swift

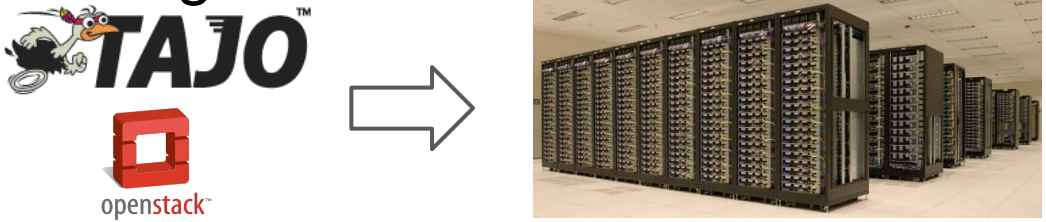
amazon web services S3 ...

Tajo on Swift

- No need to modify code of Tajo and Swift
  - Tajo can access Swift with the Hadoop-openstack library
    - But, doesn't need to install or run Hadoop
  - Just use it

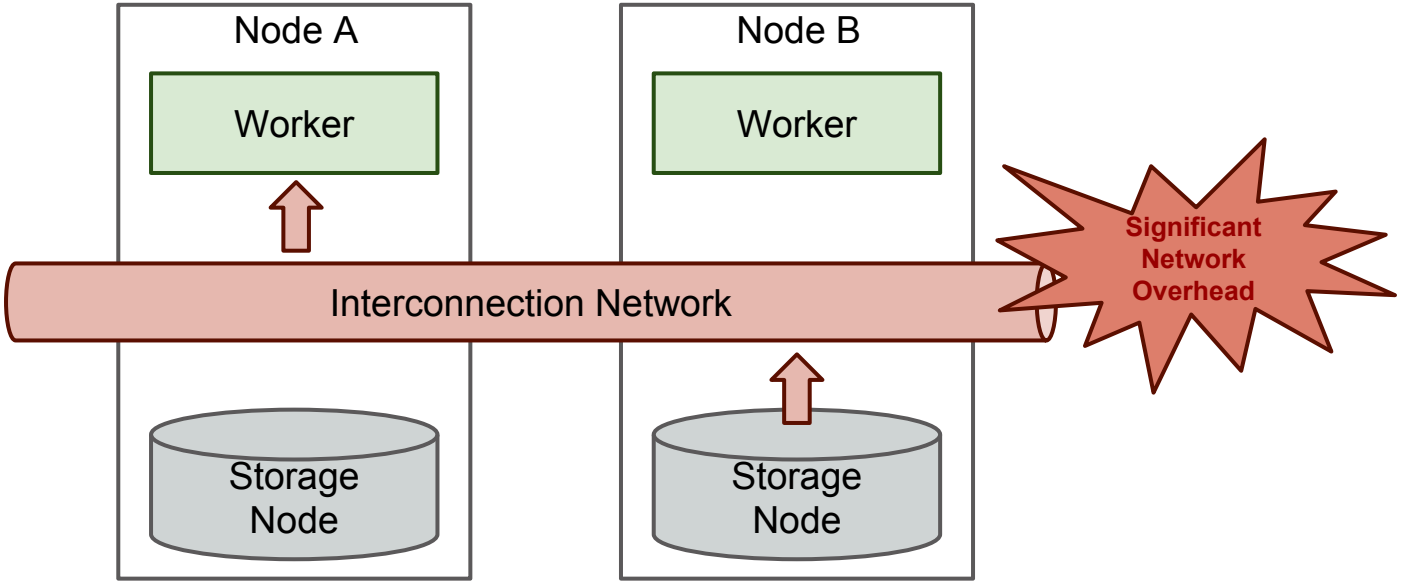
Tajo on Swift

- Focusing on private clusters
  - Tajo and Swift may share the same cluster
  - Able to optimize the cluster configuration and data deployment
- More efficient data analysis!
  - By reducing the overhead of bottlenecks



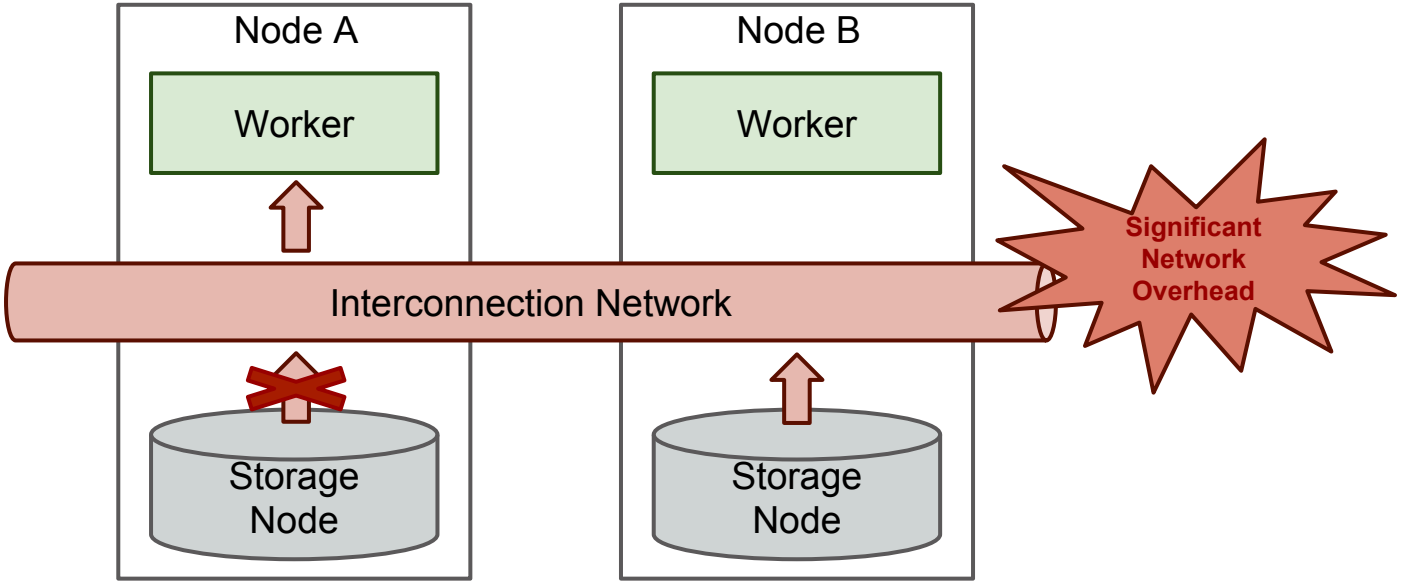
Data Locality Problem

- Network can be usually the bottleneck



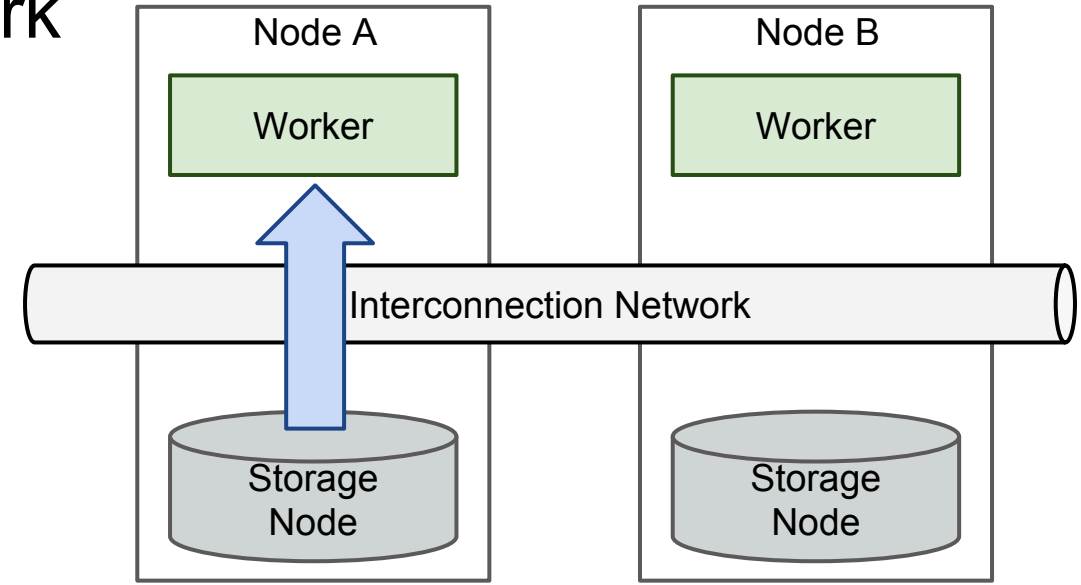
Data Locality Problem

- Network can be usually the bottleneck



Data Locality Problem

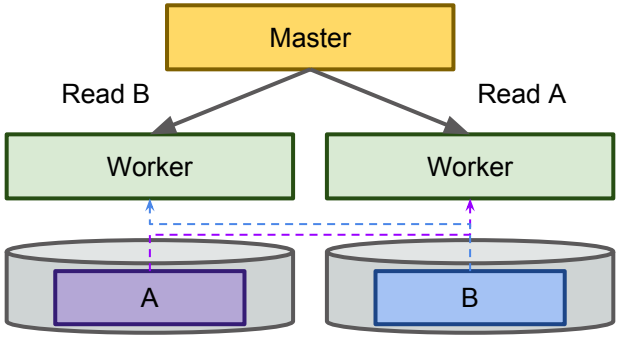
- Important to reduce data transmission over the network



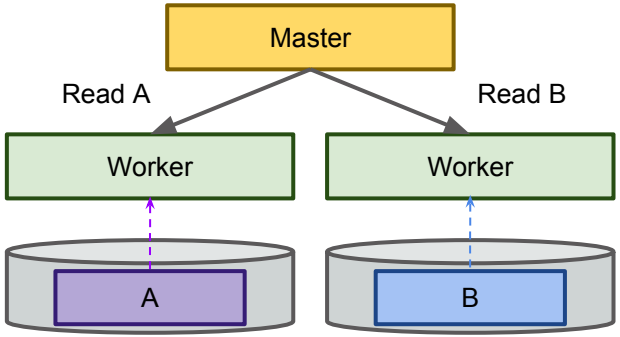
Location-aware Computing

- Moving the processing close to the data
  - Avoiding the performance degradation due to the data transfer over the network

< Moving data to tasks >



< Moving tasks to data >

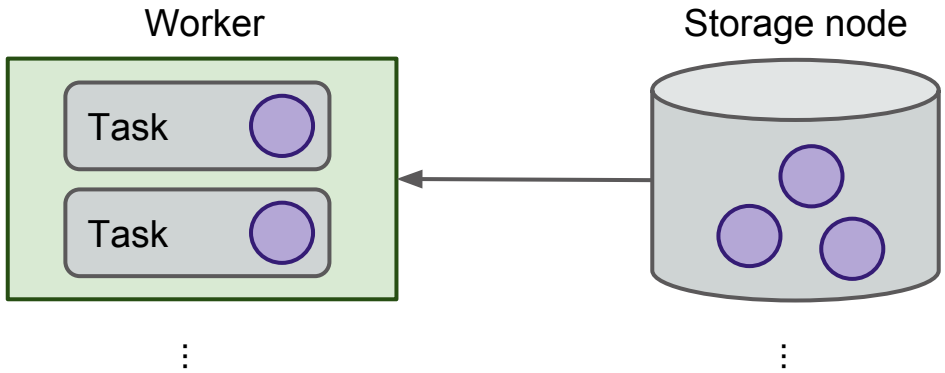


Location-aware Computing

- Three cases of user data deployment
  - Case 1: small objects without segmentation
  - Case 2: large objects with segmentation
  - Case 3: large objects without segmentation

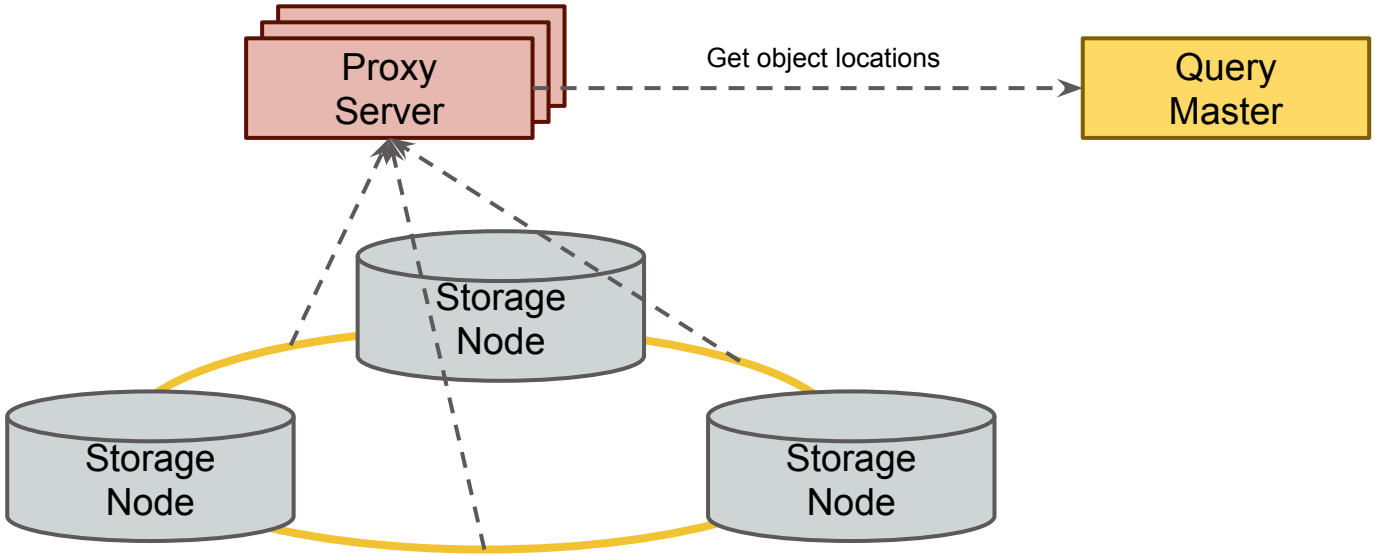
Case 1: Small Objects W/O Segmentation

- Each task processes an object
  - The object size should be sufficiently small to be processed by a task



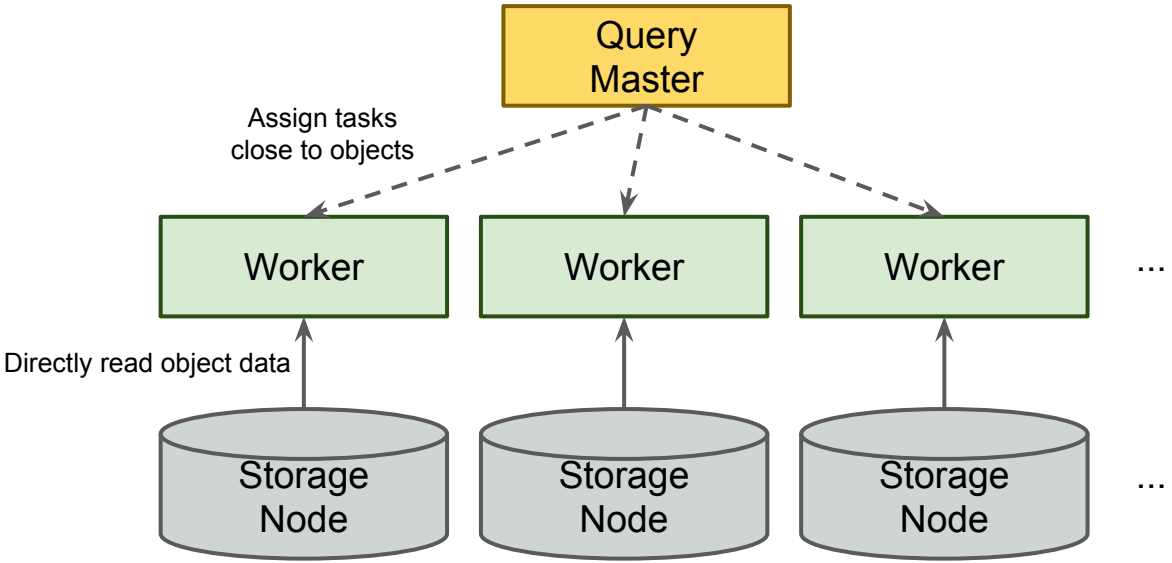
Case 1: Small Objects W/O Segmentation

1) Getting object locations from the ring



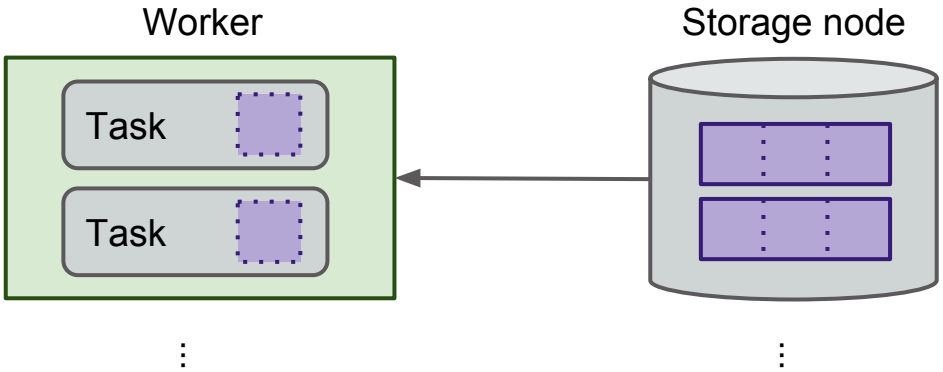
Case 1: Small Objects W/O Segmentation

2) Assigning tasks based on object locations



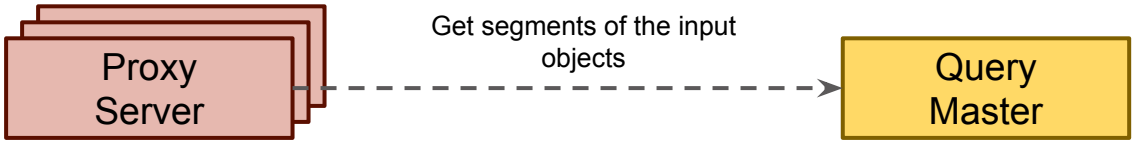
Case 2: Large Objects with Segmentation

- Each task processes a segment
  - The segment size should be sufficiently small to be processed by a task



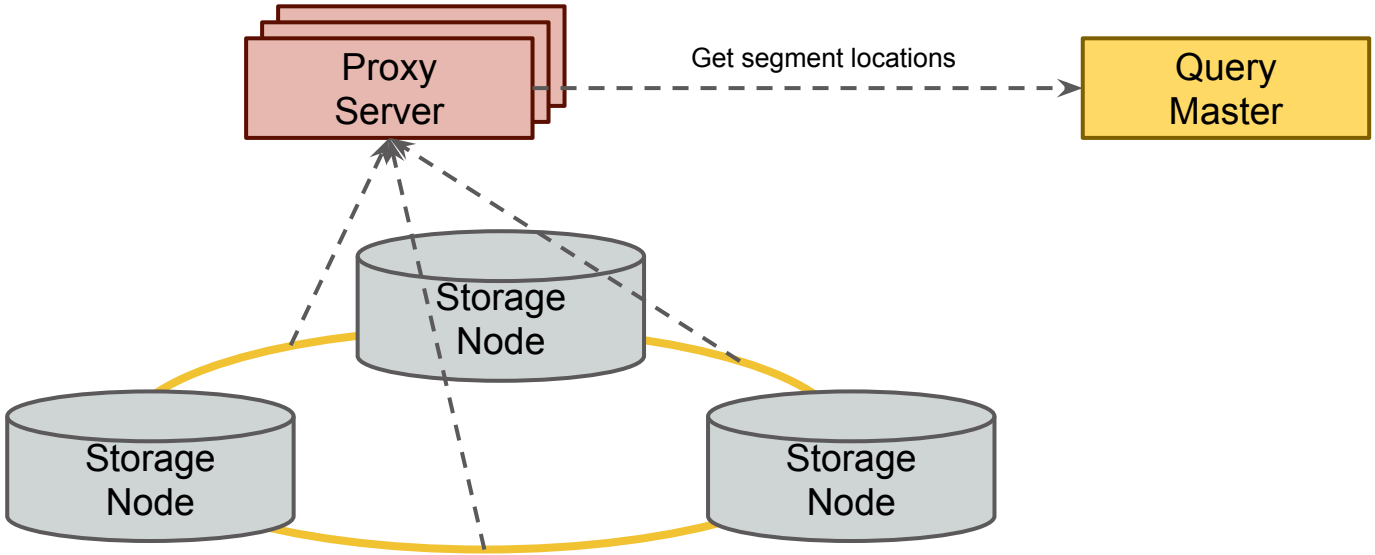
Case 2: Large Objects with Segmentation

1) Getting segments of the given objects



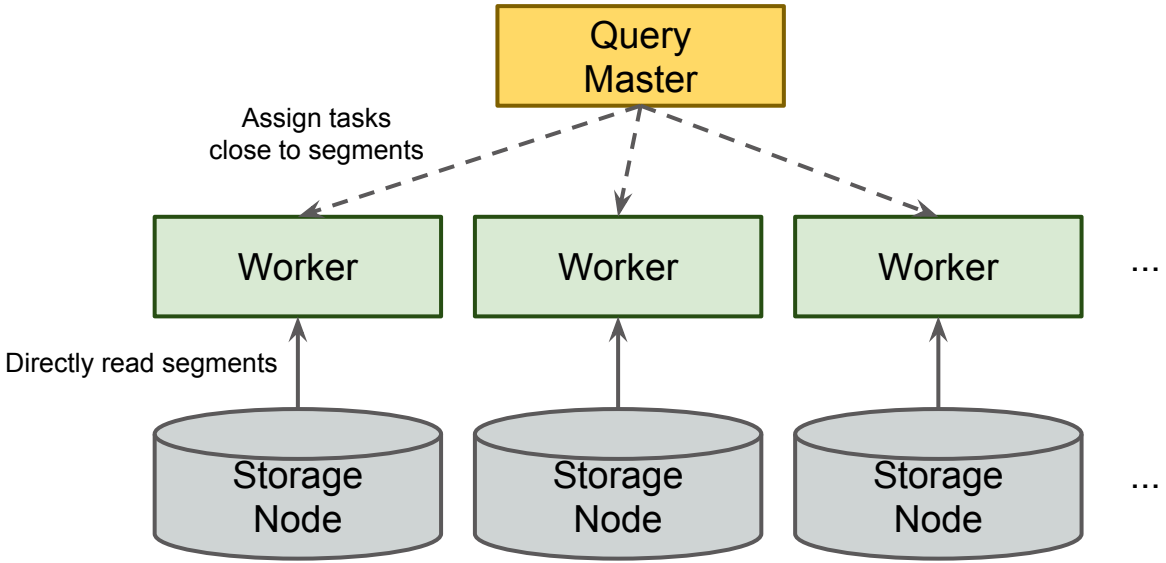
Case 2: Large Objects with Segmentation

2) Getting segment locations from the ring



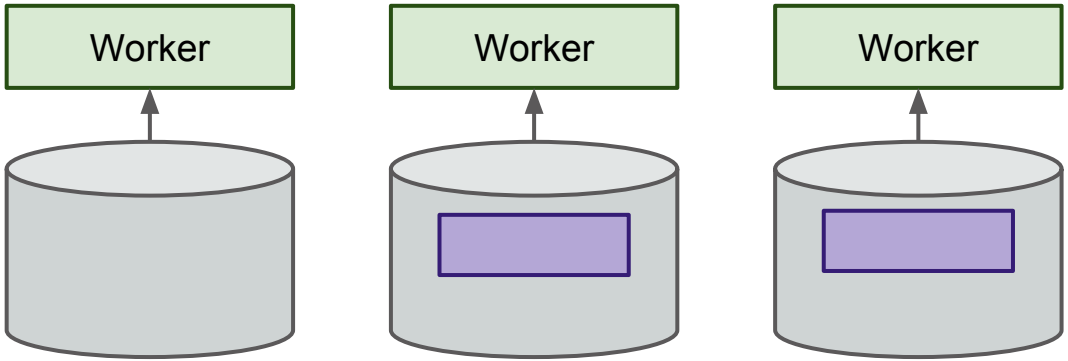
Case 2: Large Objects with Segmentation

3) Assigning tasks based on segment locations



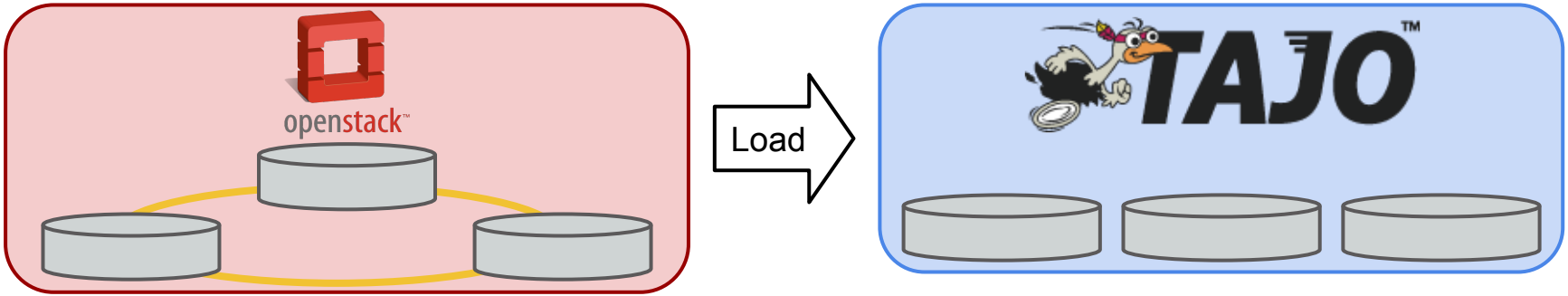
Case 3: Large Objects W/O Segmentation

- Inevitable performance degradation due to the suboptimal data deployment
  - Each Tajo worker processes large objects, which causes coarse-grained load balancing



Case 3: Large Objects W/O Segmentation

- Alternatively, Tajo will provide the *data load* feature
  - Load objects into the Tajo's optimized internal storage



Location-aware Computing

- Current status
  - Support location-aware computing without segmentation
- Future support
  - Location-aware computing for segmented objects
  - Data load into the Tajo's storage

Brief Evaluations

- Performance comparison with on another distributed storage
  - Swift VS Hadoop Distributed File System (HDFS)
- Scalability test of Swift

Brief Evaluations

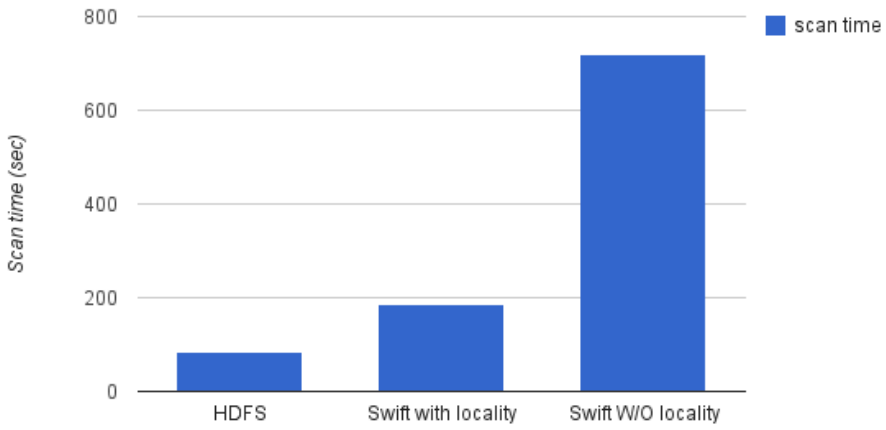
- Cluster configuration
  - 1 master + 8 slaves
    - Each worker is equipped with 1 disk
  - Swift: 1 proxy server + 8 storage nodes
  - HDFS: 1 namenode + 8 datanodes
  - Tajo: 1 master + 8 workers
    - Each worker can process 2 tasks simultaneously

Brief Evaluations

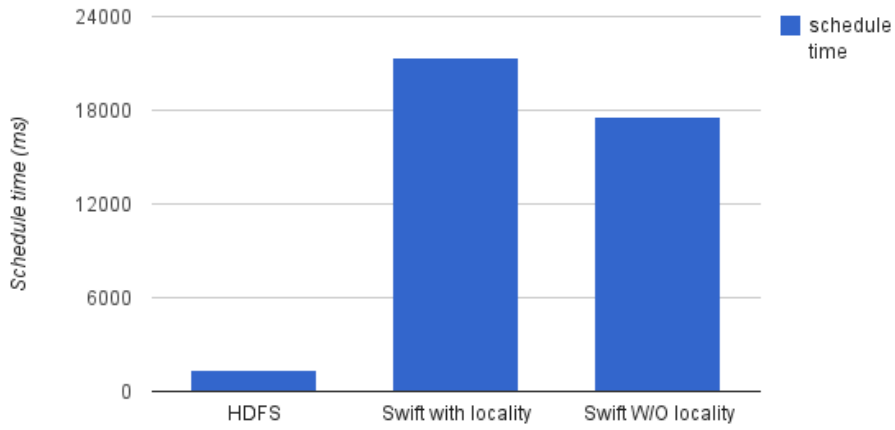
- Data set
  - Crawled twitter log
    - # of objects: 1014
    - Avg size of objects: 70MB
    - Total size: 69.5GB
- Query
  - Full scan query
    - `select * from twitter_log`

Comparison with HDFS

- Scan time

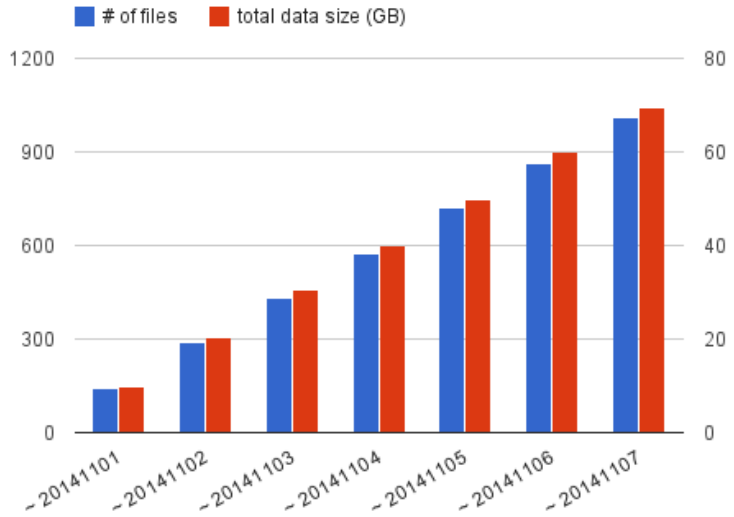


- Schedule time

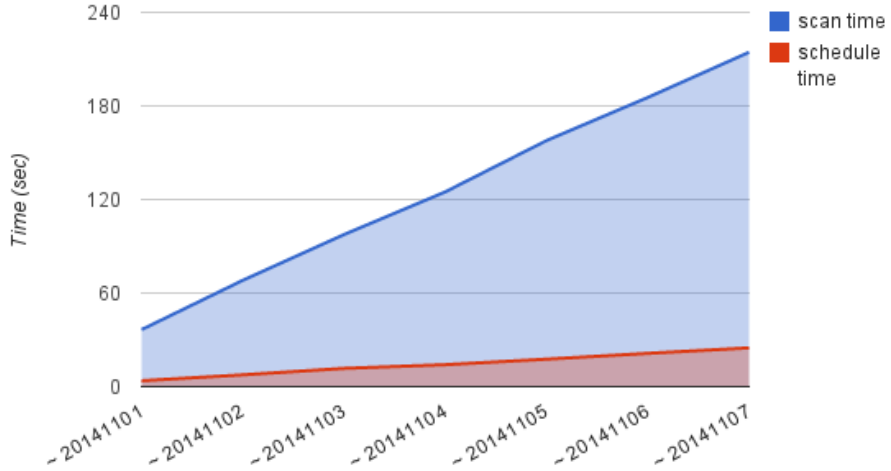


Scalability Test

- Data



- Result



Our Roadmap

- Storage layer specialized for Swift
  - Location-aware computing for segmented objects
  - Data load into the Tajo's storage
- Block storage support
  - Cinder and Ceph
- Provisioning Tajo clusters
  - Sahara
  - Heat, TOSCA

Thanks!

http://tajo.apache.org/