Serving Large-scale Batch Computed Data with Project Voldemort

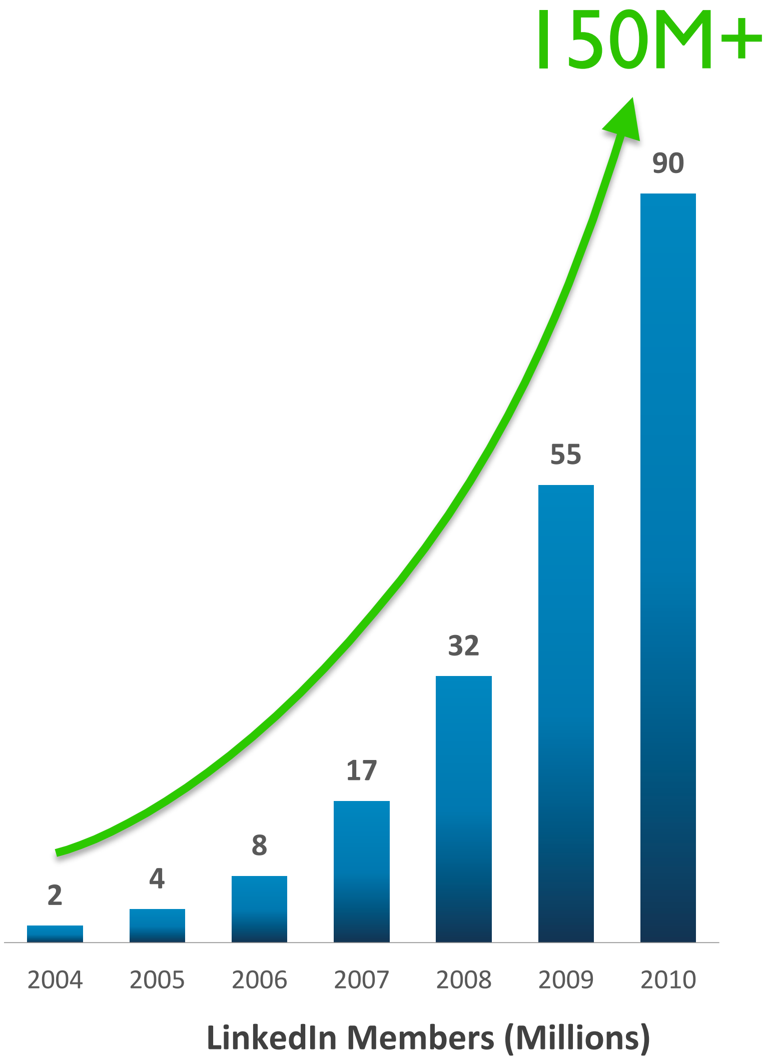
Roshan Sumbaly, Jay Kreps, Lei Gao, Alex Feinberg, Chinmay Soman, and Sam Shah

LinkedIn

USENIX Conference on File and Storage Technologies

February, 2012

LinkedIn



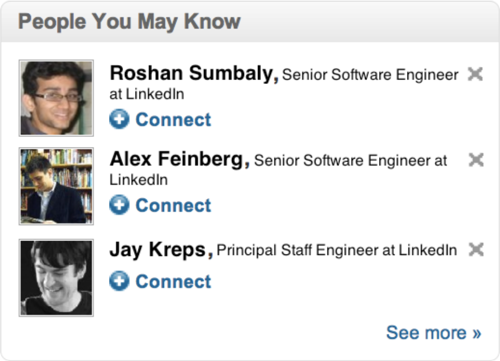
75%
Fortune 100 Companies use LinkedIn to hire

>2M
Company Pages

~4B
Searches in 2011

Data Features on LinkedIn

People You May Know



People You May Know

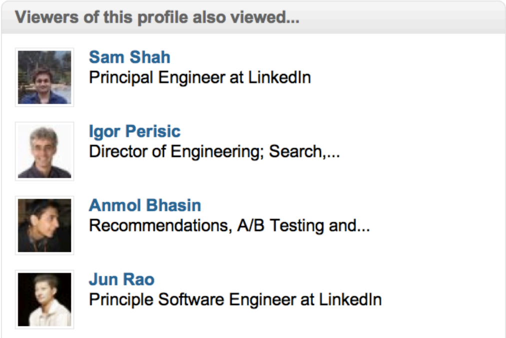
Roshan Sumbaly, Senior Software Engineer at LinkedIn ✕
Connect

Alex Feinberg, Senior Software Engineer at LinkedIn ✕
Connect

Jay Kreps, Principal Staff Engineer at LinkedIn ✕
Connect

See more »

Viewers of this profile also viewed



Viewers of this profile also viewed...

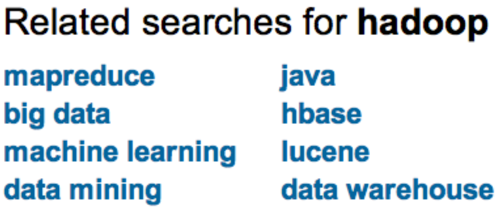
Sam Shah
Principal Engineer at LinkedIn

Igor Perisic
Director of Engineering; Search,...

Anmol Bhasin
Recommendations, A/B Testing and...

Jun Rao
Principle Software Engineer at LinkedIn

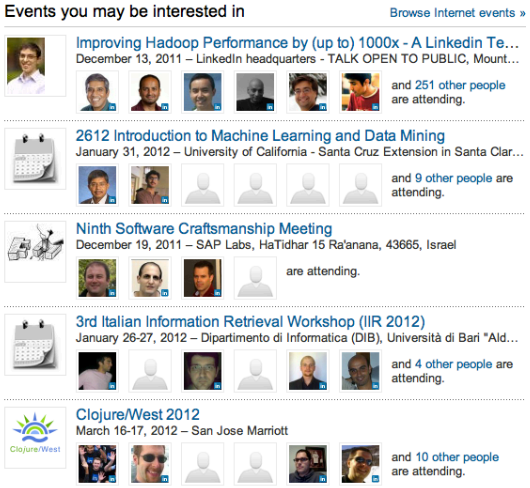
Related Searches



Related searches for **hadoop**

mapreduce	java
big data	hbase
machine learning	lucene
data mining	data warehouse

Events you may be interested in



Events you may be interested in — Browse Internet events »

Improving Hadoop Performance by (up to) 1000x - A Linkedin Te...
December 13, 2011 – LinkedIn headquarters - TALK OPEN TO PUBLIC, Mount...
and 251 other people are attending.

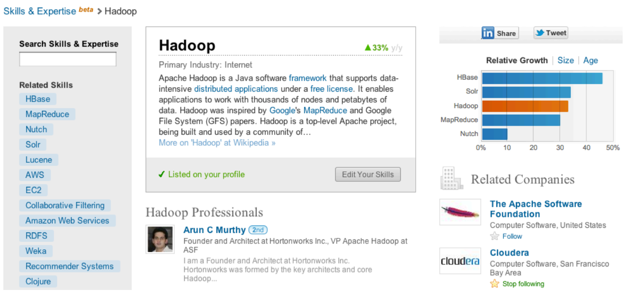
2612 Introduction to Machine Learning and Data Mining
January 31, 2012 – University of California - Santa Cruz Extension in Santa Clar...
and 9 other people are attending.

Ninth Software Craftsmanship Meeting
December 19, 2011 – SAP Labs, HaTidhar 15 Ra'anana, 43665, Israel
are attending.

3rd Italian Information Retrieval Workshop (IIR 2012)
January 26-27, 2012 – Dipartimento di Informatica (DIB), Università di Bari "Ald...
and 4 other people are attending.

Clojure/West 2012
March 16-17, 2012 – San Jose Marriott
and 10 other people are attending.

LinkedIn Skills



Skills & Expertise *beta* > Hadoop

Search Skills & Expertise

Related Skills
HBase
MapReduce
Nutch
Solr
Lucene
AWS
EC2
Collaborative Filtering
Amazon Web Services
RDFS
Weka
Recommender Systems
Clojure

Hadoop ▲33% y/y
Primary Industry: Internet
Apache Hadoop is a Java software framework that supports data-intensive distributed applications under a free license. It enables applications to work with thousands of nodes and petabytes of data. Hadoop was inspired by Google's MapReduce and Google File System (GFS) papers. Hadoop is a top-level Apache project, being built and used by a community of...
More on 'Hadoop' at Wikipedia »

✓ Listed on your profile — Edit Your Skills

Hadoop Professionals

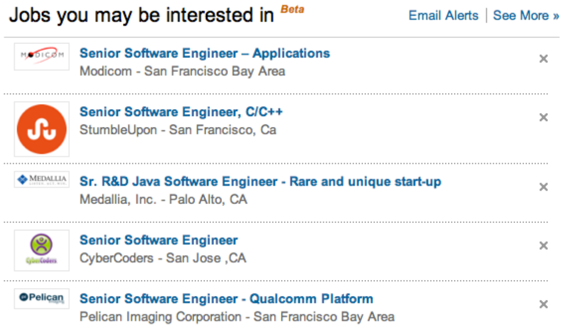
Arun C Murthy 2nd
Founder and Architect at Hortonworks Inc., VP Apache Hadoop at ASF
I am a Founder and Architect at Hortonworks Inc. Hortonworks was formed by the key architects and core Hadoop...

Related Companies

The Apache Software Foundation
Computer Software, United States
☆ Follow

Cloudera
Computer Software, San Francisco Bay Area
☆ Stop following

Jobs you may be interested in



Jobs you may be interested in *Beta* — Email Alerts | See More »

Senior Software Engineer – Applications
Modicom - San Francisco Bay Area ×

Senior Software Engineer, C/C++
StumbleUpon - San Francisco, Ca ×

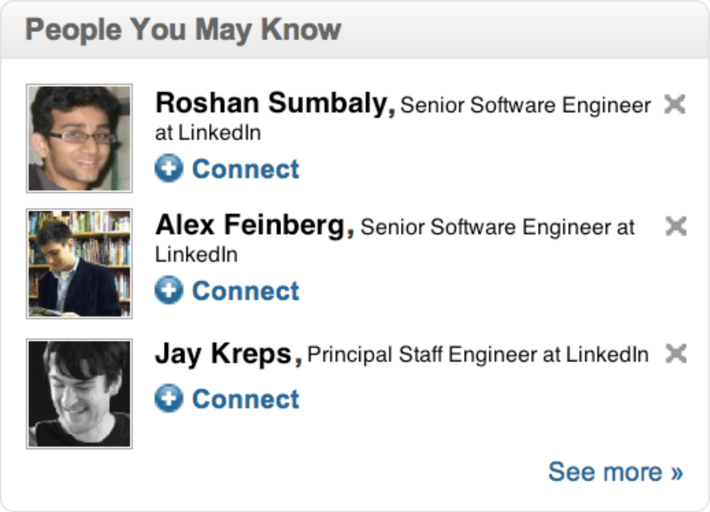
Sr. R&D Java Software Engineer - Rare and unique start-up
Medallia, Inc. - Palo Alto, CA ×

Senior Software Engineer
CyberCoders - San Jose ,CA ×

Senior Software Engineer - Qualcomm Platform
Pelican Imaging Corporation - San Francisco Bay Area ×

Data Features on LinkedIn

People You May Know



People You May Know

Roshan Sumbaly, Senior Software Engineer at LinkedIn

Connect

Alex Feinberg, Senior Software Engineer at LinkedIn

Connect

Jay Kreps, Principal Staff Engineer at LinkedIn

Connect

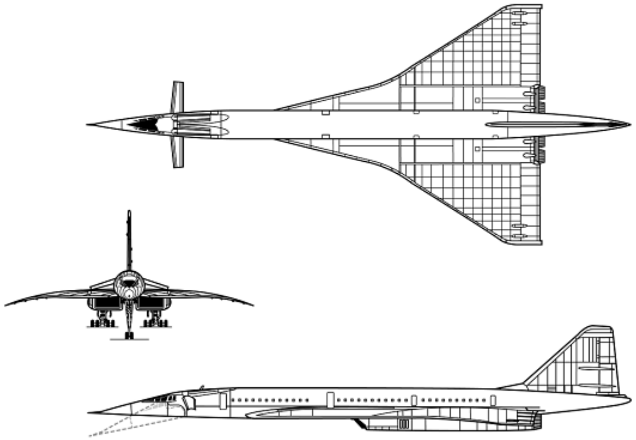
See more »

- Batch computed algorithms
  - MapReduce (Hadoop)
- Output
  - Large
  - Immutable
  - Key-value
  - Full refresh

Problem

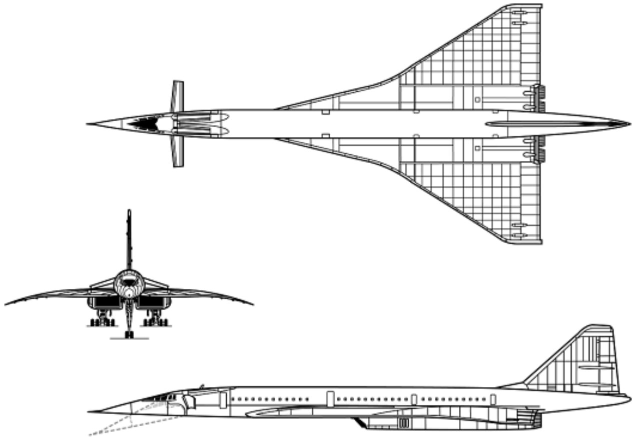
How do we serve these massive outputs to our 150 million members?

System features



- Fast, available and elastic
- Bulk load massive data-sets
- Minimum time in error
- Easy to use
- Open-source

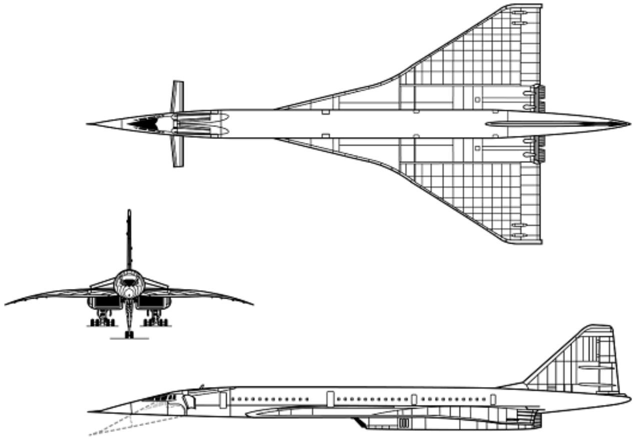
System features



- Fast, available and elastic
- Bulk load massive data-sets
- Minimum time in error
- Easy to use
- Open-source

- Distributed key-value system

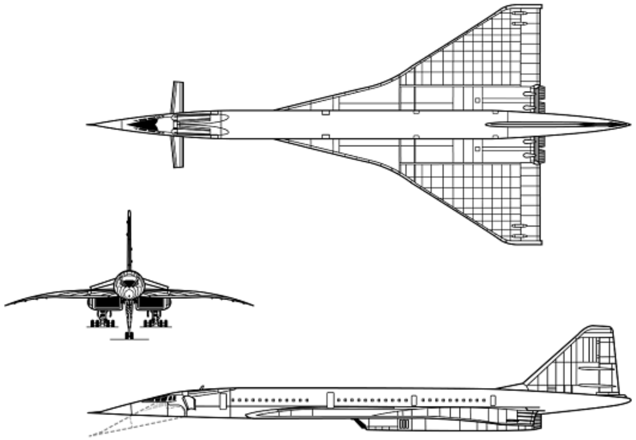
System features



- Fast, available and elastic
- Bulk load massive data-sets
- Minimum time in error
- Easy to use
- Open-source

- Minimum performance impact during bulk loads
- Offload index construction to processing system

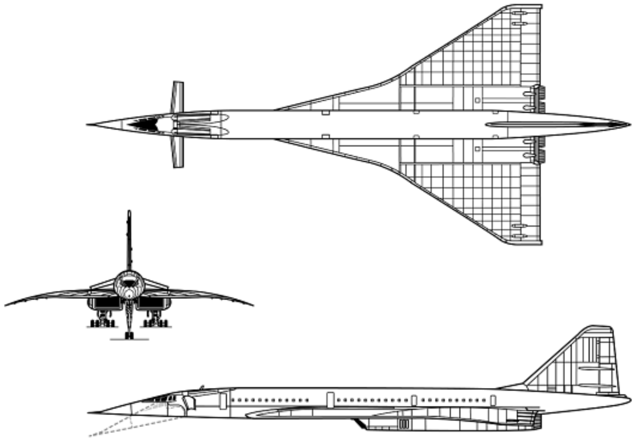
System features



- Fast, available and elastic
- Bulk load massive data-sets
- Minimum time in error
- Easy to use
- Open-source

- Error in algorithm → Bulk load bad data → Bad state till next push
- Quick rollback capability

System features

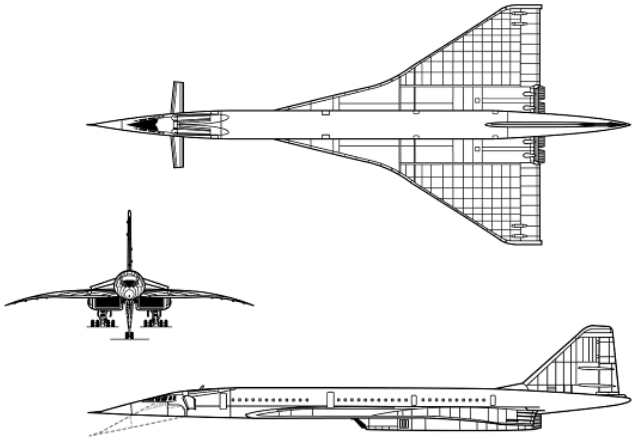


- Fast, available and elastic
- Bulk load massive data-sets
- Minimum time in error
- **Easy to use**
- Open-source

```
job.class=com.linkedin.jobs.BuildAndPushJob

build.input.path=/algorithm/output
push.store.name=people-you-may-know
push.cluster=tcp://testing-cluster-url:6666
build.replication.factor=1
```

System features

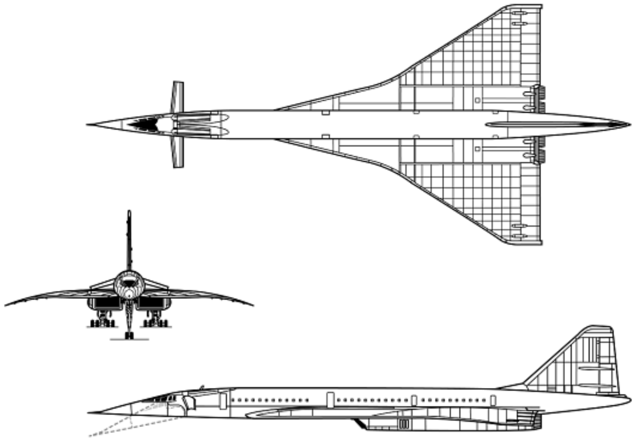


- Fast, available and elastic
- Bulk load massive data-sets
- Minimum time in error
- Easy to use
- Open-source

```
job.class=com.linkedin.jobs.BuildAndPushJob

build.input.path=/algorithm/output
push.store.name=people-you-may-know
push.cluster=tcp://production-cluster-url:6666
build.replication.factor=2
```

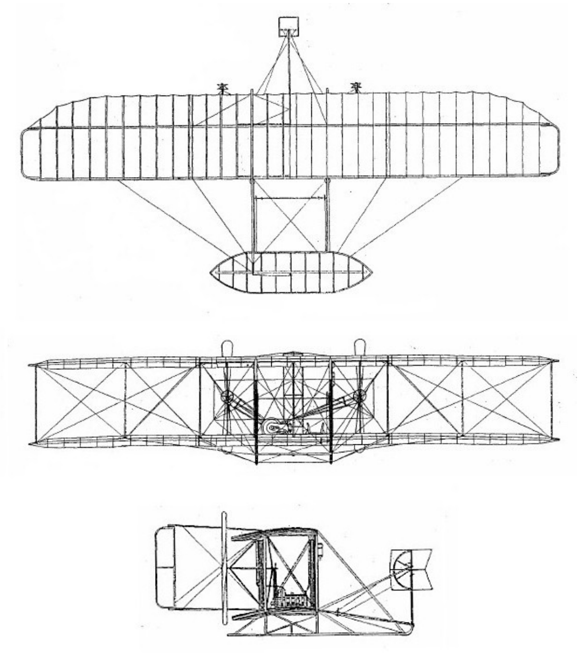
System features



- Fast, available and elastic
- Bulk load massive data-sets
- Minimum time in error
- Easy to use
- Open-source

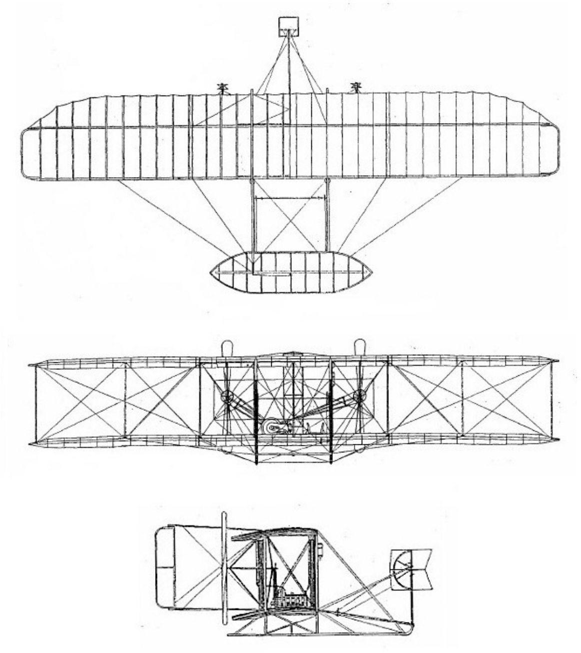
- Apache License v2.0
- Project Voldemort – http://project-voldemort.com

Roadmap



- Background – Voldemort architecture
- Custom Voldemort storage engine
    - Minimal impact on live system
    - Fast rollback
    - Fast lookups
    - Easy rebalancing
- Performance

Roadmap

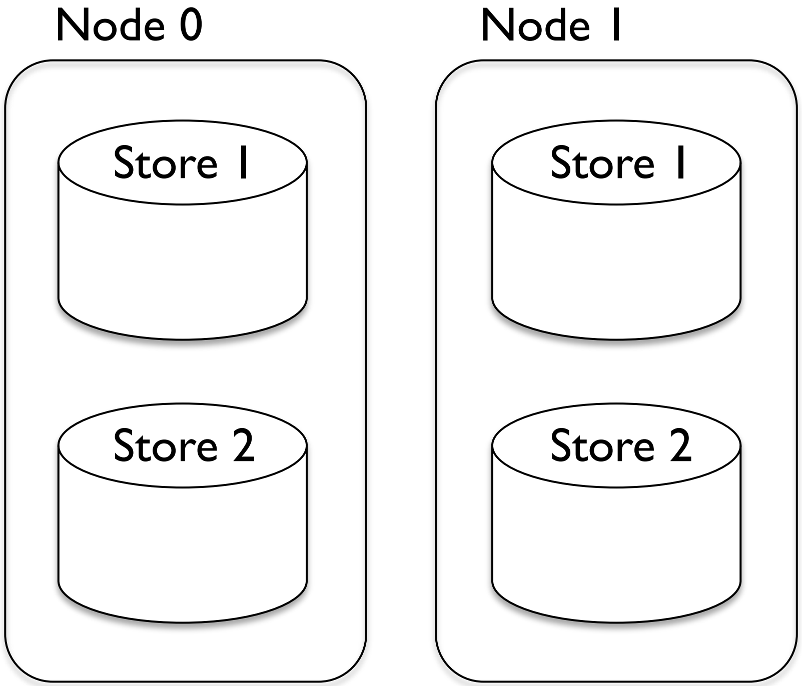


- Background – Voldemort architecture
- Custom Voldemort storage engine
    - Minimal impact on live system
    - Fast rollback
    - Fast lookups
    - Easy rebalancing
- Performance

Project Voldemort

- Amazon’s Dynamo clone
    - Distributed key-value store
- Pluggable architecture
- Initially written for read-write storage engines
    - MySQL, BDB

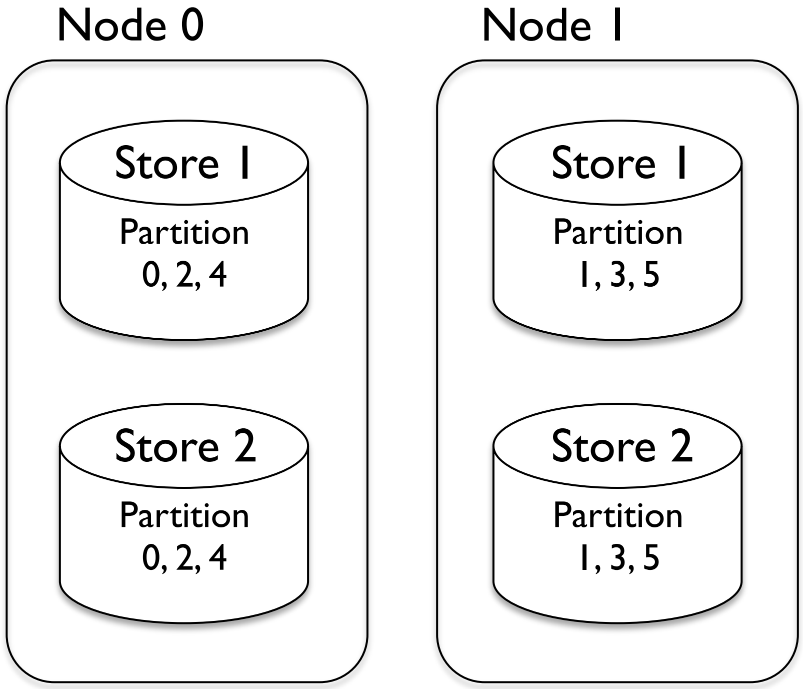
Project Voldemort

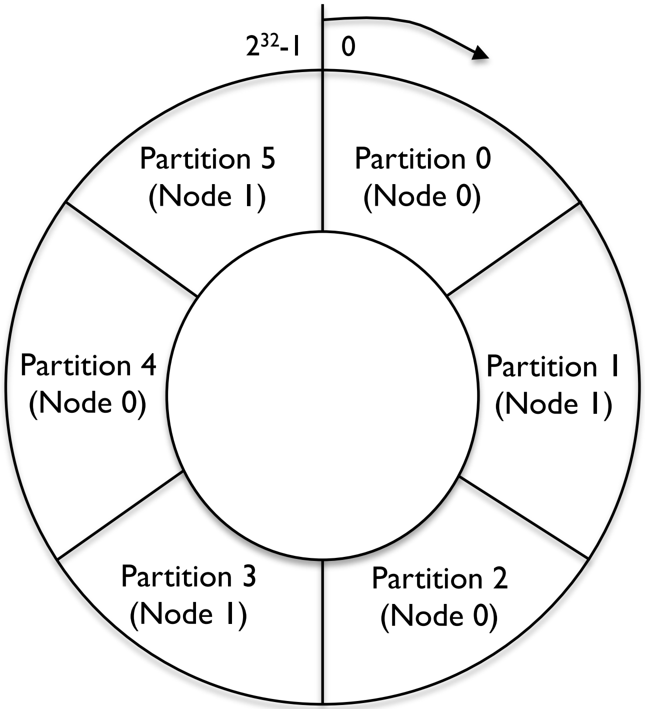


- Multiple peers
- Multiple stores (~ tables)
- Different replication factor per store

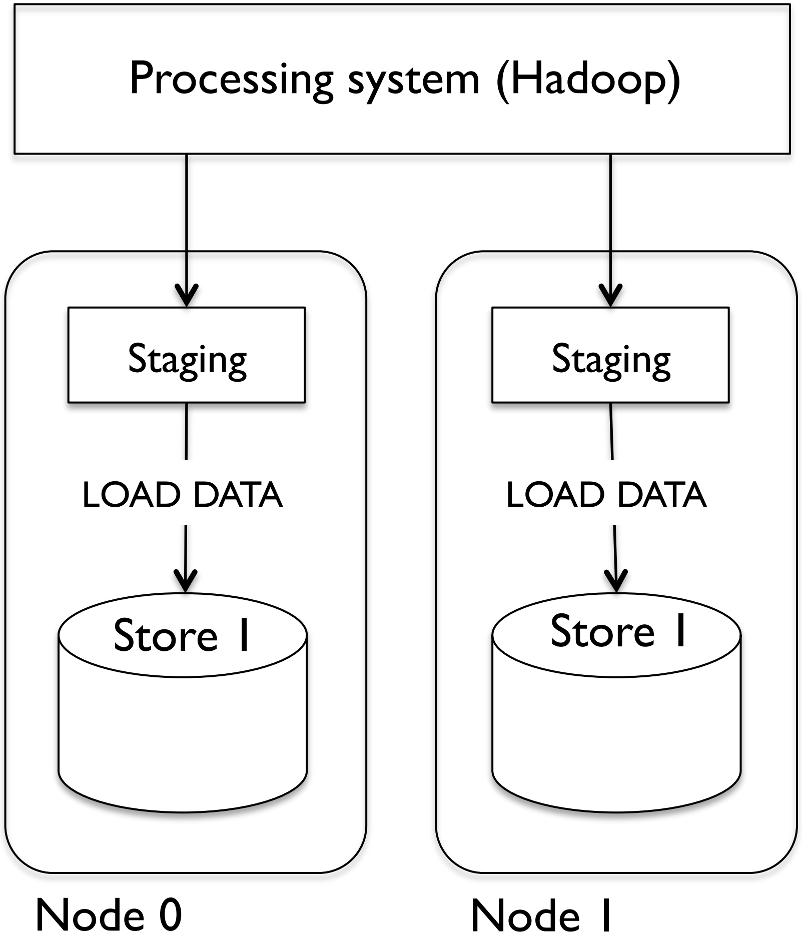
Project Voldemort

- Hash ring per store
- Ring split into partitions



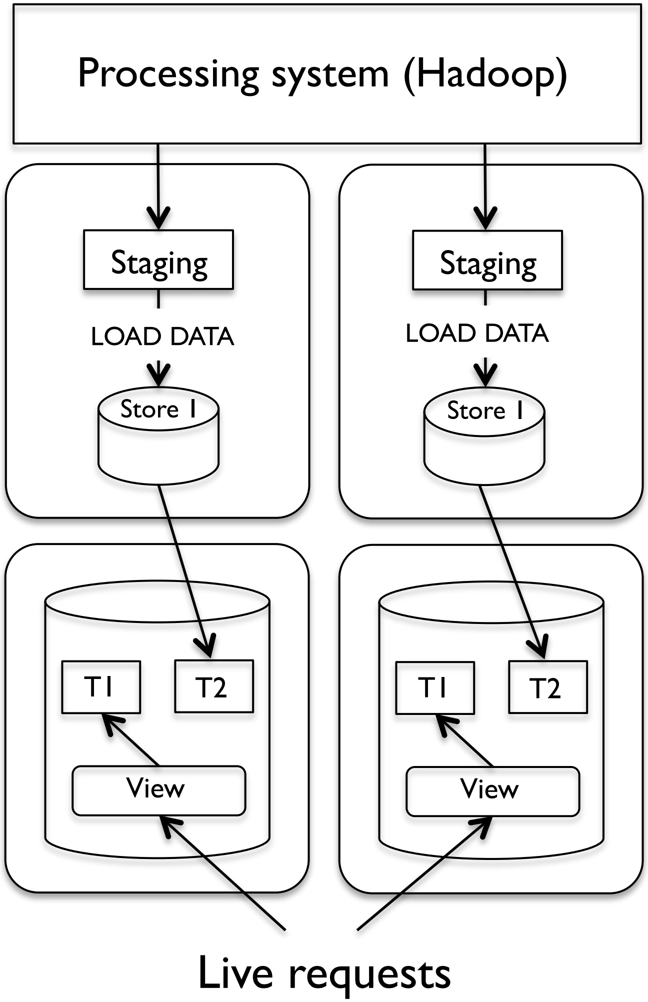


Existing approaches – Bulk load solution 1



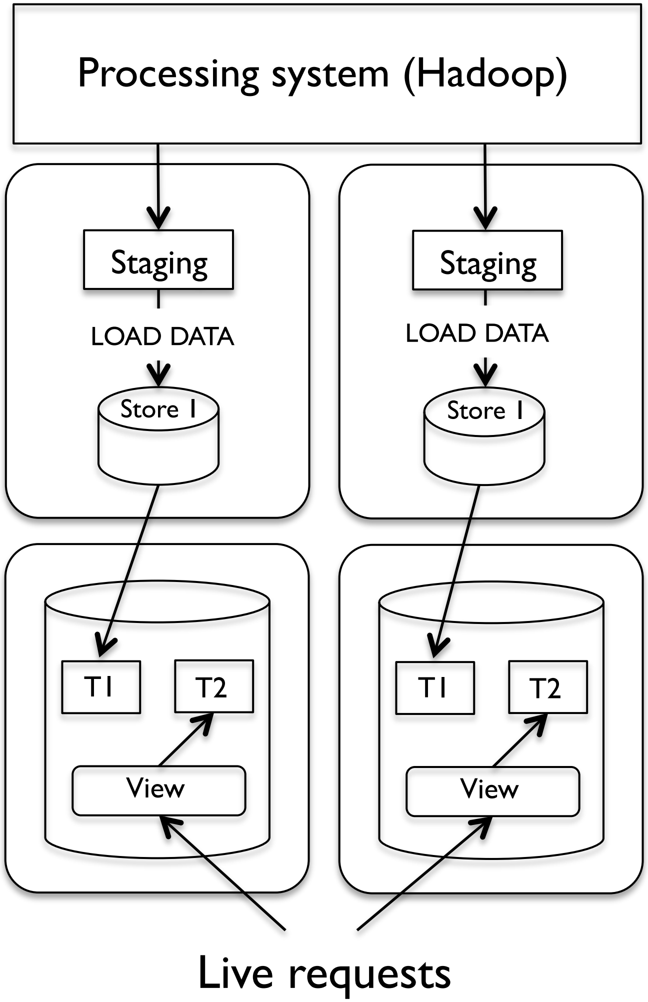
- Latency degradation during load

Existing approaches – Bulk load solution 2



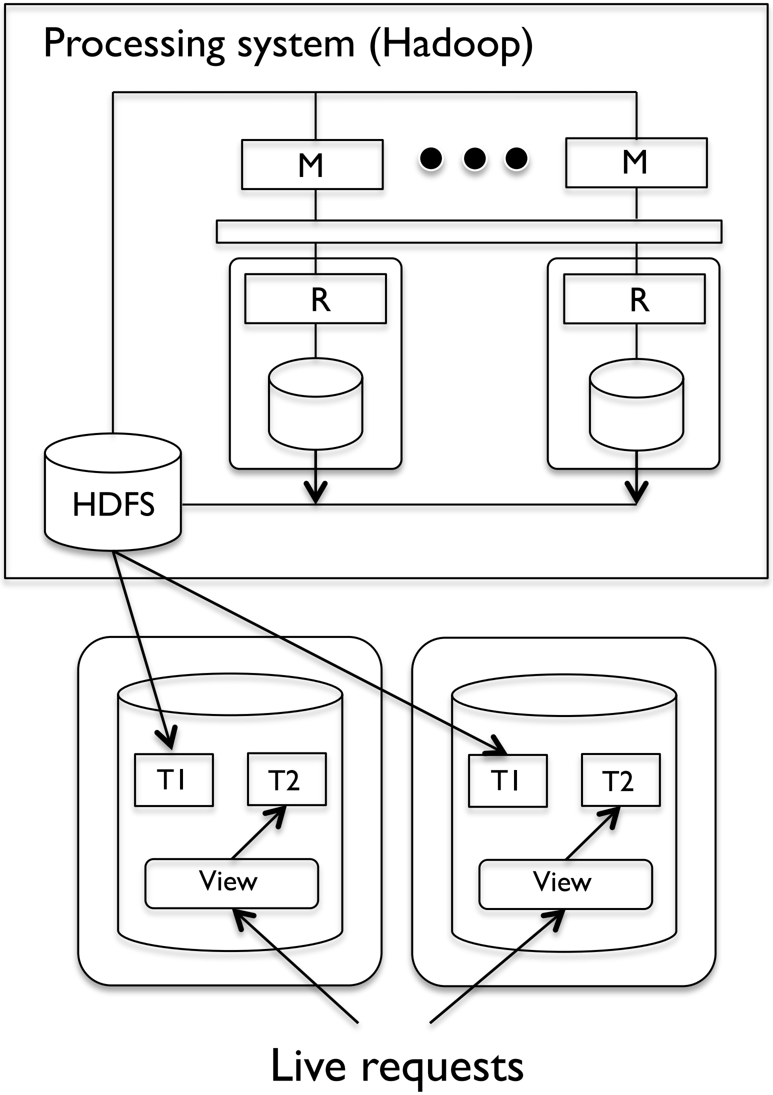
- Maintaining extra cluster

Existing approaches – Bulk load solution 2



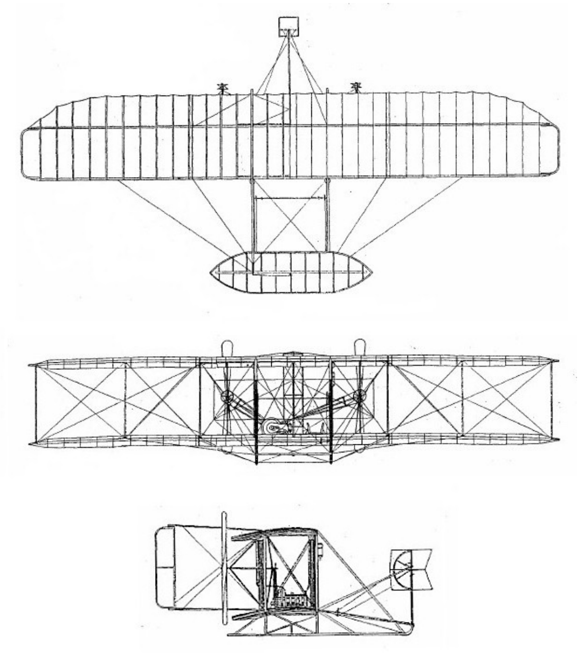
- Maintaining extra cluster

Existing approaches – Bulk load solution 3



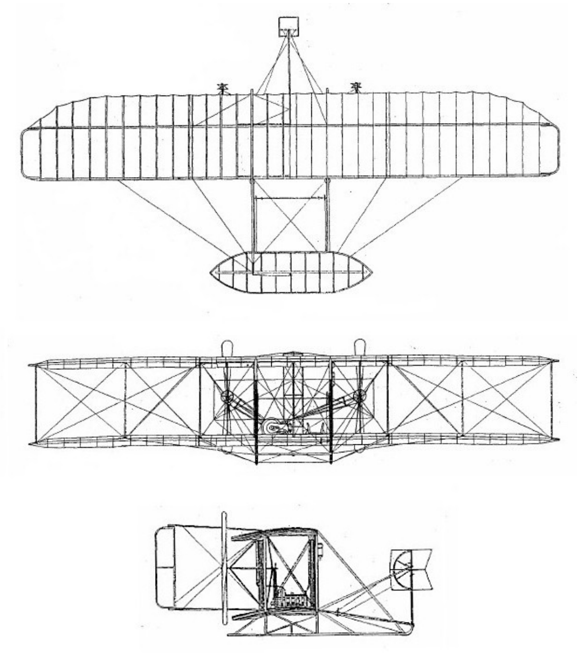
- Multiple copy operations

Roadmap



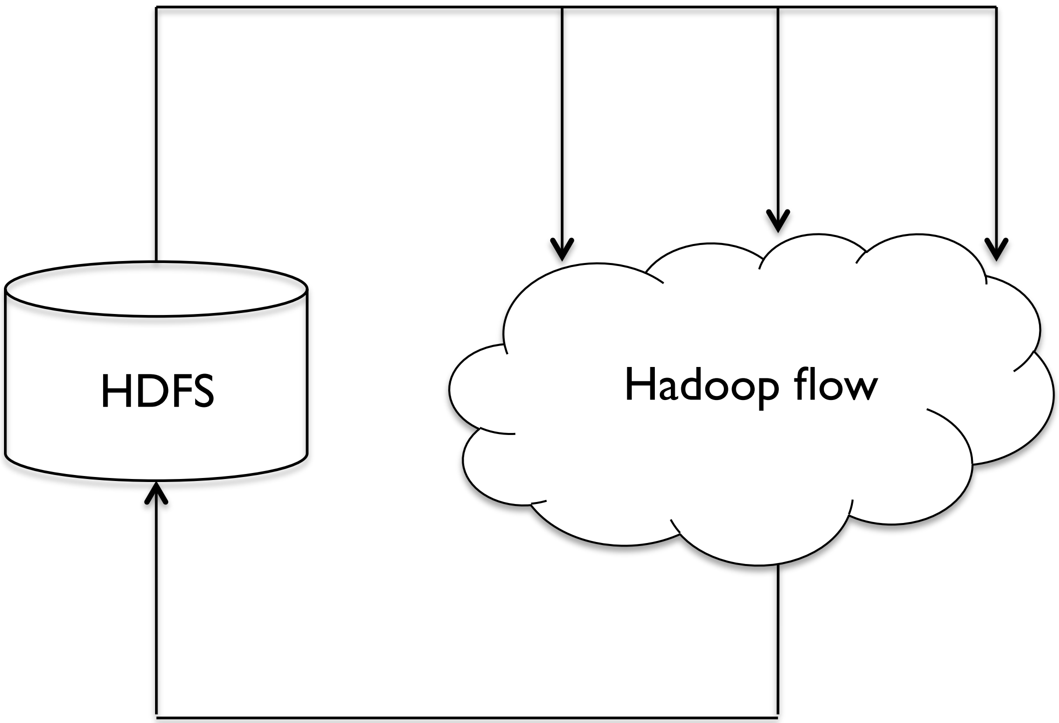
- Background – Voldemort architecture
- Custom Voldemort storage engine
  - Minimal impact on live system
  - Fast rollback
  - Fast lookups
  - Easy rebalancing
- Performance

Roadmap

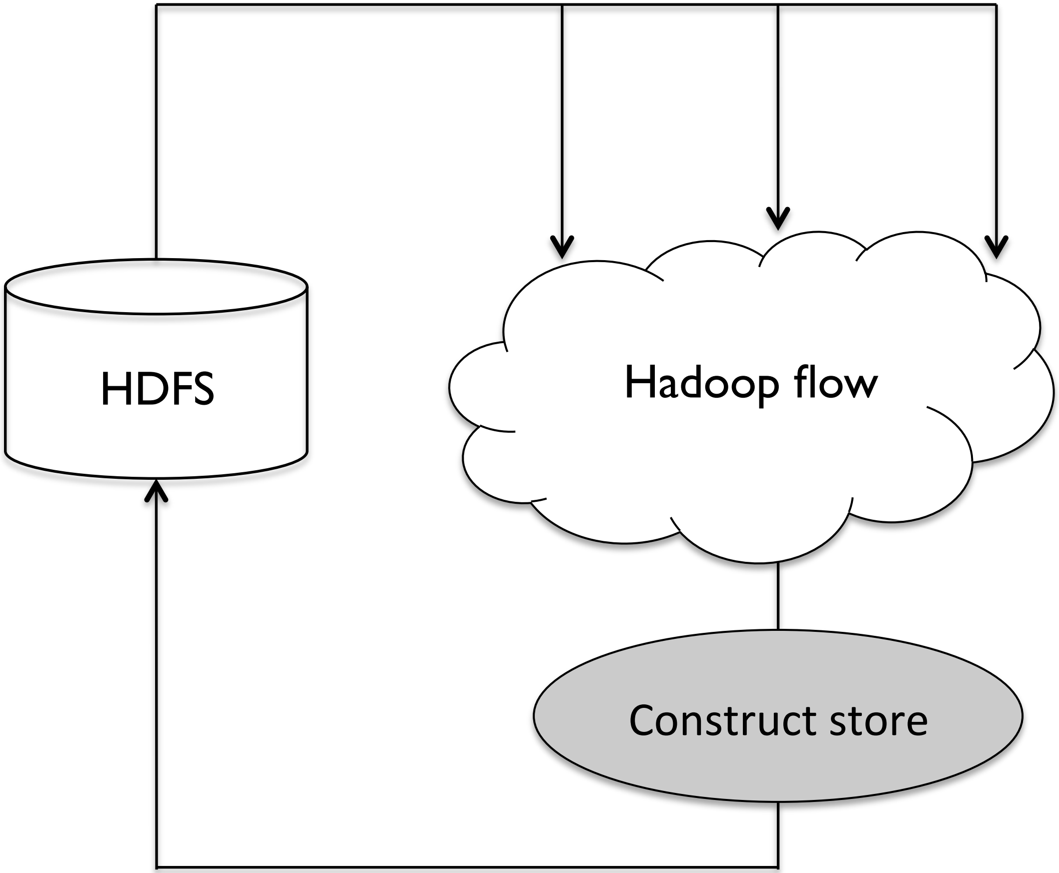


- Background – Voldemort architecture
- Custom Voldemort storage engine
  - Minimal impact on live system
  - Fast rollback
  - Fast lookups
  - Easy rebalancing
- Performance

Bulk load extensions – Minimal impact on live system



Bulk load extensions – Minimal impact on live system

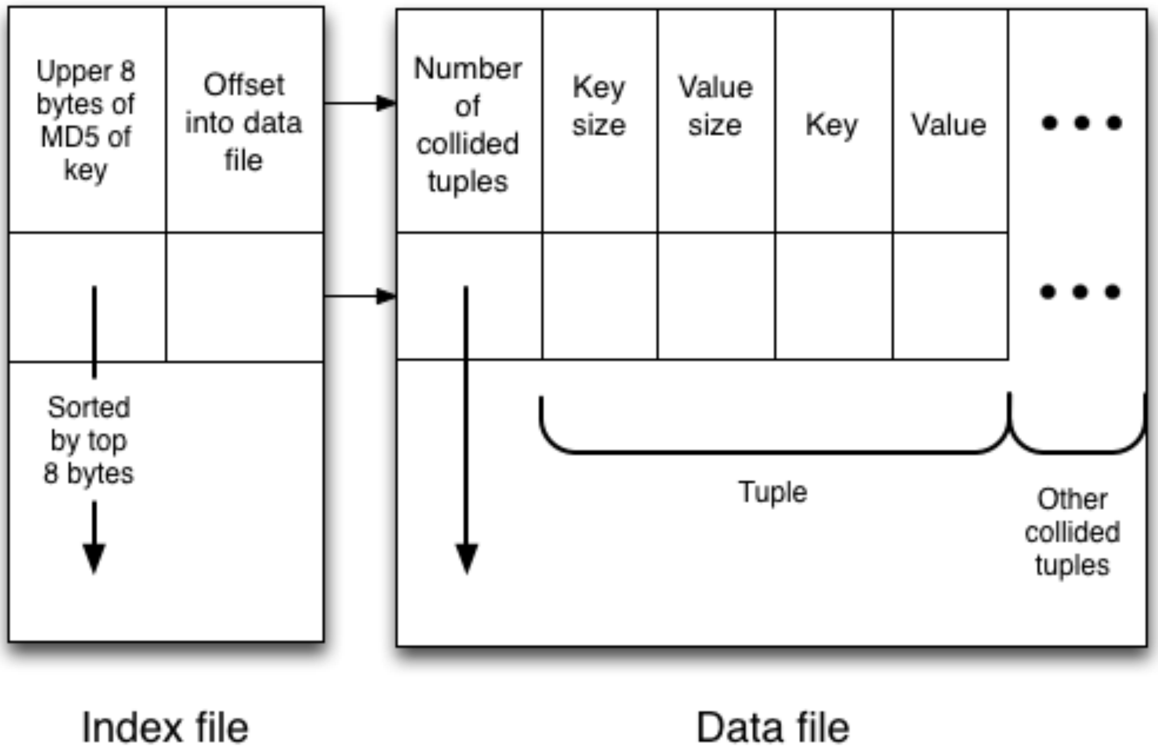


Bulk load extensions – Minimal impact on live system

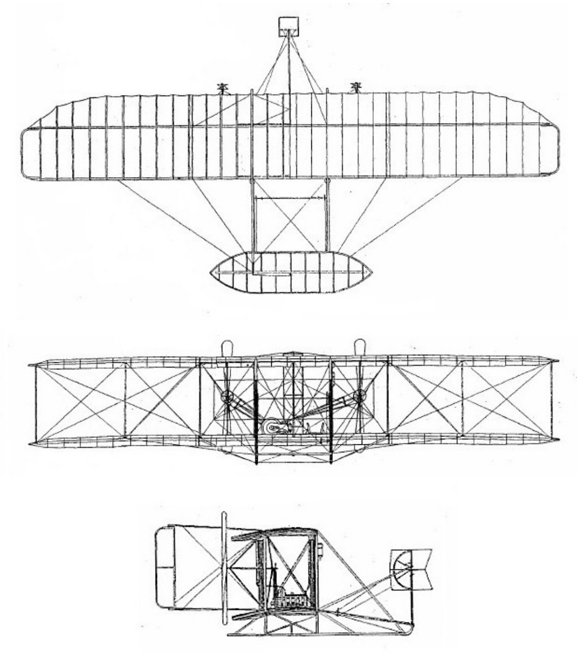
- “Construct Store” step
  - Single MapReduce job
  - Map
    - Input - Output of algorithm
    - Output – Emit replication factor number of times
  - Partitioner
    - Redirect to appropriate reducer
  - Reducer
    - Output to Voldemort node based folders

Bulk load extensions – Minimal impact on live system

- What is output in the reducer phase?
  - Store $\xrightarrow{\text{split into}}$ Partitions $\xrightarrow{\text{split into}}$ Chunk sets
  - One reducer = one chunk set
  - Chunk set = Index + data file



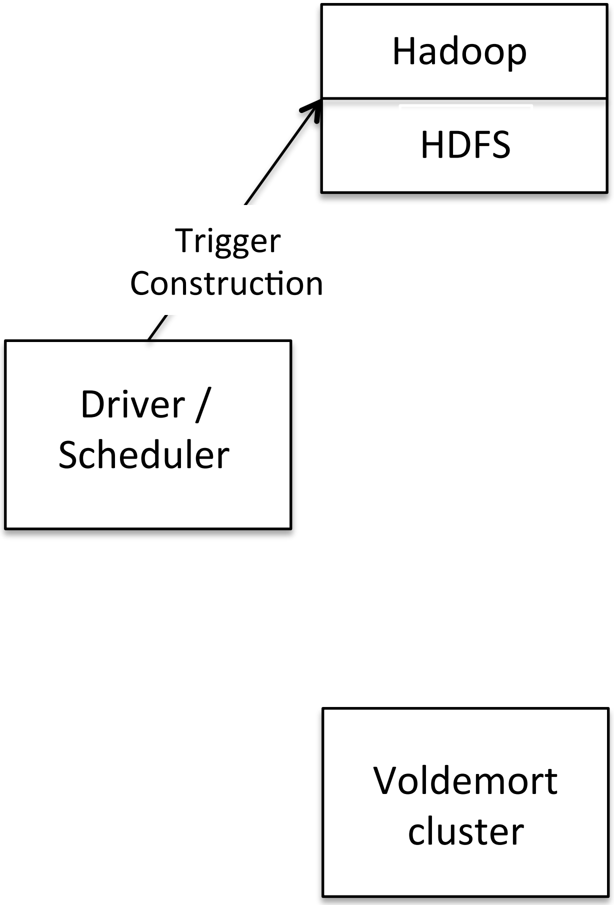
Roadmap



- Background – Voldemort architecture
- Custom Voldemort storage engine
  - Minimal impact on live system
  - Fast rollback
  - Fast lookups
  - Easy rebalancing
- Performance

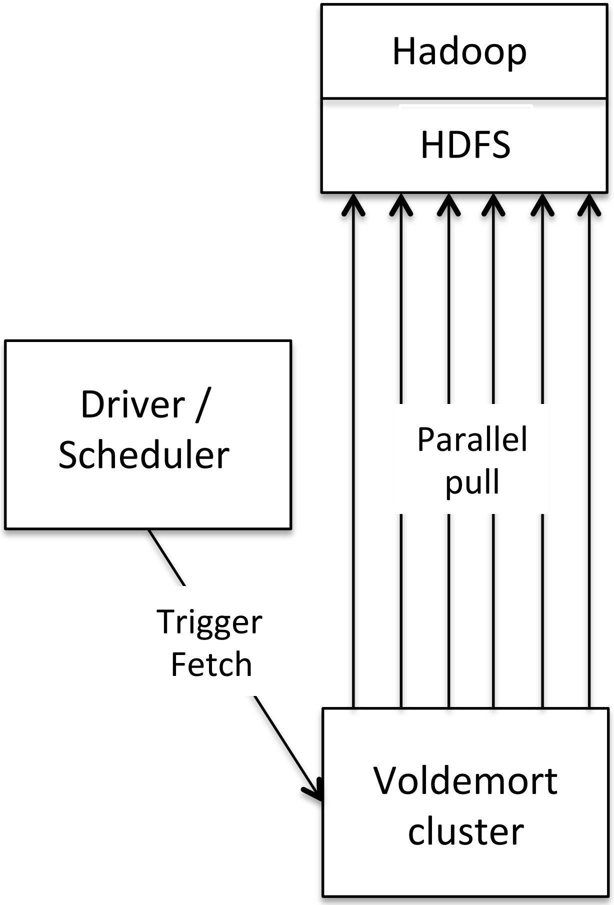
Bulk load extensions – Full pipeline

- Construction
  - MapReduce job
- Fetch
  - Pull chunk sets in parallel
  - Store into new version folder
- Swap
  - Close latest version's index files
  - Change latest version link
  - Memory map new version's index files
- Rollback
  - Close latest version's index file
  - Change latest version link
  - Memory map old version's index file

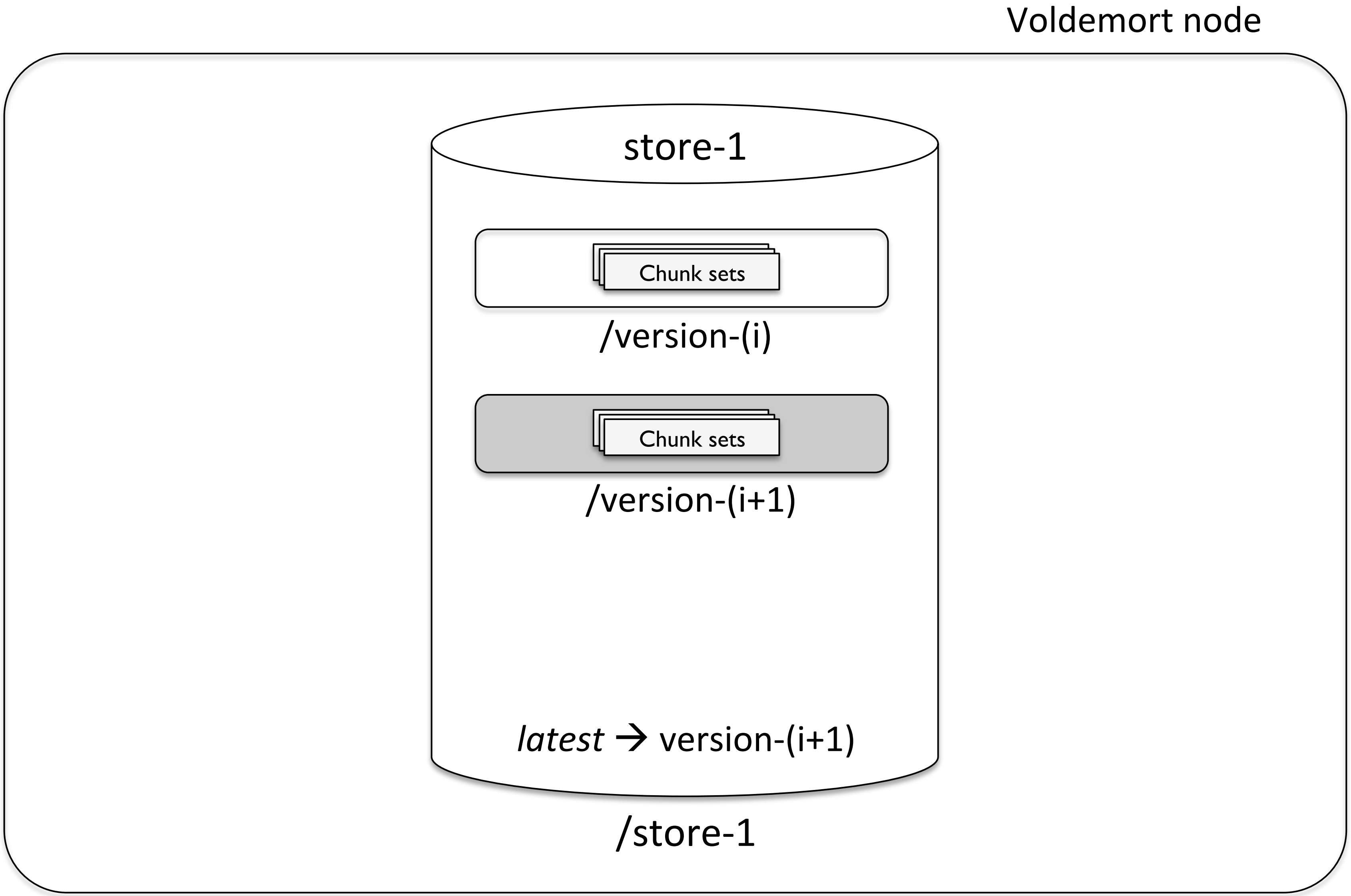


Bulk load extensions – Full pipeline

- Construction
  - MapReduce job
- Fetch
  - Pull chunk sets in parallel
  - Store into new version folder
- Swap
  - Close latest version’s index files
  - Change latest version link
  - Memory map new version’s index files
- Rollback
  - Close latest version’s index file
  - Change latest version link
  - Memory map old version’s index file

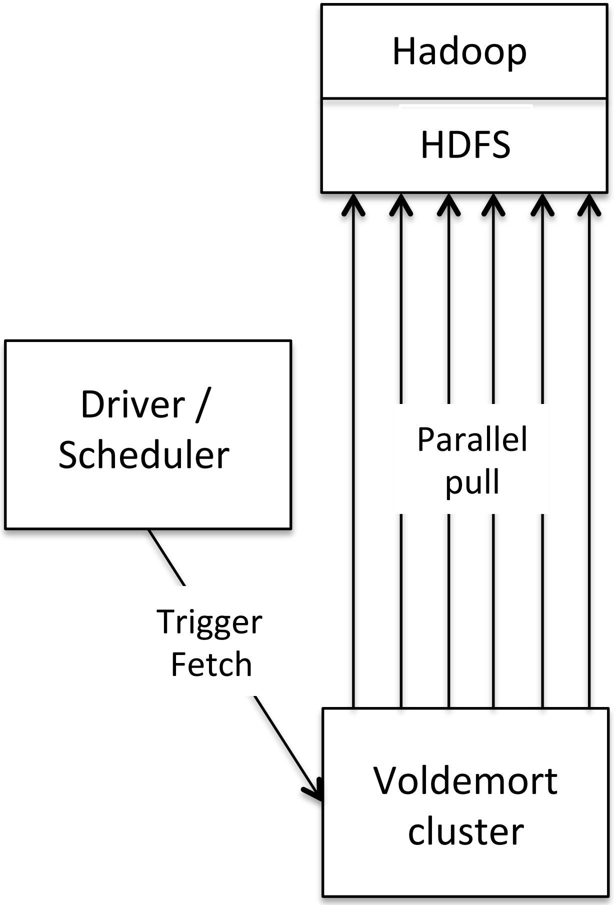


Bulk load extensions - Rollback



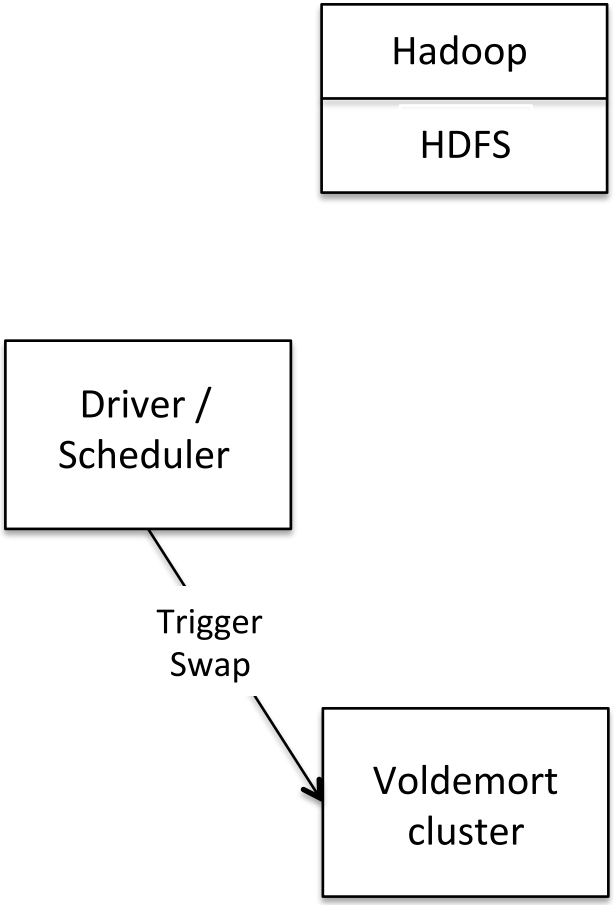
Bulk load extensions – Full pipeline

- Construction
  - MapReduce job
- Fetch
  - Pull chunk sets in parallel
  - Store into new version folder
- Swap
  - Close latest version's index files
  - Change latest version link
  - Memory map new version's index files
- Rollback
  - Close latest version's index file
  - Change latest version link
  - Memory map old version's index file

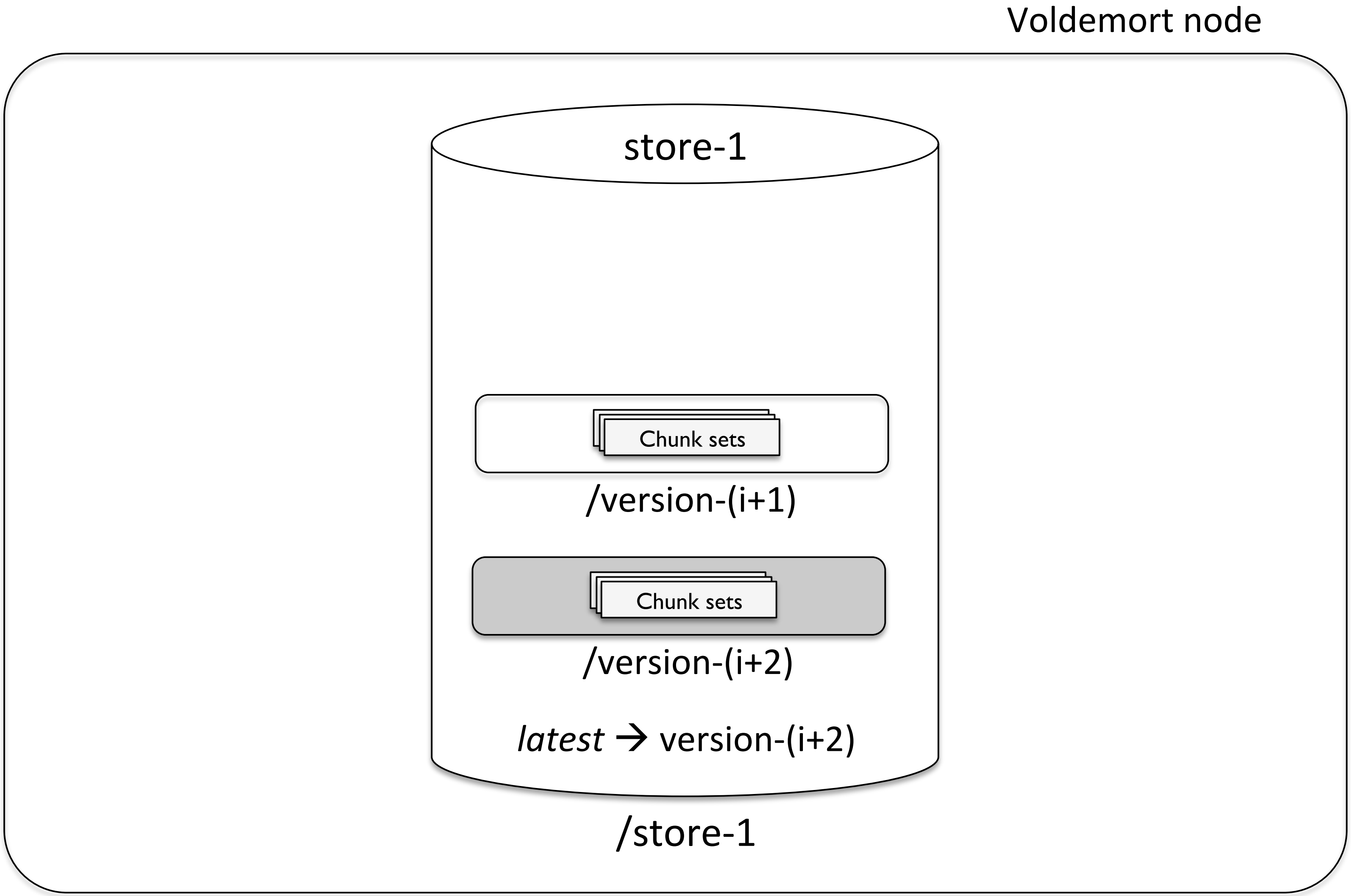


Bulk load extensions – Full pipeline

- Construction
  - MapReduce job
- Fetch
  - Pull chunk sets in parallel
  - Store into new version folder
- Swap
  - Close latest version's index files
  - Change latest version link
  - Memory map new version's index files
- Rollback
  - Close latest version's index file
  - Change latest version link
  - Memory map old version's index file

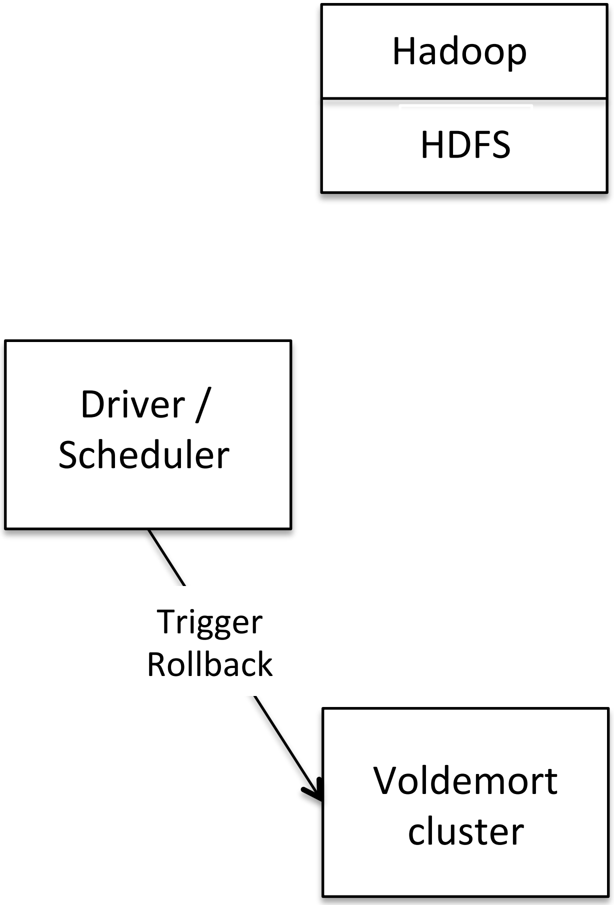


Bulk load extensions - Rollback

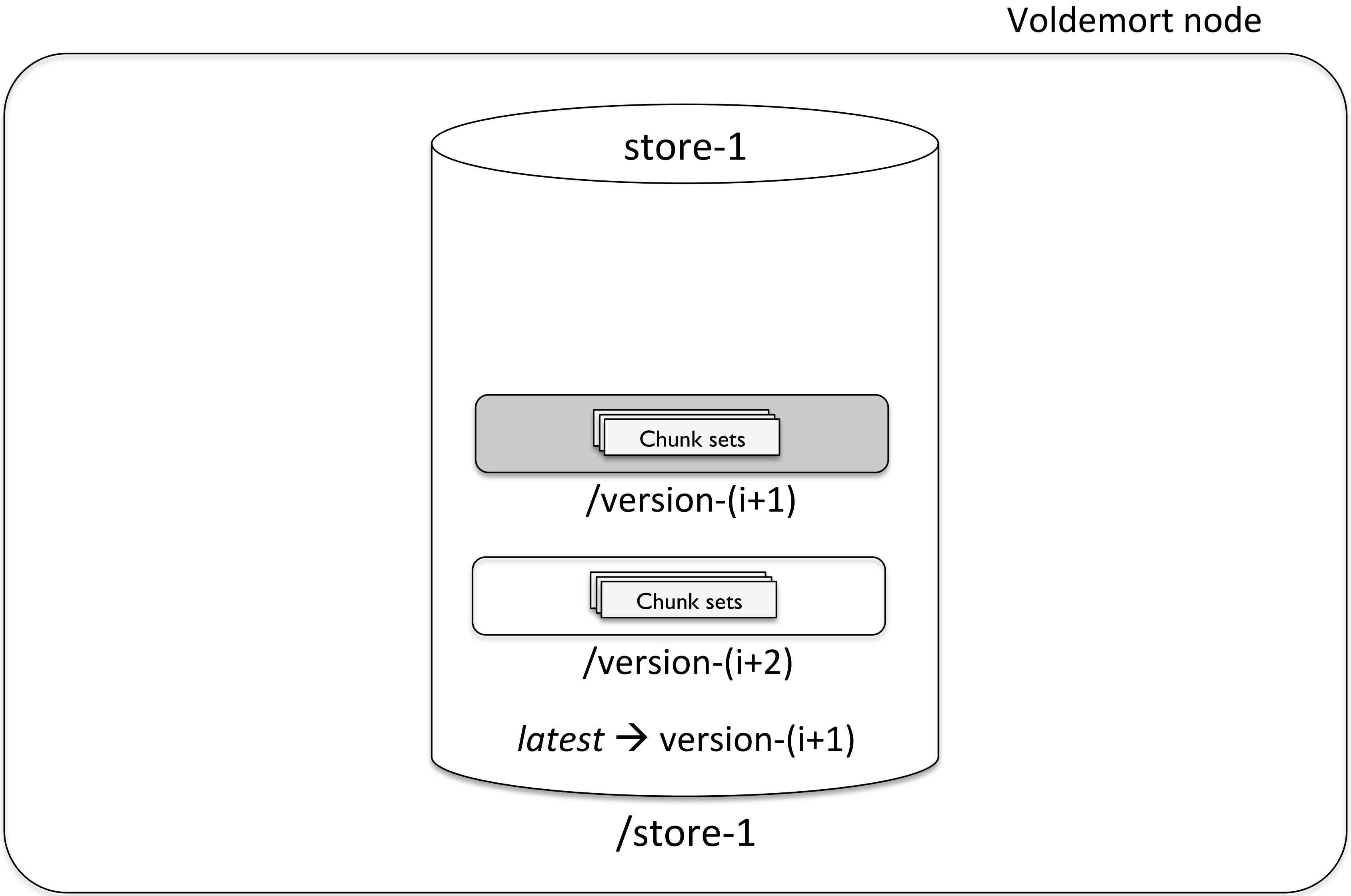


Bulk load extensions – Full pipeline

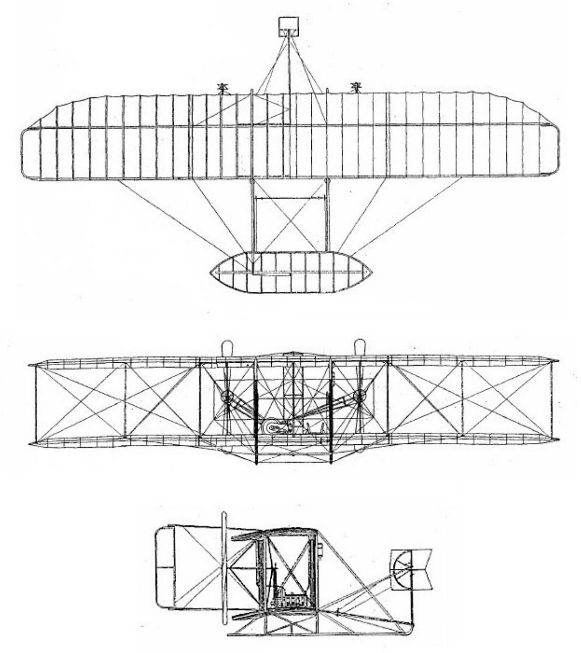
- Construction
  - MapReduce job
- Fetch
  - Pull chunk sets in parallel
  - Store into new version folder
- Swap
  - Close latest version's index files
  - Change latest version link
  - Memory map new version's index files
- Rollback
  - Close latest version's index file
  - Change latest version link
  - Memory map old version's index file



Bulk load extensions - Rollback



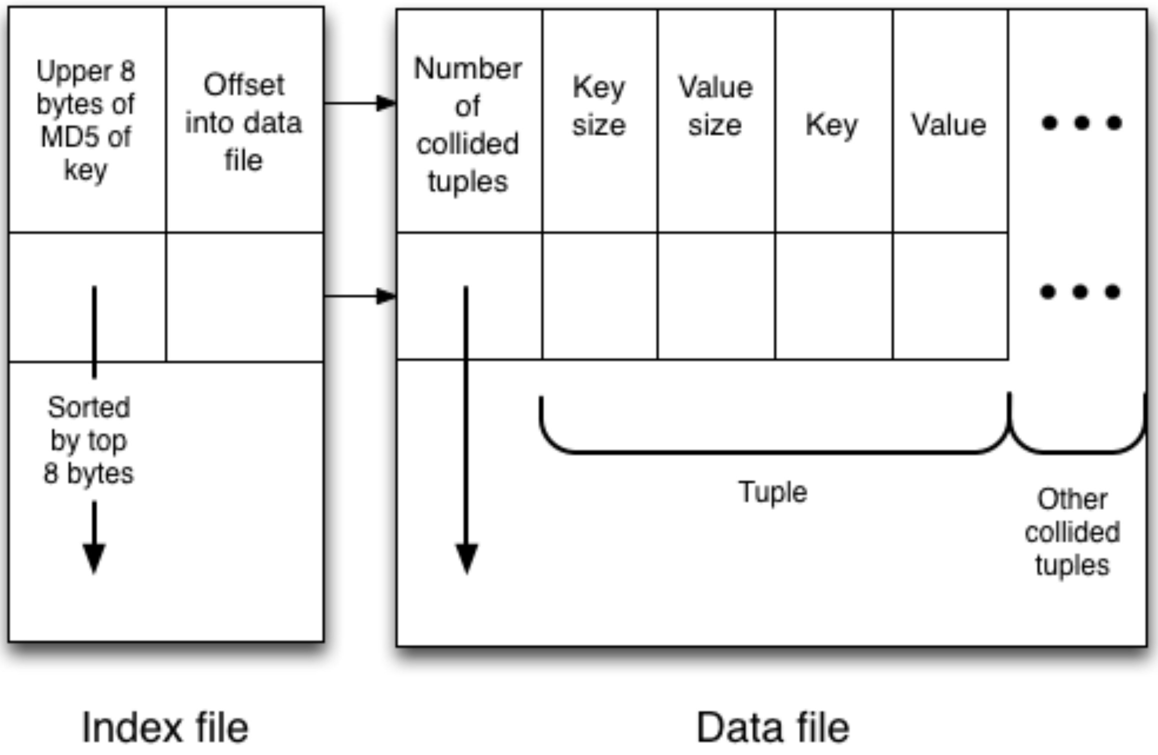
Roadmap



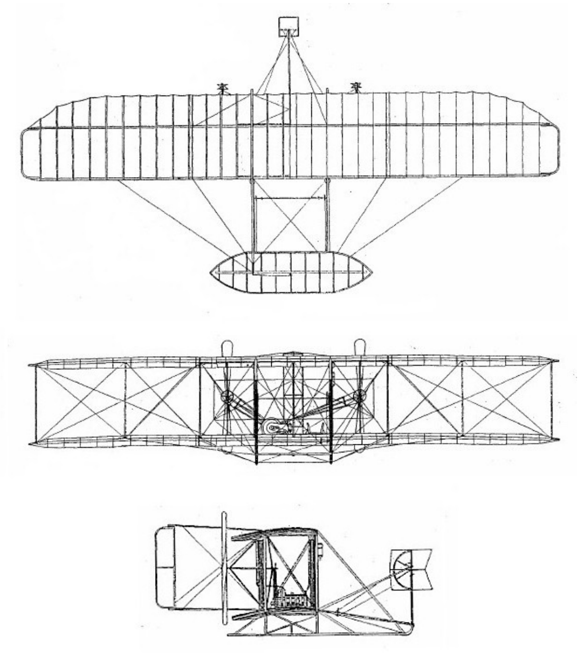
- Background – Voldemort architecture
- Custom Voldemort storage engine
  - Minimal impact on live system
  - Fast rollback
  - Fast lookups
  - Easy rebalancing
- Performance

Bulk load extensions – Lookup

- Find partition and chunk set to read
- Binary search in index file of chunk set
- Jump to offset in data file
- Go through all collided tuples



Roadmap

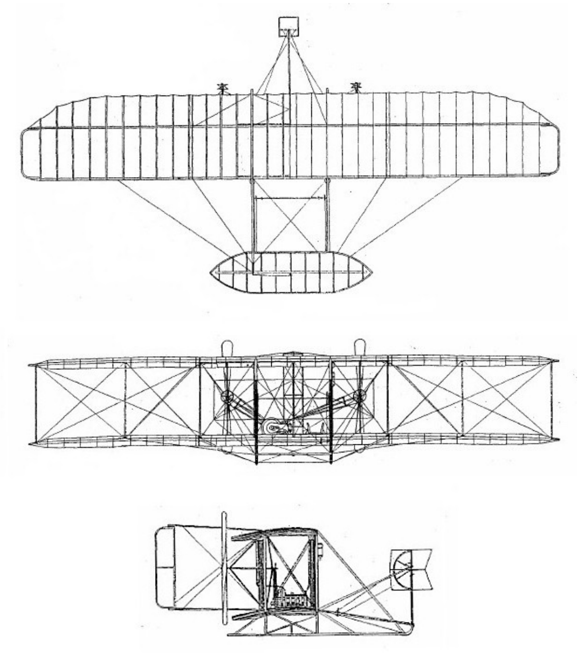


- Background – Voldemort architecture
- Custom Voldemort storage engine
  - Minimal impact on live system
  - Fast rollback
  - Fast lookups
  - Easy rebalancing
- Performance

Bulk load extensions – Rebalancing

- Adding new nodes with no downtime
- Change ownership of partitions to new nodes
    - Simple move of corresponding chunk sets + swap

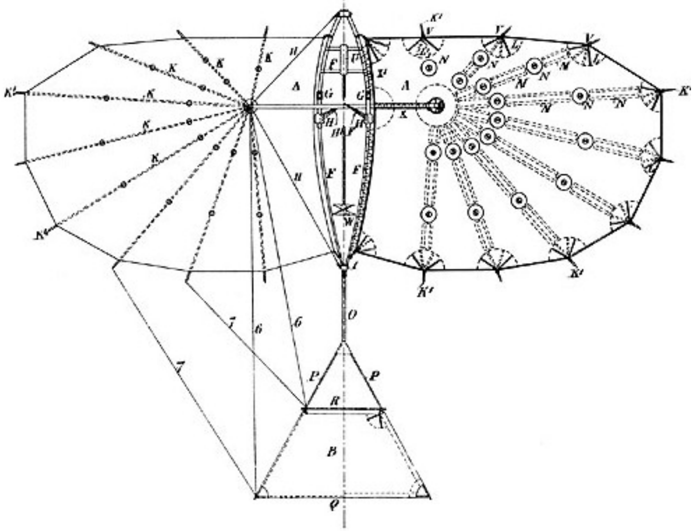
Roadmap



- Background – Voldemort architecture
- Custom Voldemort storage engine
  - Minimal impact on live system
  - Fast rollback
  - Fast lookups
  - Easy rebalancing
- Performance

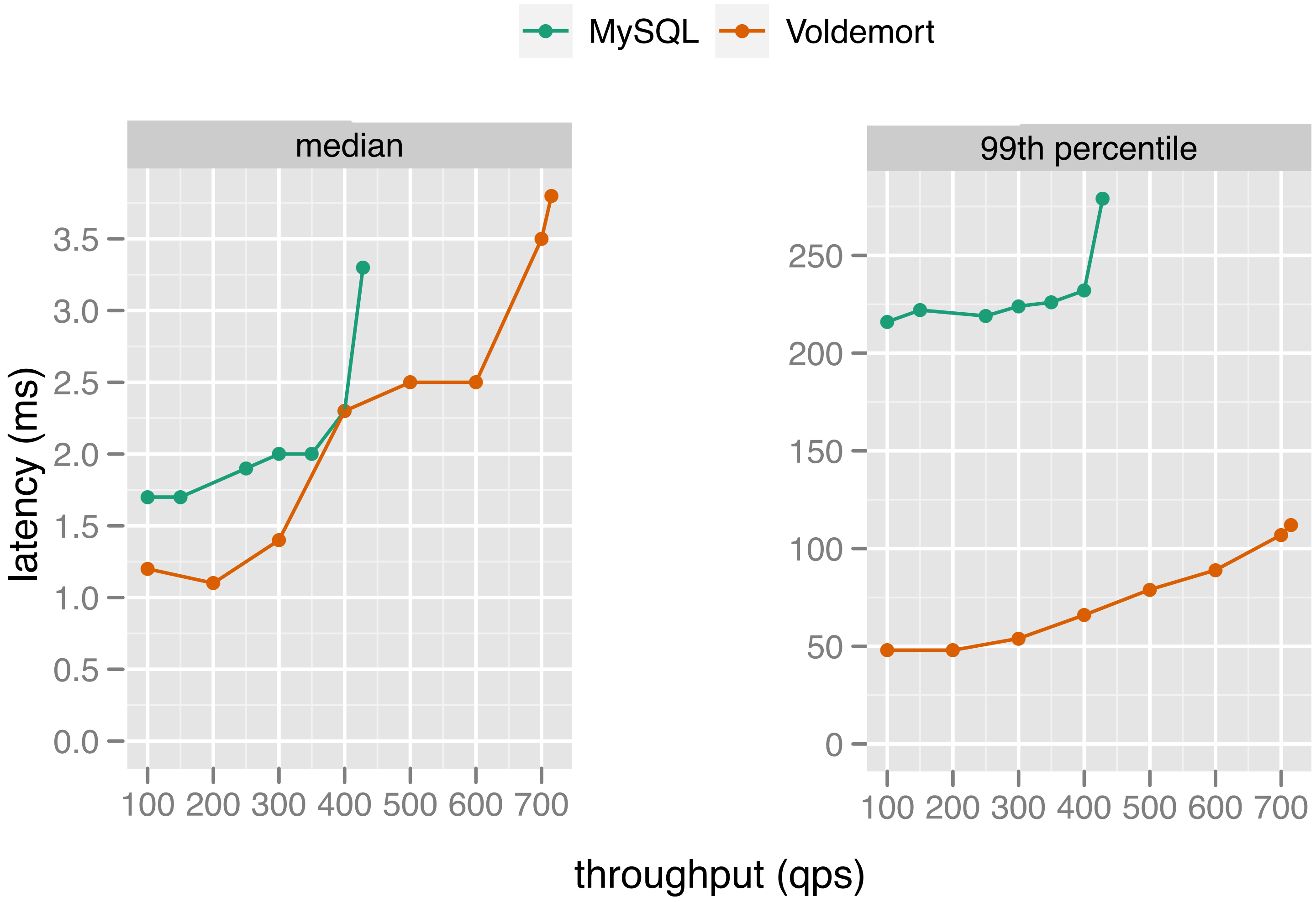
Performance

Fig. 4.



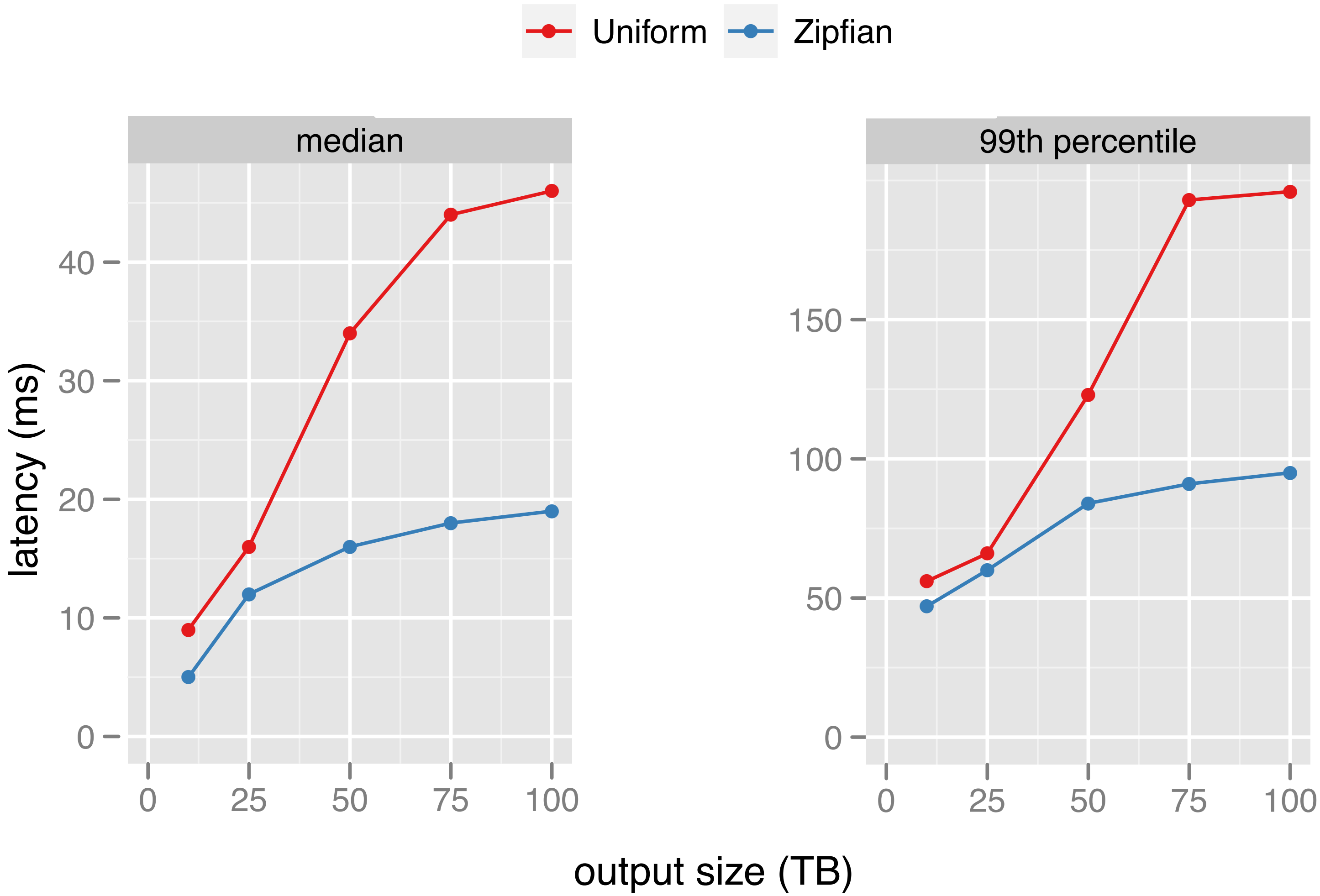
- Single node latency
- Multi-node latency
- Production

Performance – Single node latency



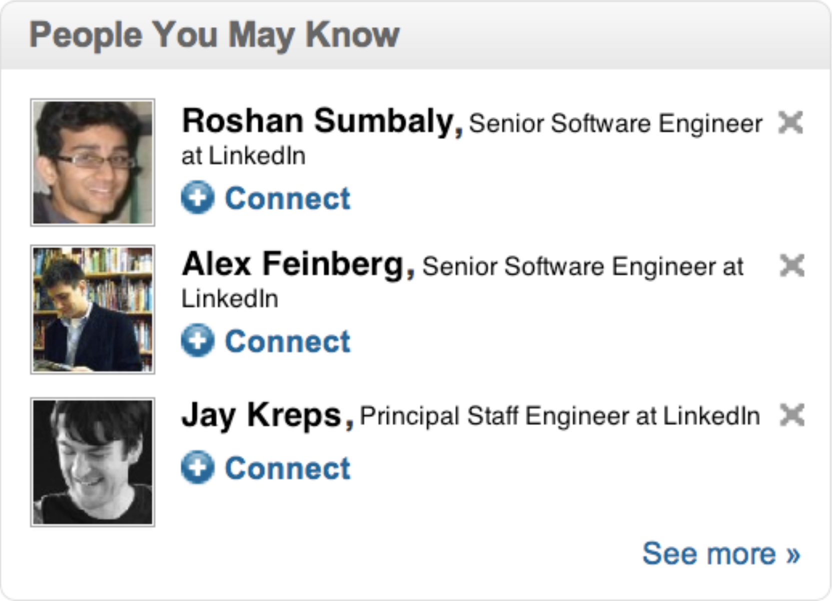
100 GB data, 24 GB RAM

Performance – Multi-node latency



32 nodes client side latency

Performance – Production



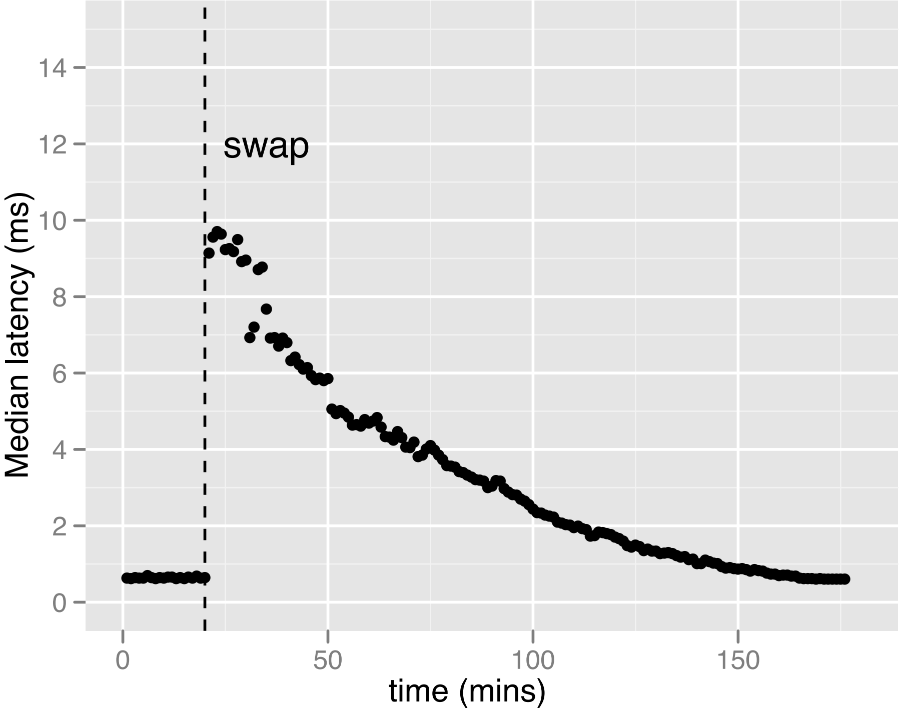
People You May Know

Roshan Sumbaly, Senior Software Engineer at LinkedIn
Connect

Alex Feinberg, Senior Software Engineer at LinkedIn
Connect

Jay Kreps, Principal Staff Engineer at LinkedIn
Connect

See more »



People You May Know – 25 nodes client side latency

Related Work

- Shared nothing databases
  - Yahoo’s PNUTS [Silberstein08]
    - Bulk load into range partitioned tables
    - Requires some compaction on serving system
  - HBase [Konstantinou11], Cassandra [Lebresne11]
    - Offline tablet construction
    - Expensive rollback
- Search systems
  - Build index offline and pull
    - MapReduce[Dean04] use-case

Conclusion

- LinkedIn
  - Serving ~120 stores in production for past 2 years
  - Fetching ~ 4 TB of data every day
  - 76 stores swapped every day
- Open-source
  - http://project-voldemort.com

Bibliography

Images

- Slide 6 - www.aviastar.org/air/inter/inter_concorde.php
- Slide 15 - www.wright-brothers.org/Information_Desk/Just_the_Facts/Airplanes/Flyer_II.htm
- Slide 17 - www.youtube.com/watch?v=oz-7wJJ9HZ0
- Slide 48 - www.wright-brothers.org/History_Wing/History_of_the_Airplane/Century_Before/Road_to_Kitty_Hawk/Road_to_Kitty_Hawk.htm

Papers

[1] Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall, and Werner Vogels. 2007. Dynamo: Amazon's Highly Available Key-value Store. In Proceedings of 21st ACM SIGOPS symposium on Operating systems principles (SOSP '07)

[2] Jeffrey Dean and Sanjay Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. In Proceedings of the 6th Conference on Symposium on Operating Systems Design & Implementation (OSDI '04)

[3] Adam Silberstein, Brian F. Cooper, Utkarsh Srivastava, Erik Vee, Ramana Yerneni, and Raghu Ramakrishnan. 2008. Efficient bulk insertion into a distributed ordered table. In Proceedings of the 2008 ACM SIGMOD international conference on Management of data (SIGMOD '08)

[4] Ioannis Konstantinou, Evangelos Angelou, Dimitrios Tsoumakos, and Nectarios Koziris. Distributed Indexing of Web Scale Datasets for the Cloud. In Proceedings of the 2010 Workshop on Massive Data Analytics on the Cloud (MDAC '10)

[5] Sasha Pachev. Understanding MySQL Internals. O'Reilly Media, 2007.

[6] Tom White. Hadoop: The Definite Guide. O'Reilly Media, 2010

[7] Sylvain Lebresne. Using the Cassandra Bulk Loader. http://www.datastax.com/dev/blog/bulk-loading