Mambo
Running Analytics on Enterprise Storage

Jingxin Feng, Xing Lin[1], Gokul Soundararajan

Advanced Technology Group

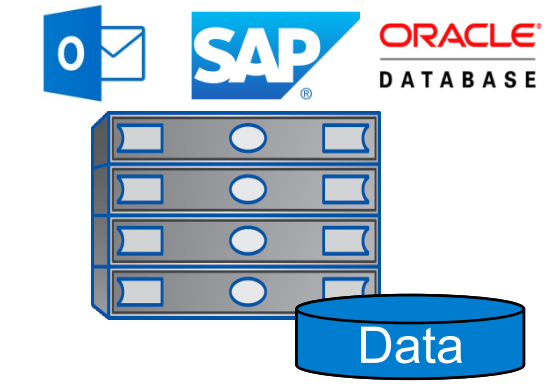
[1] University of Utah

Motivation

No easy way to analyze data stored in enterprise storage (NFS)

Motivation

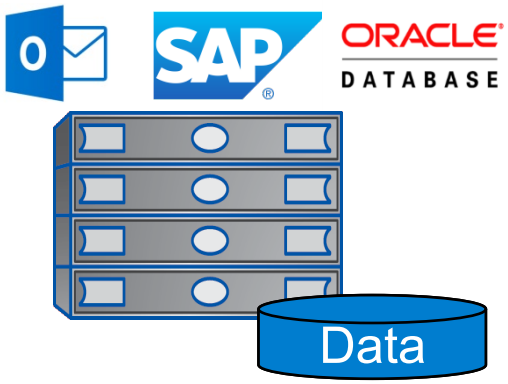
No easy way to analyze data stored in enterprise storage (NFS)



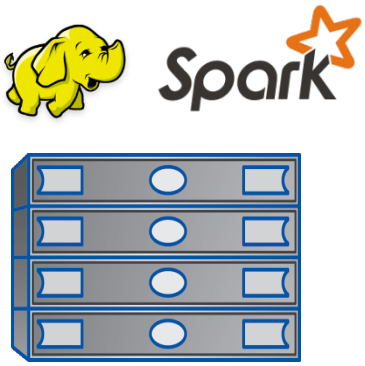
Production System

Motivation

No easy way to analyze data stored in enterprise storage (NFS)



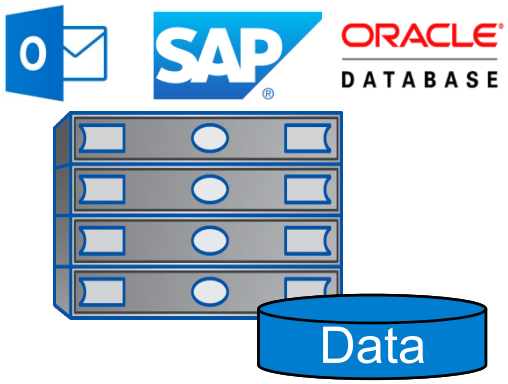
Production System



Analytics System

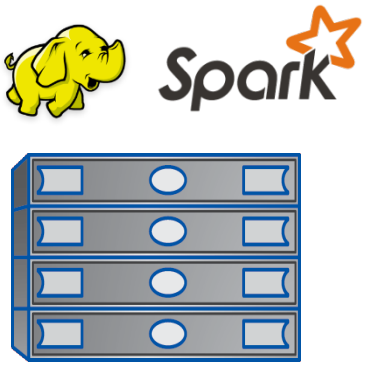
Motivation

No easy way to analyze data stored in enterprise storage (NFS)



Production System

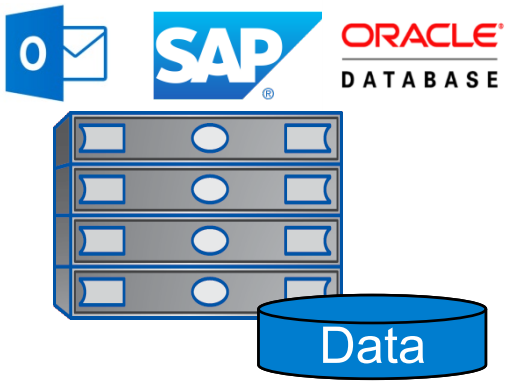
- Bank



Analytics System

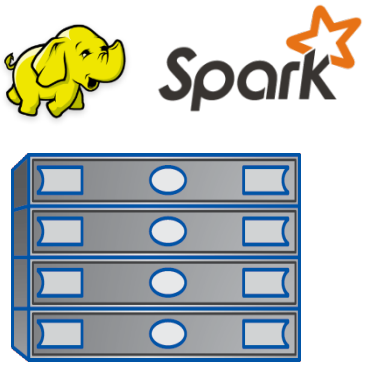
Motivation

No easy way to analyze data stored in enterprise storage (NFS)



Production System

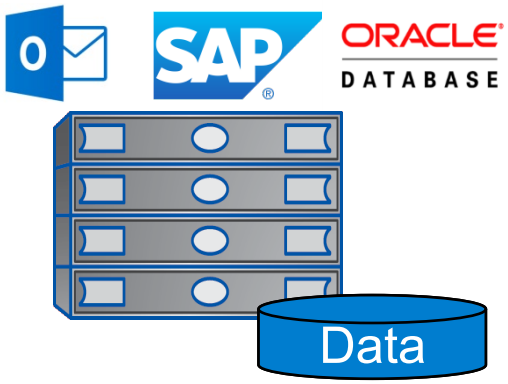
- Bank
- AutoSupport



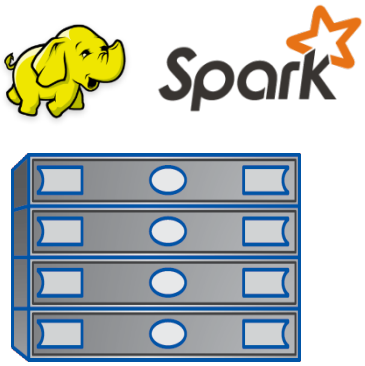
Analytics System

Motivation

No easy way to analyze data stored in enterprise storage (NFS)



Production System

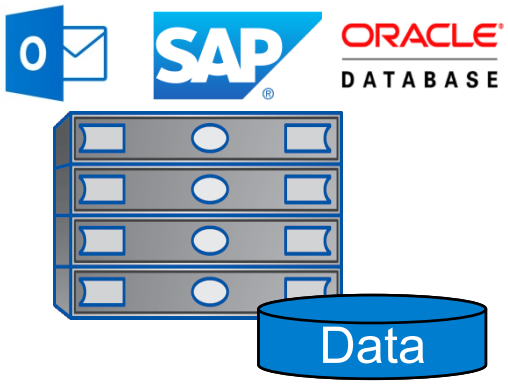


Analytics System

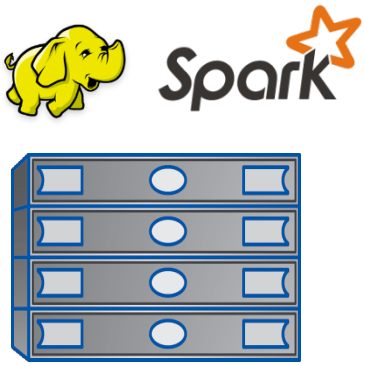
Motivation

No easy way to analyze data stored in enterprise storage (NFS)

- Separate infrastructures for production systems and analytics systems



Production System

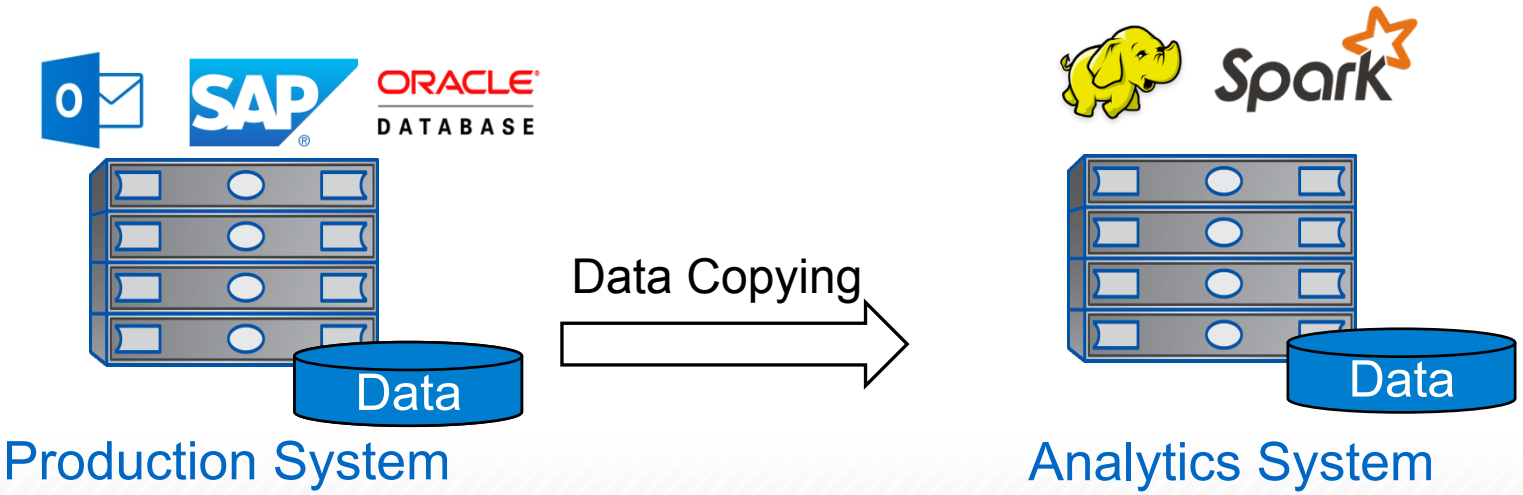


Analytics System

Motivation

No easy way to analyze data stored in enterprise storage (NFS)

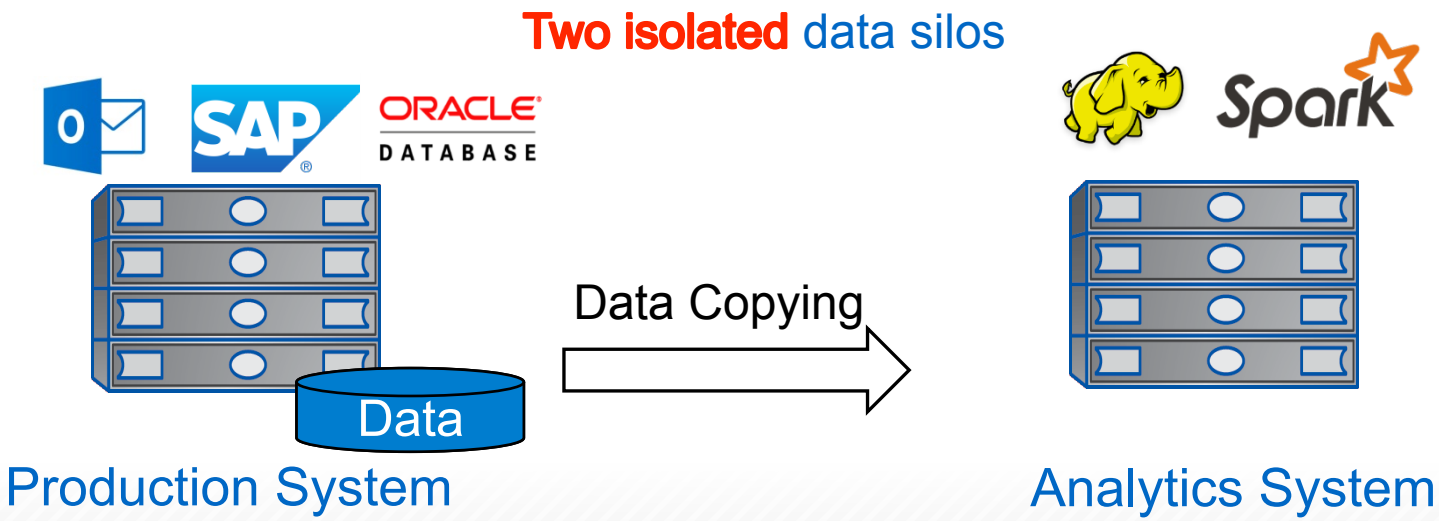
- Separate infrastructures for production systems and analytics systems



Motivation

No easy way to analyze data stored in enterprise storage (NFS)

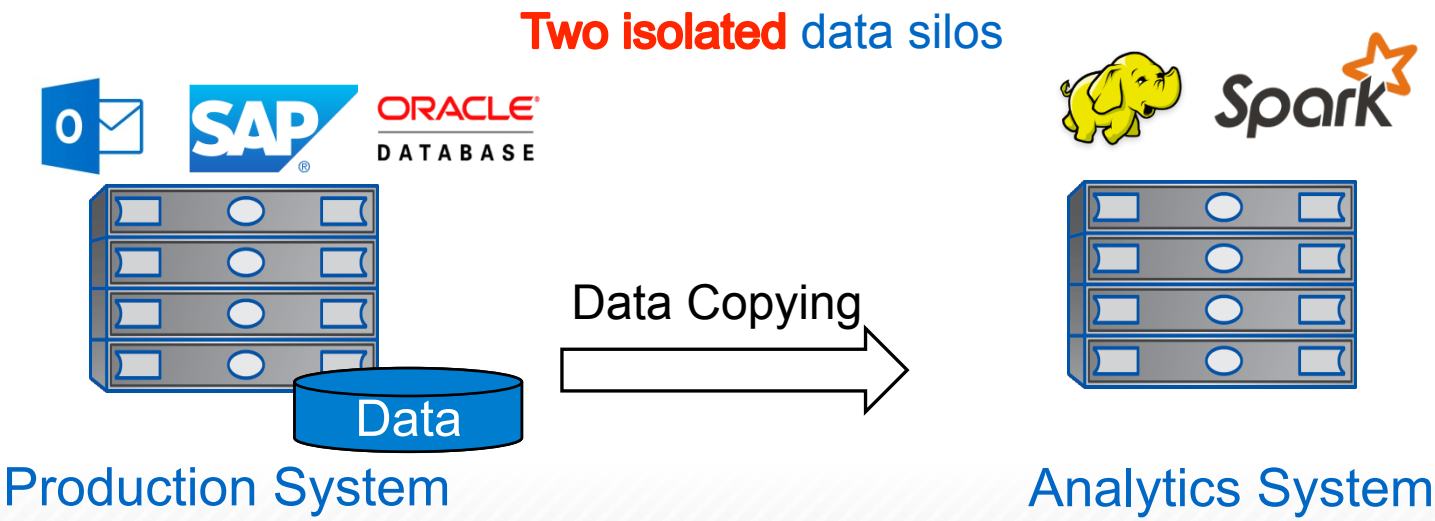
- Separate infrastructures for production systems and analytics systems



Motivation

No easy way to analyze data stored in enterprise storage (NFS)

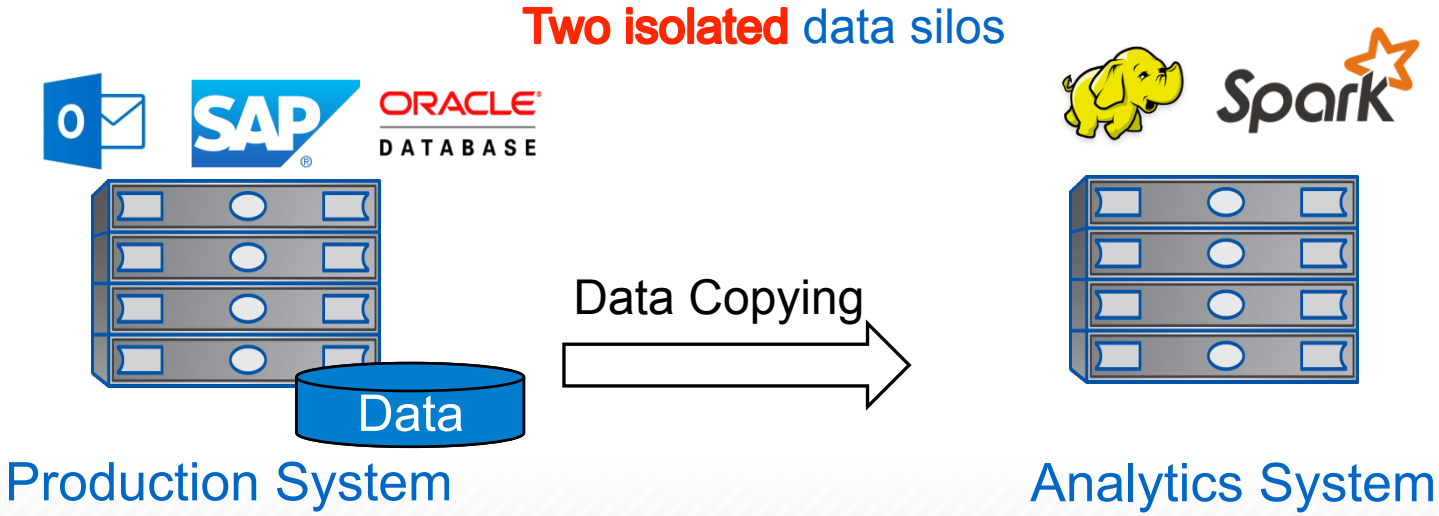
- Separate infrastructures for production systems and analytics systems
- Problems
  - Copying PBs of data is time consuming



Motivation

No easy way to analyze data stored in enterprise storage (NFS)

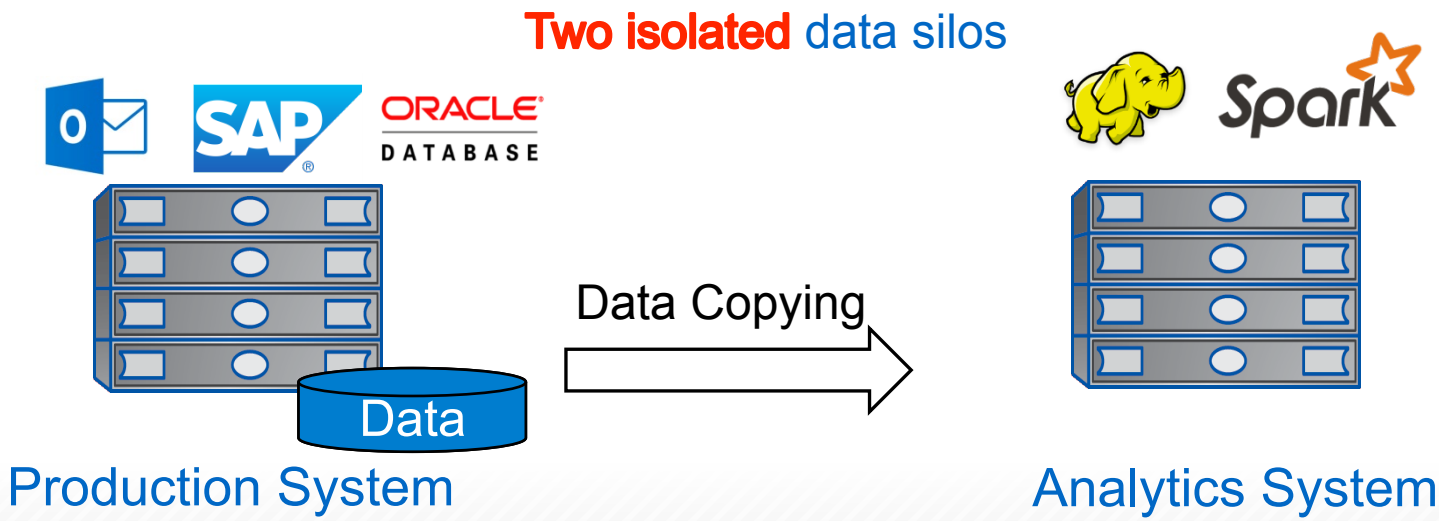
- Separate infrastructures for production systems and analytics systems
- Problems
  - Copying PBs of data is time consuming
  - 3 × storage overhead in HDFS



Motivation

No easy way to analyze data stored in enterprise storage (NFS)

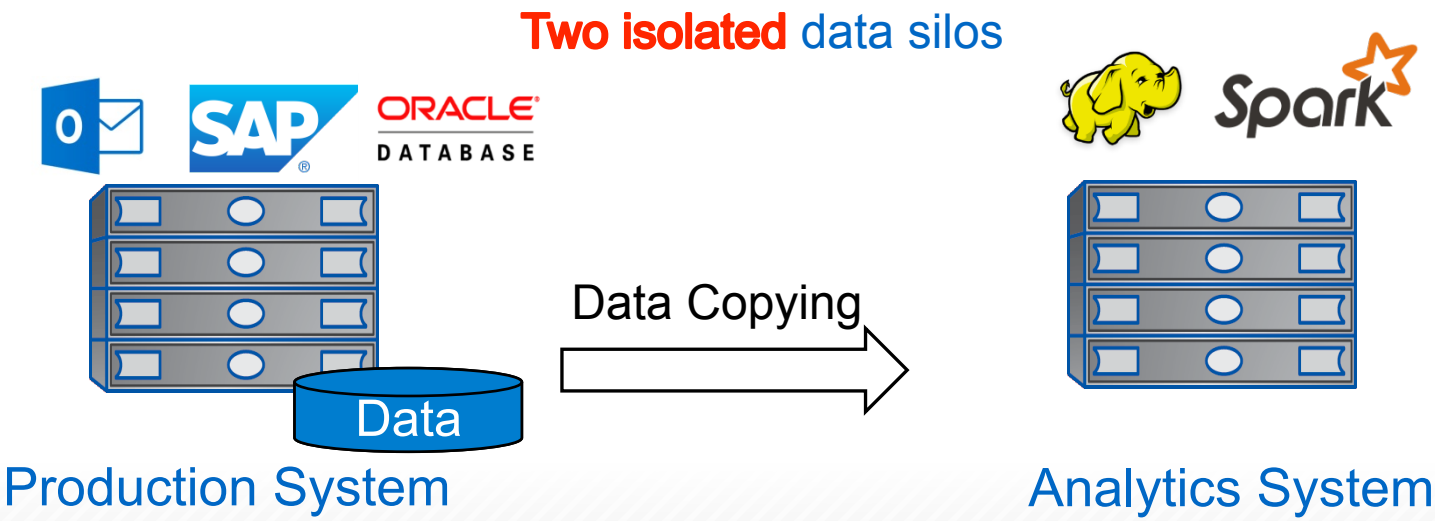
- Separate infrastructures for production systems and analytics systems
- Problems
  - Copying PBs of data is time consuming
  - 3 × storage overhead in HDFS
  - Periodical re-synchronization later on



Motivation

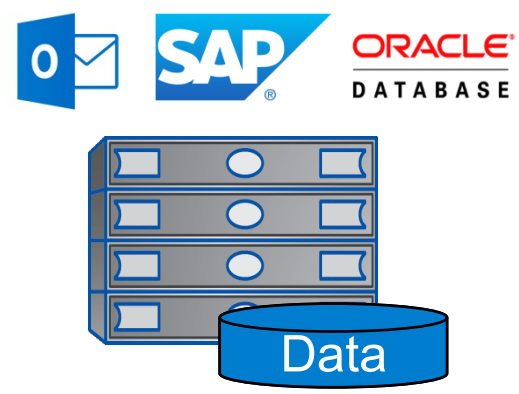
No easy way to analyze data stored in enterprise storage (NFS)

- Separate infrastructures for production systems and analytics systems
- Problems
  - Copying PBs of data is time consuming
  - 3 × storage overhead in HDFS
  - Periodical re-synchronization later on
  - Legal prevents data copying

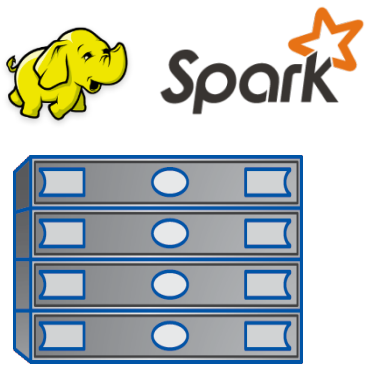


Mambo

An NFS connector, enabling direct analytics for data on Enterprise storage (NFS)



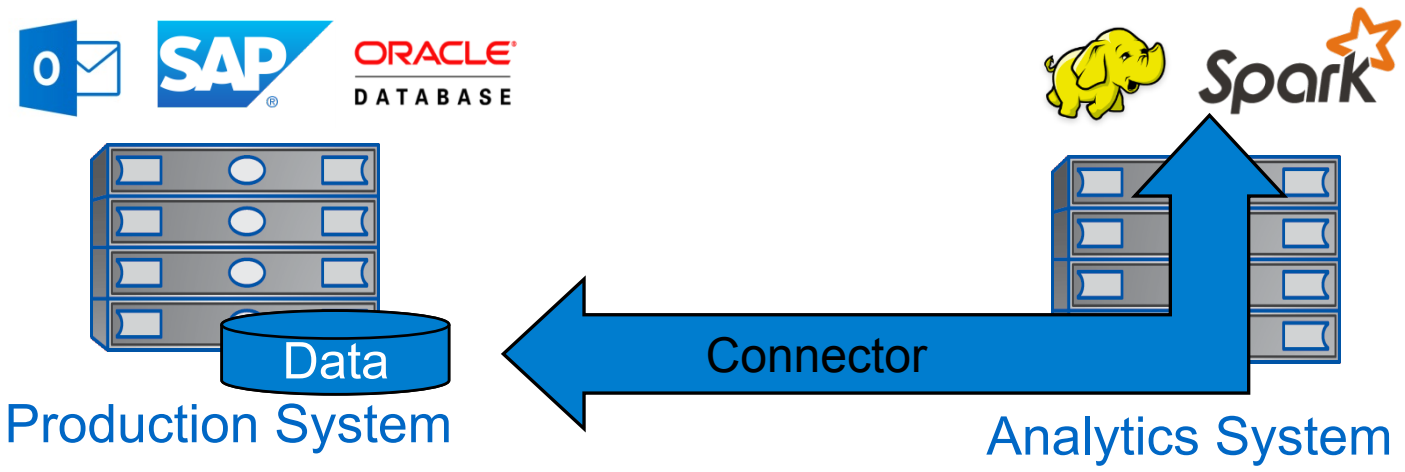
Production System



Analytics System

Mambo

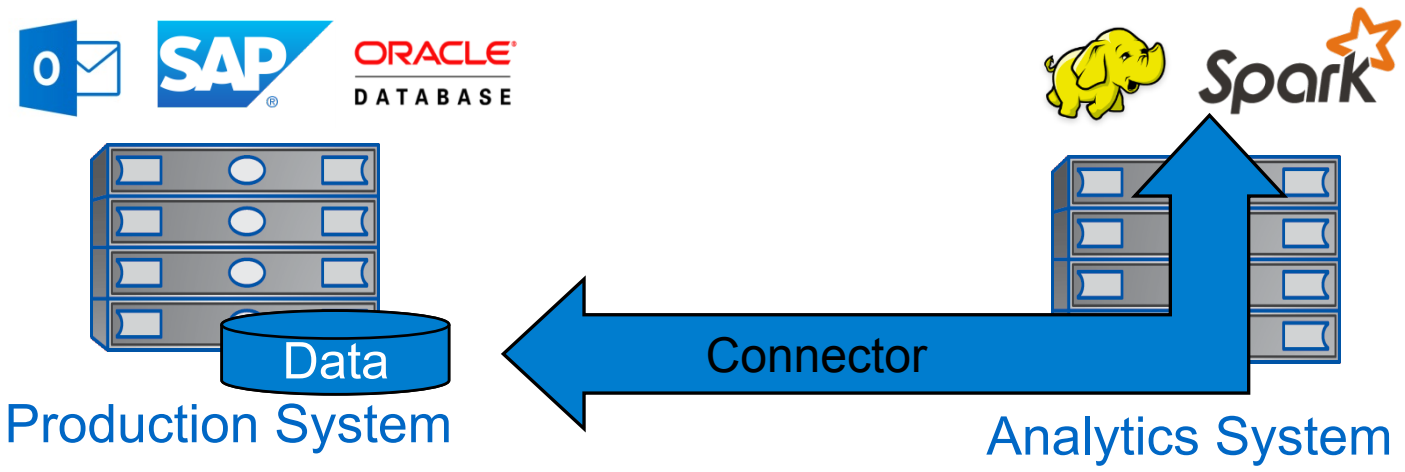
An NFS connector, enabling direct analytics for data on Enterprise storage (NFS)



Mambo

An NFS connector, enabling direct analytics for data on Enterprise storage (NFS)

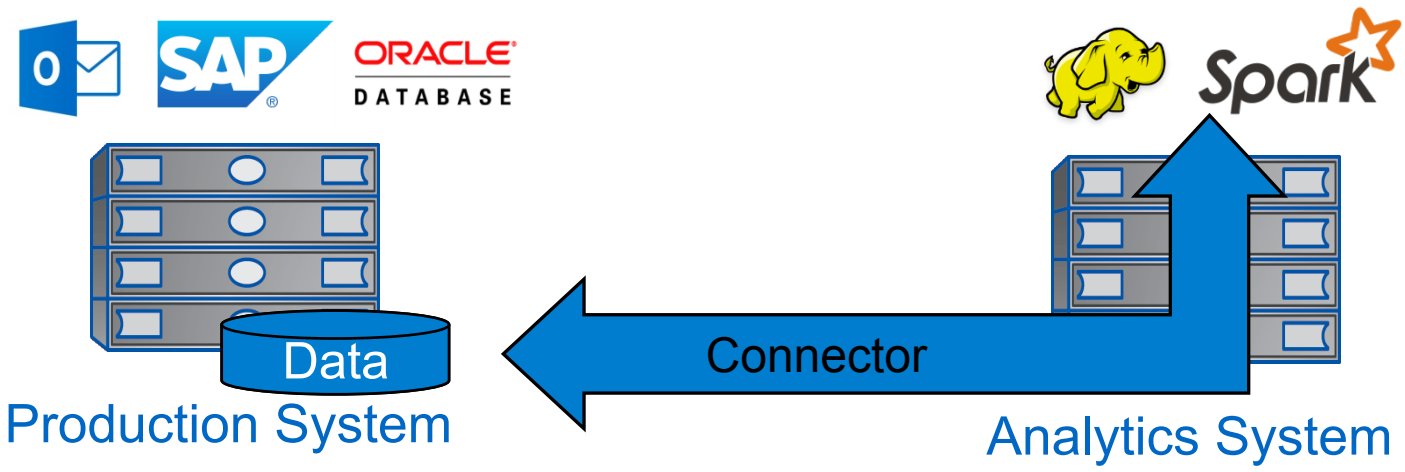
- Remove data copying
- Remove storage overhead (single copy)
- Remove data re-synchronization
- No legal issue



Mambo

An NFS connector, enabling direct analytics for data on Enterprise storage (NFS)

- Remove data copying
- Remove storage overhead (single copy)
- Remove data re-synchronization
- No legal issue



Copying is not required; you can do analytics in-place

Journey From Research to Product

Project History

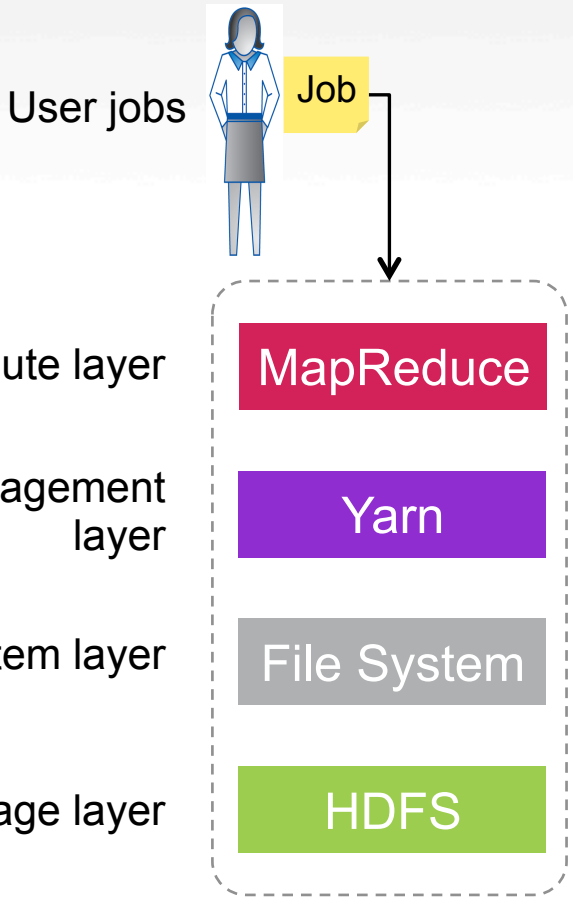
From Research to Product

2011	• Talked with customers • Developed initial prototype
2012	• Madalin Mihailescu refined prototype • Added a distributed cache • Obtained traces from UC Berkeley • Published in FAST'13
2013 ~ now	• Xing Lin refactored code for Hadoop 2.0 • Optimized for 10 Gb networks • Obtained legal approval for open-source • Posted to GitHub • Customer Proof-of-Concepts (PoCs) • Pushing to merge into Hadoop

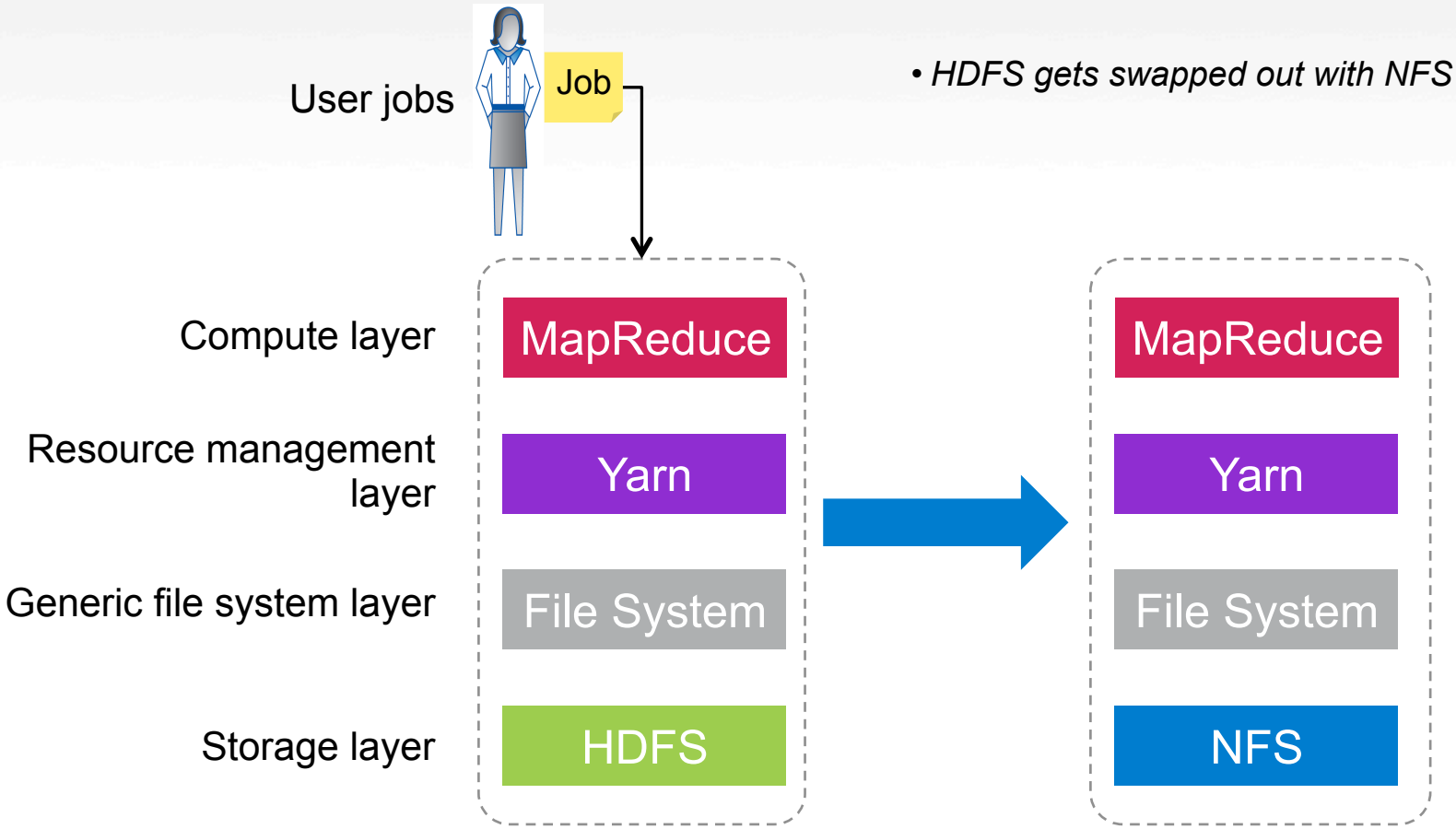
Use Cases

How many ways can you use Mambo?

Analyze Enterprise Data In-place

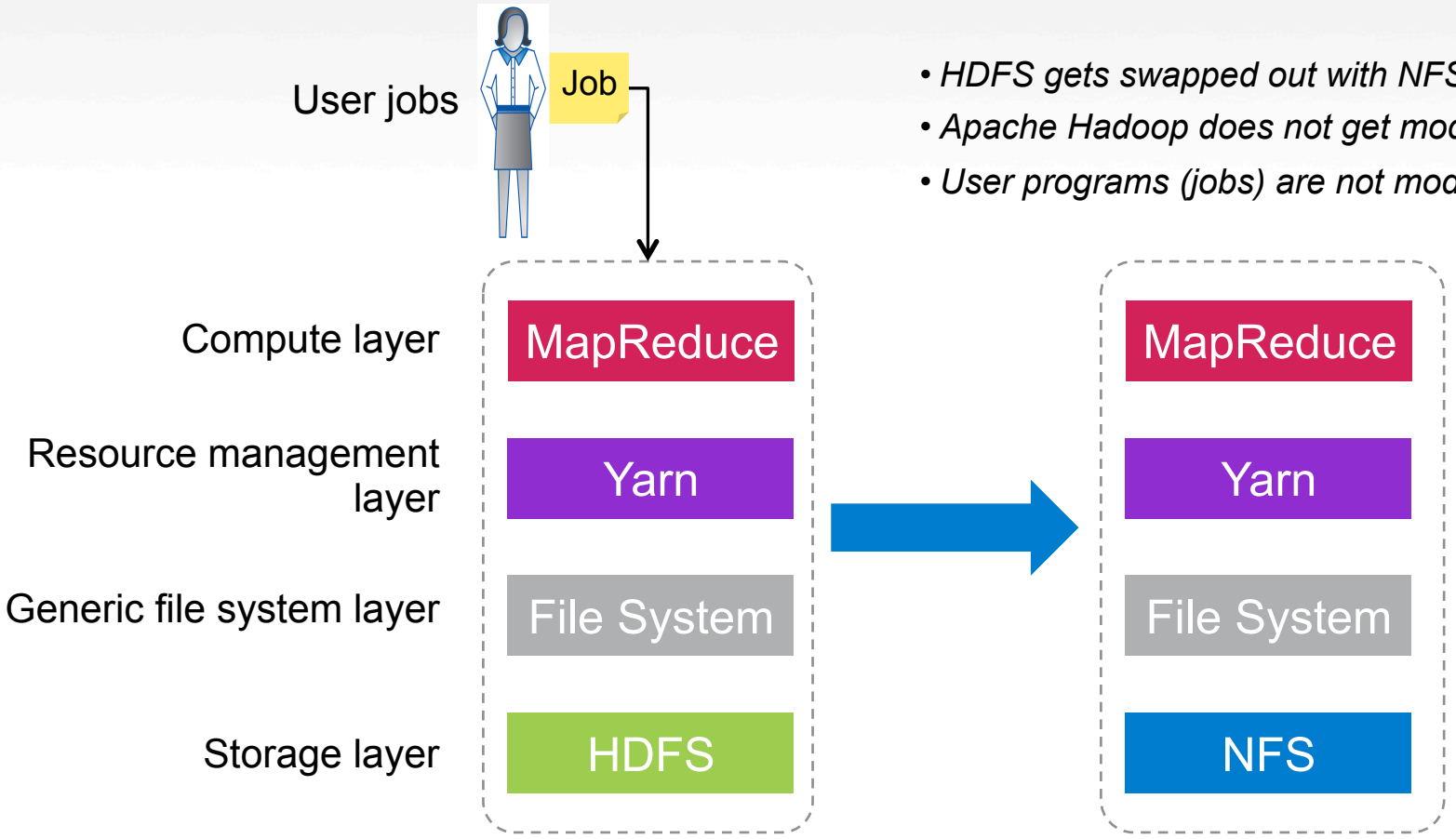


Analyze Enterprise Data In-place



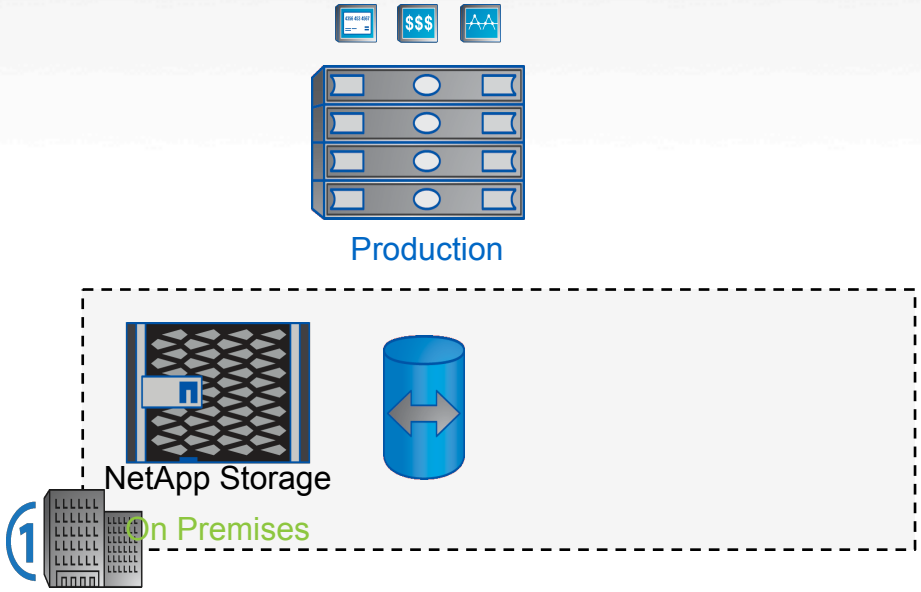
- *HDFS gets swapped out with NFS**

Analyze Enterprise Data In-place



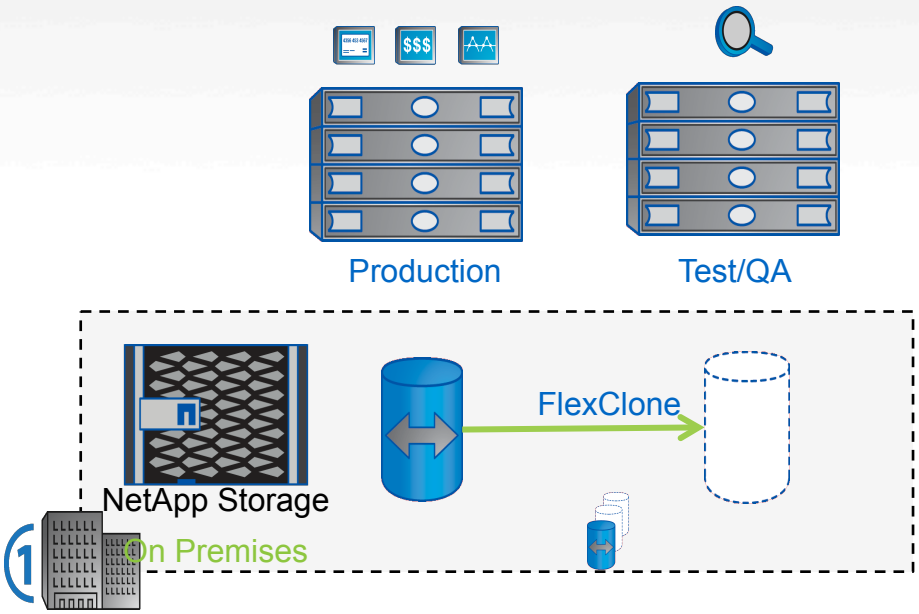
- *HDFS gets swapped out with NFS**
- *Apache Hadoop does not get modified.*
- *User programs (jobs) are not modified.*

Easily Launch Test Environments



- Use NetApp FlexClones for creating test environments quickly
  - Use a copy of production data for realistic Test/QA environments (e.g., AutoSupport)

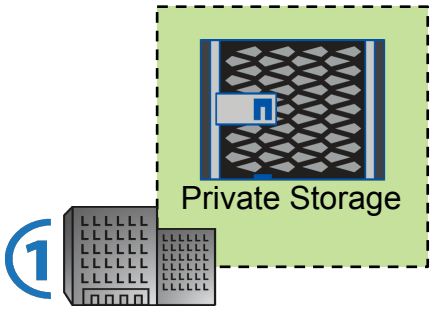
Easily Launch Test Environments



- Use NetApp FlexClones for creating test environments quickly
  - Use a copy of production data for realistic Test/QA environments (e.g., AutoSupport)

Use cloud to Analyze Data

Secondary private storage at a colocation facility (e.g., Equinix), for backup and fast restoration with cloud



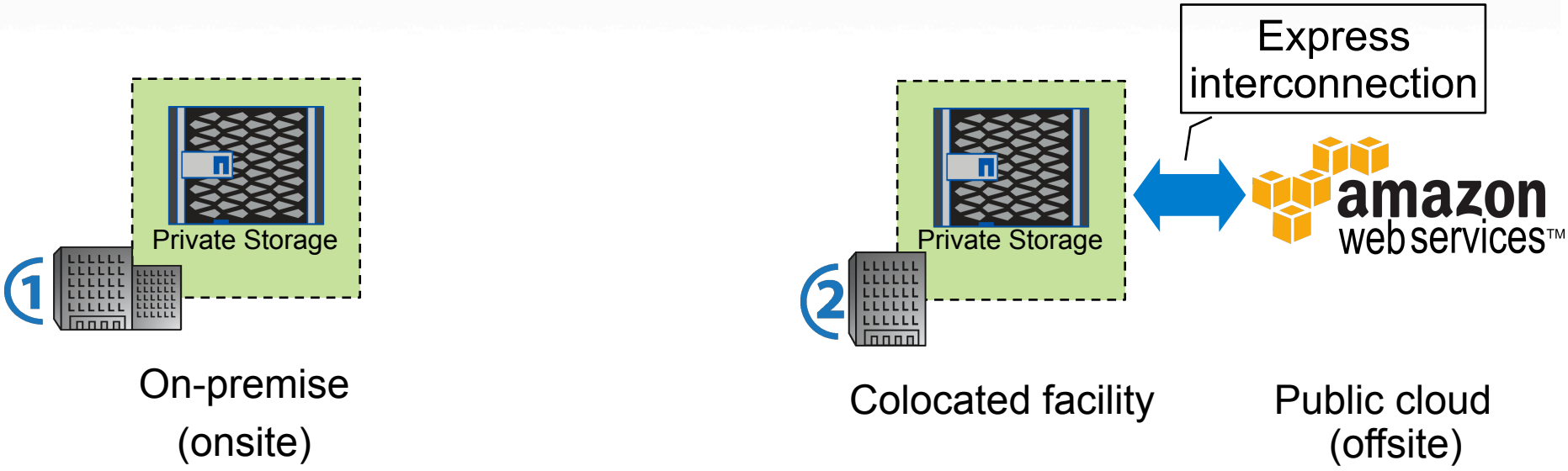
On-premise
(onsite)

amazon web services™

Public cloud
(offsite)

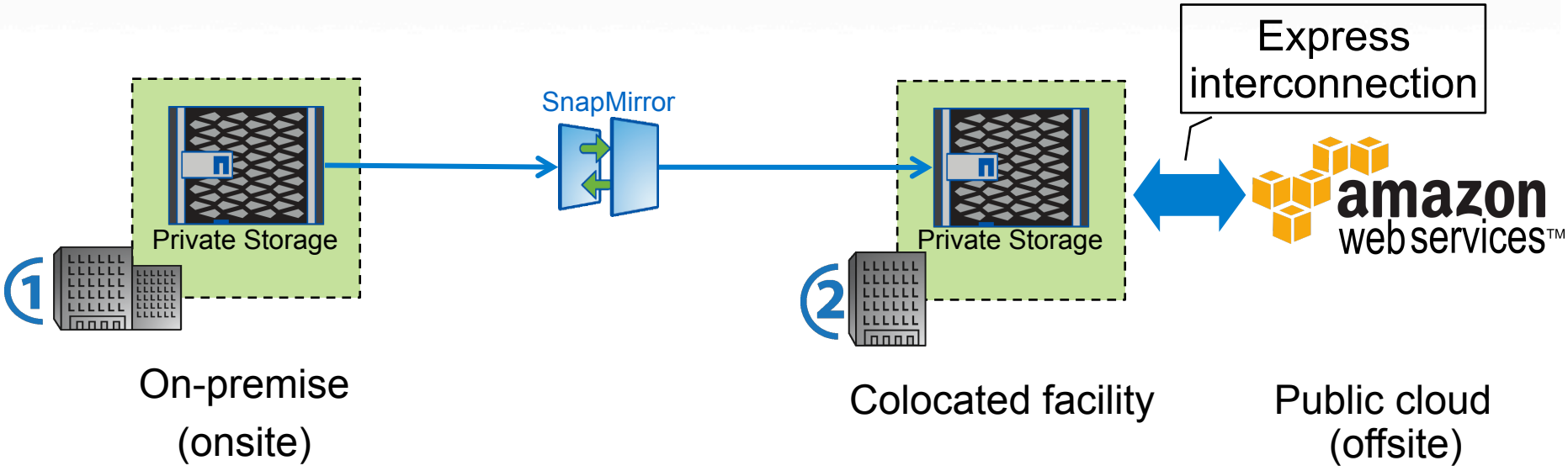
Use cloud to Analyze Data

Secondary private storage at a colocation facility (e.g., Equinix), for backup and fast restoration with cloud



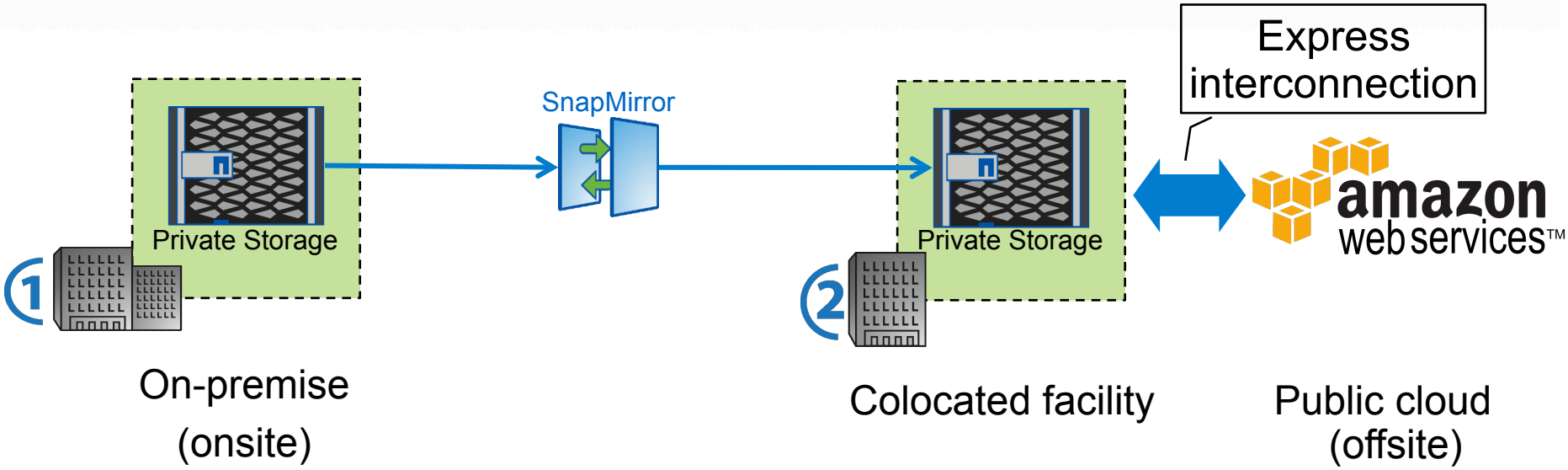
Use cloud to Analyze Data

Secondary private storage at a colocation facility (e.g., Equinix), for backup and fast restoration with cloud



Use cloud to Analyze Data

Secondary private storage at a colocation facility (e.g., Equinix), for backup and fast restoration with cloud

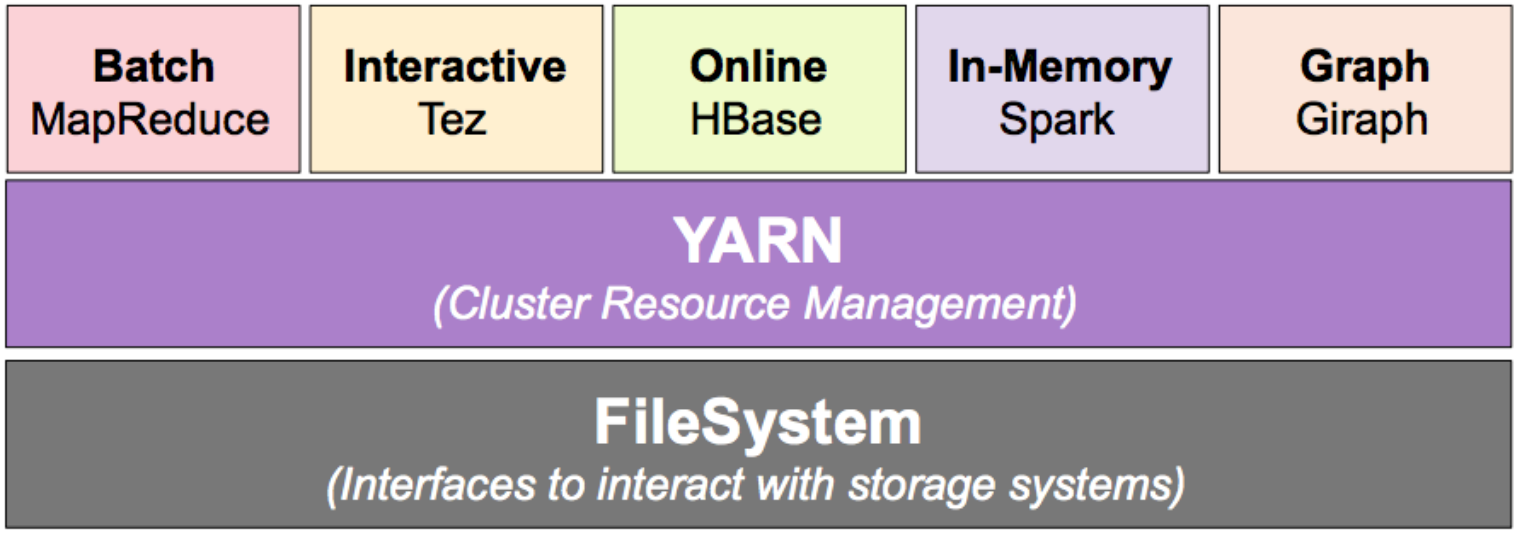


Launch Hadoop in the cloud and use data on private storage

Design and Implementation

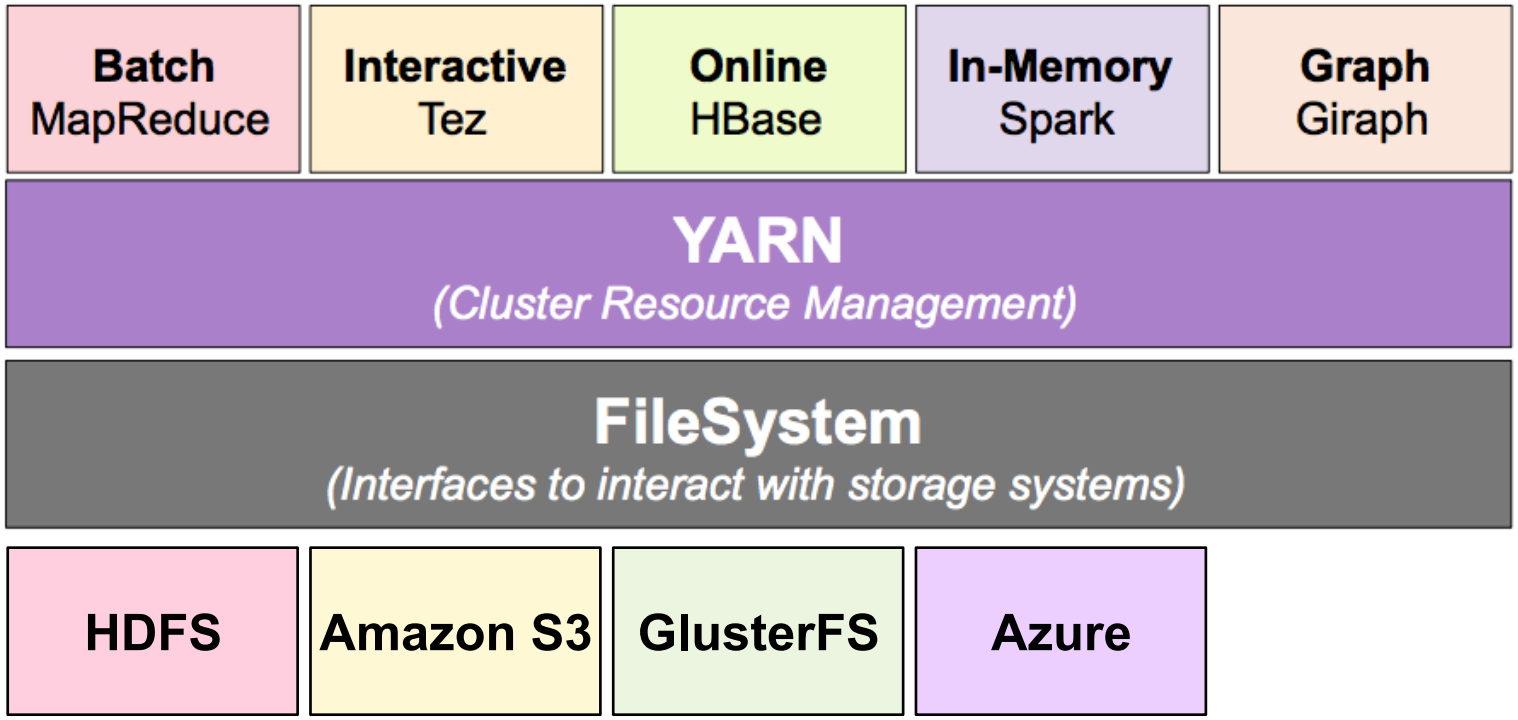
Architecture Overview

Mambo: an NFS client in Java, implementing the Hadoop generic file system API



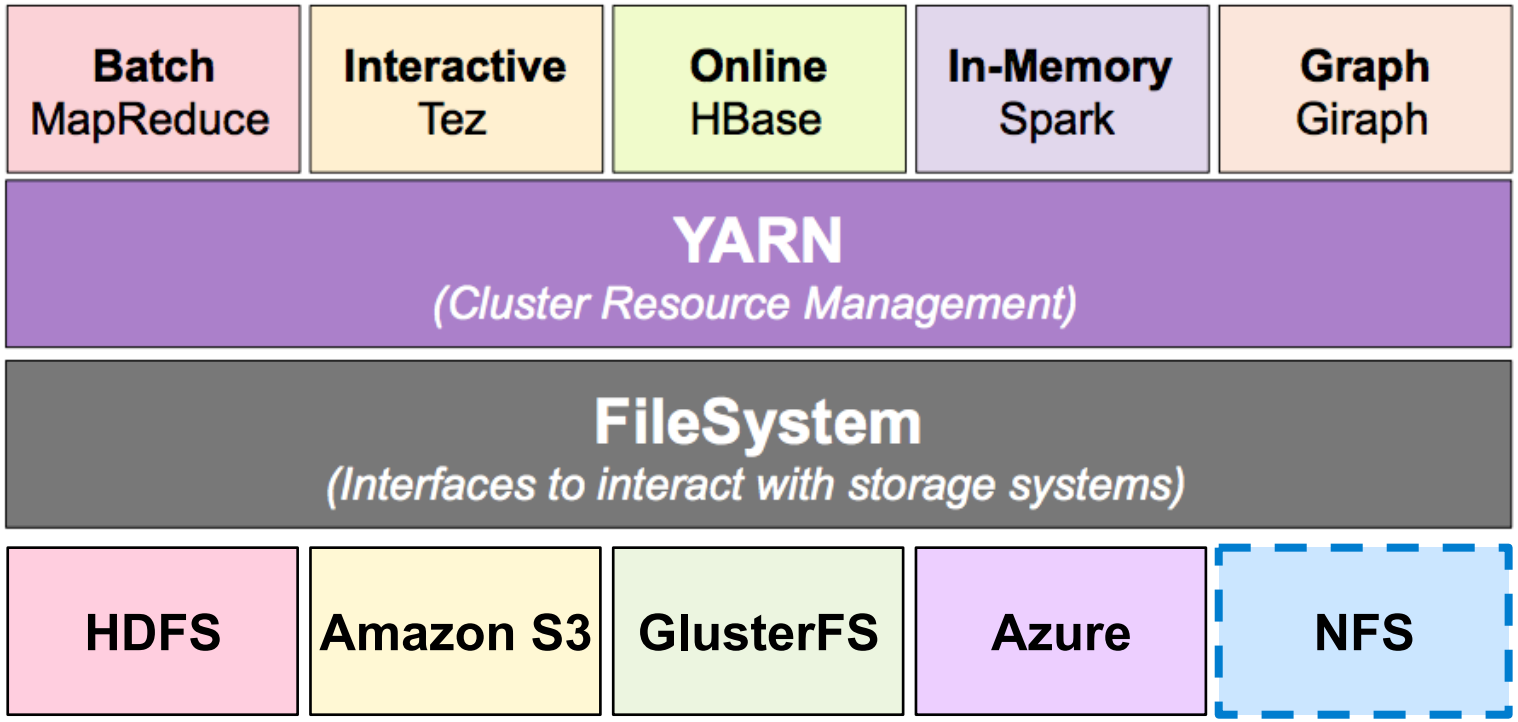
Architecture Overview

Mambo: an NFS client in Java, implementing the Hadoop generic file system API



Architecture Overview

Mambo: an NFS client in Java, implementing the Hadoop generic file system API

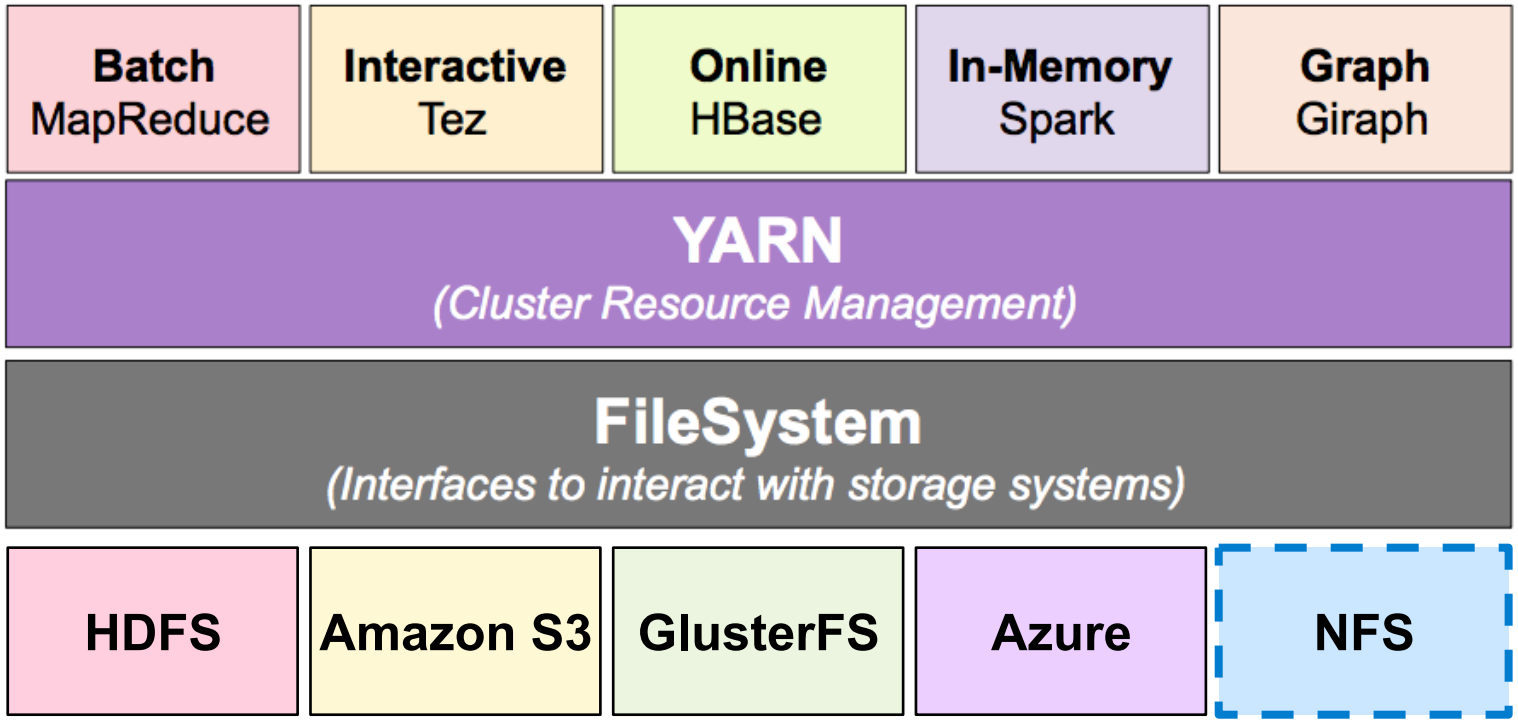


Filled the missing piece

Architecture Overview

Mambo: an NFS client in Java, implementing the Hadoop generic file system API

Copying is not required



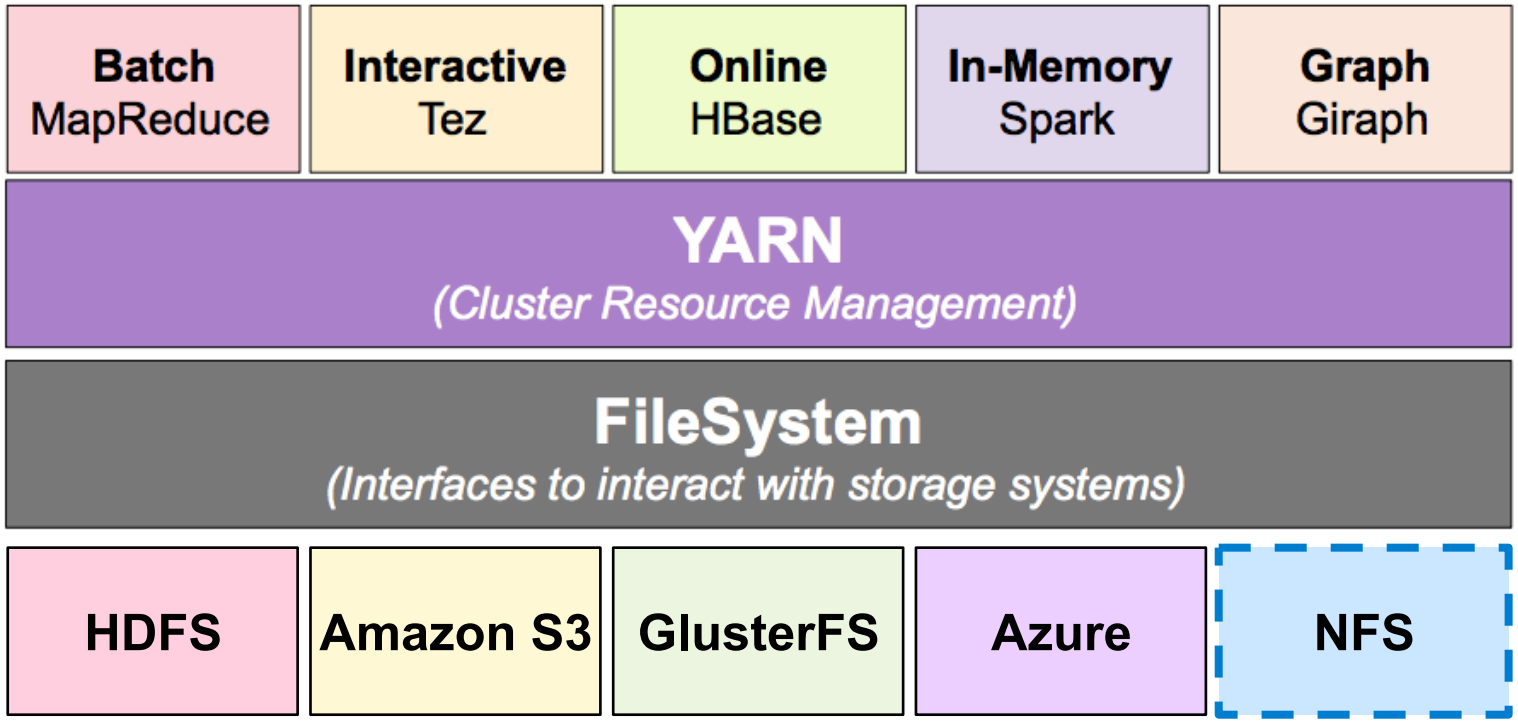
Filled the missing piece

Architecture Overview

Mambo: an NFS client in Java, implementing the Hadoop generic file system API

- No changes to Hadoop framework
- No changes to user programs

Copying is not required

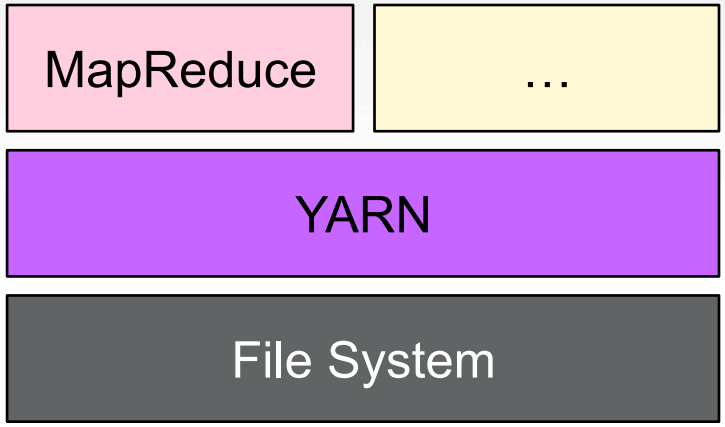


Filled the missing piece

Tight integration with Hadoop/MapReduce

- Optimized for large sequential I/O (e.g., 1MB IO)
- Commit data to disk only when a task succeeds
- Intelligent prefetching for streaming reads; aware of task sizes

Implementation

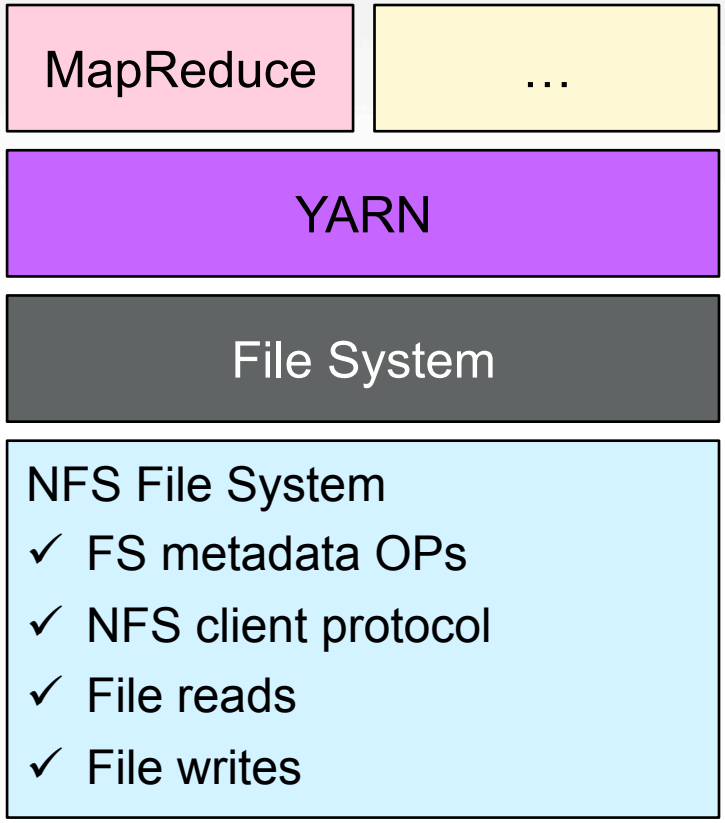


Computation frameworks

Resource management layer

Hadoop generic filesystem API

Implementation

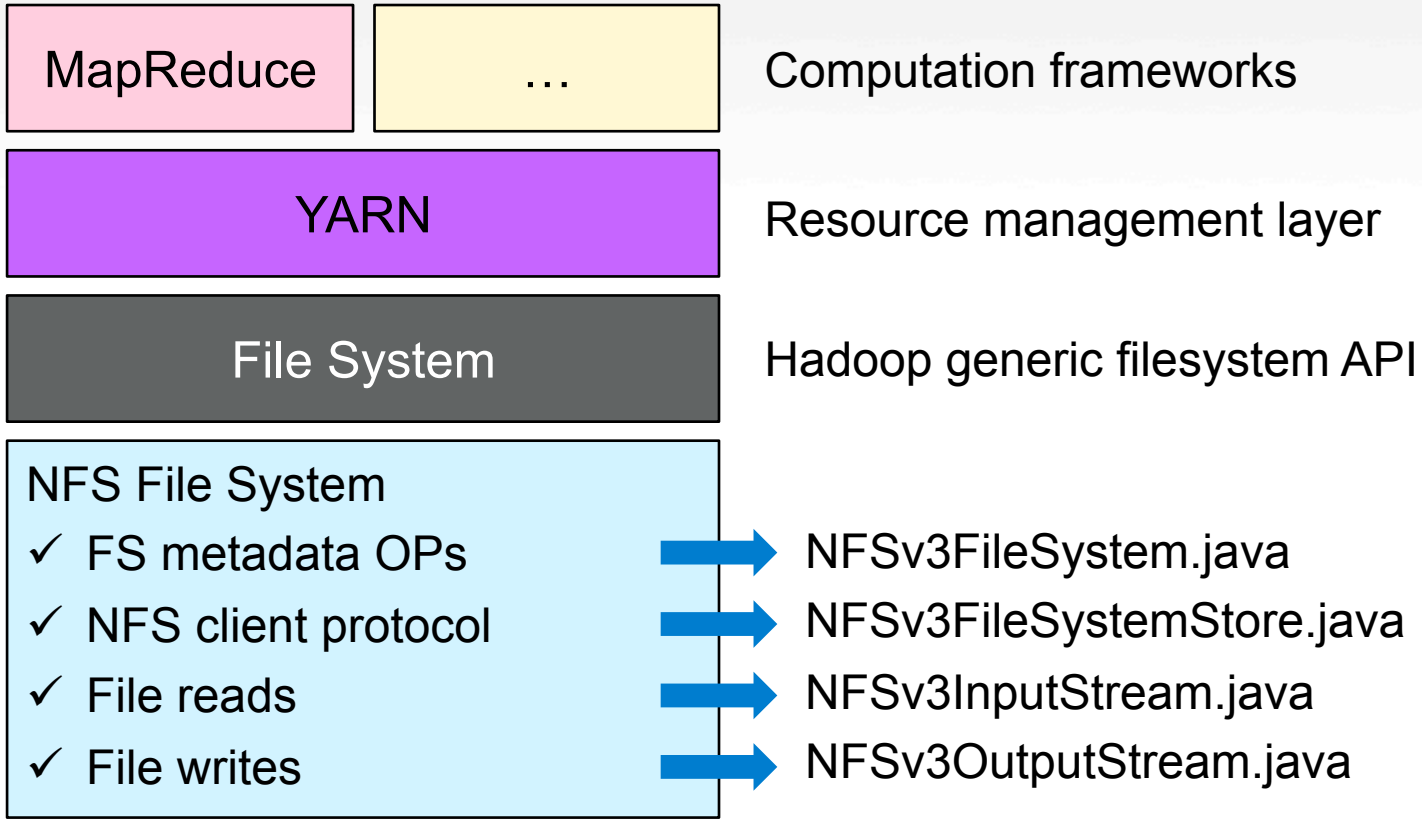


Computation frameworks

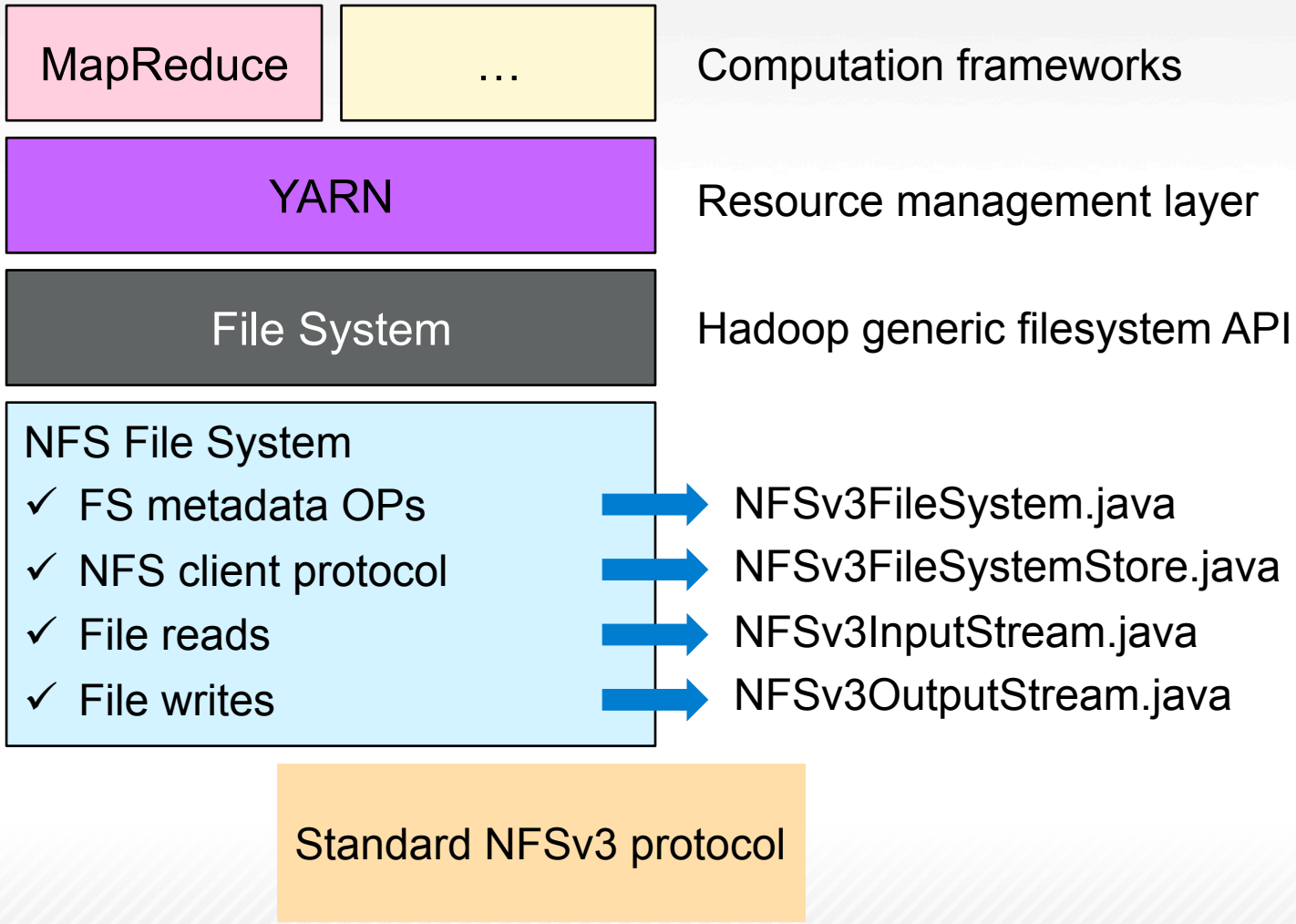
Resource management layer

Hadoop generic filesystem API

Implementation



Implementation



How to Use it?

- Source code (jar library file)
  - Get code from GitHub
  - Compile the code
  - Install the jar file
    - Copy the jar file to the library directory for Hadoop installation
  - **Only need to modify two configuration files**
    - core-site.xml (hadoop core configuration file)
    - nfs-mapping.json (nfs configuration file)

How to Use it?

- Source code (jar library file)
  - Get code from GitHub
  - Compile the code
  - Install the jar file
    - Copy the jar file to the library directory for Hadoop installation
  - **Only need to modify two configuration files**
    - core-site.xml (hadoop core configuration file)
    - nfs-mapping.json (nfs configuration file)
- Or just try the Amazon Cloud Formation template with everything configured

Configure core-site.xml

HDFS

Property	Value
fs.defaultFS	hdfs://namenode:54310/

NFS

Property	Value
fs.defaultFS	nfs://nfsserver:2049/
fs.nfs.configuration	<path-to-configuration-file>
fs.nfs.impl	org.apache.hadoop.fs.nfs. NFSv3FileSystem
fs.AbstractFileSystem.nfs.impl	org.apache.hadoop.fs.nfs. NFSv3AbstractFileSystem

Configure nfs-mapping.json

- Configurable properties
  - Export path
  - Read/write sizes
  - Split size (Hadoop task granularity)
  - Authentication method (supporting AUTH_NONE or AUTH_UNIX)
  - …
- Supports multiple controllers (for NetApp clustered ONTAP)
  - Aggregated bandwidth

Performance Evaluation

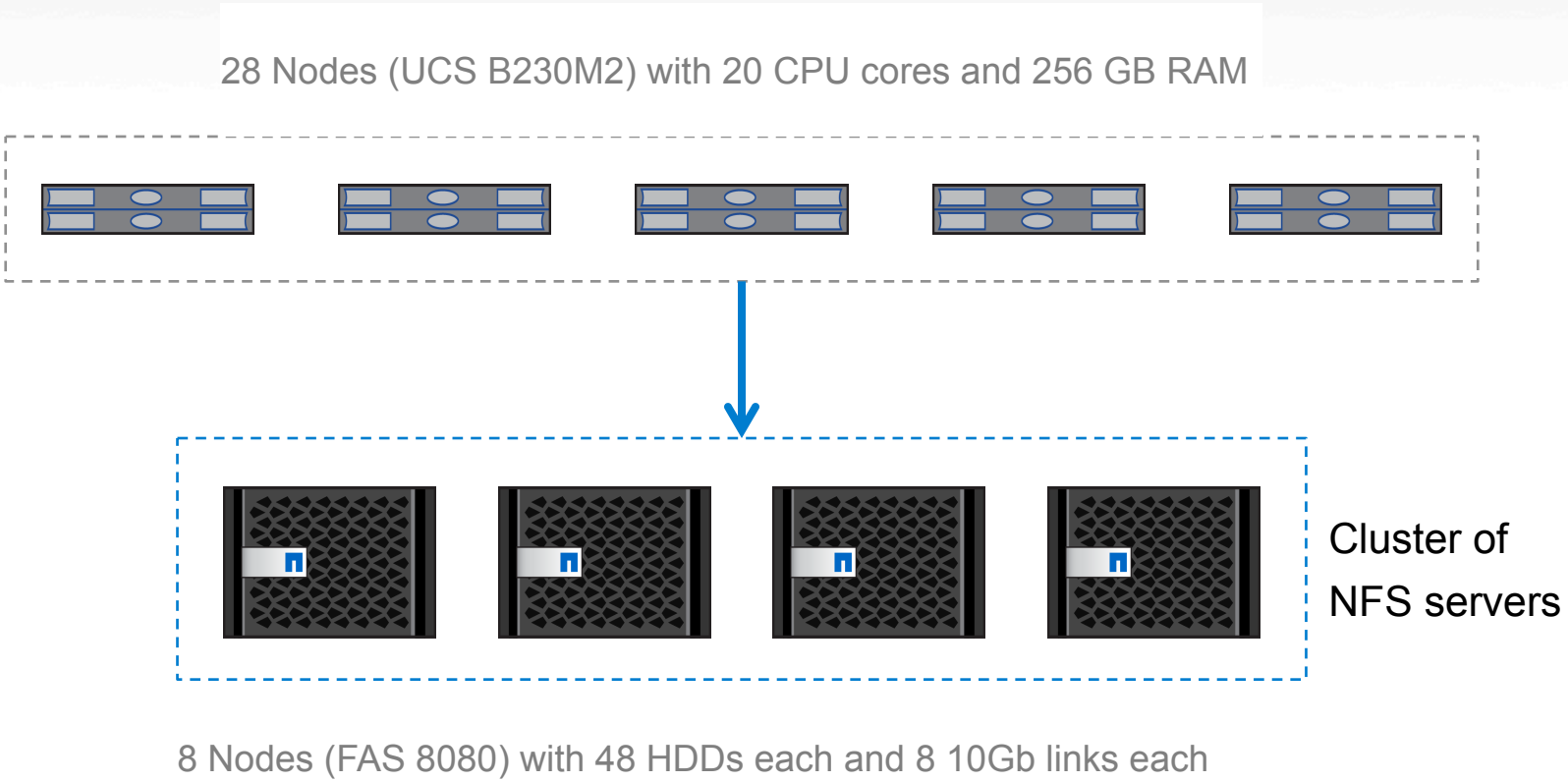
Highlights from MixApart

- MixApart: NFS connector + data prefetcher + local disk as cache
- Better performance with NFS connector than Hadoop with ingest (18%~26% reduction in job duration)
  - Overlaps data ingest with task computation
- Matches ideal Hadoop (data ingested into HDFS before-hand), with moderate/high data reuse across jobs

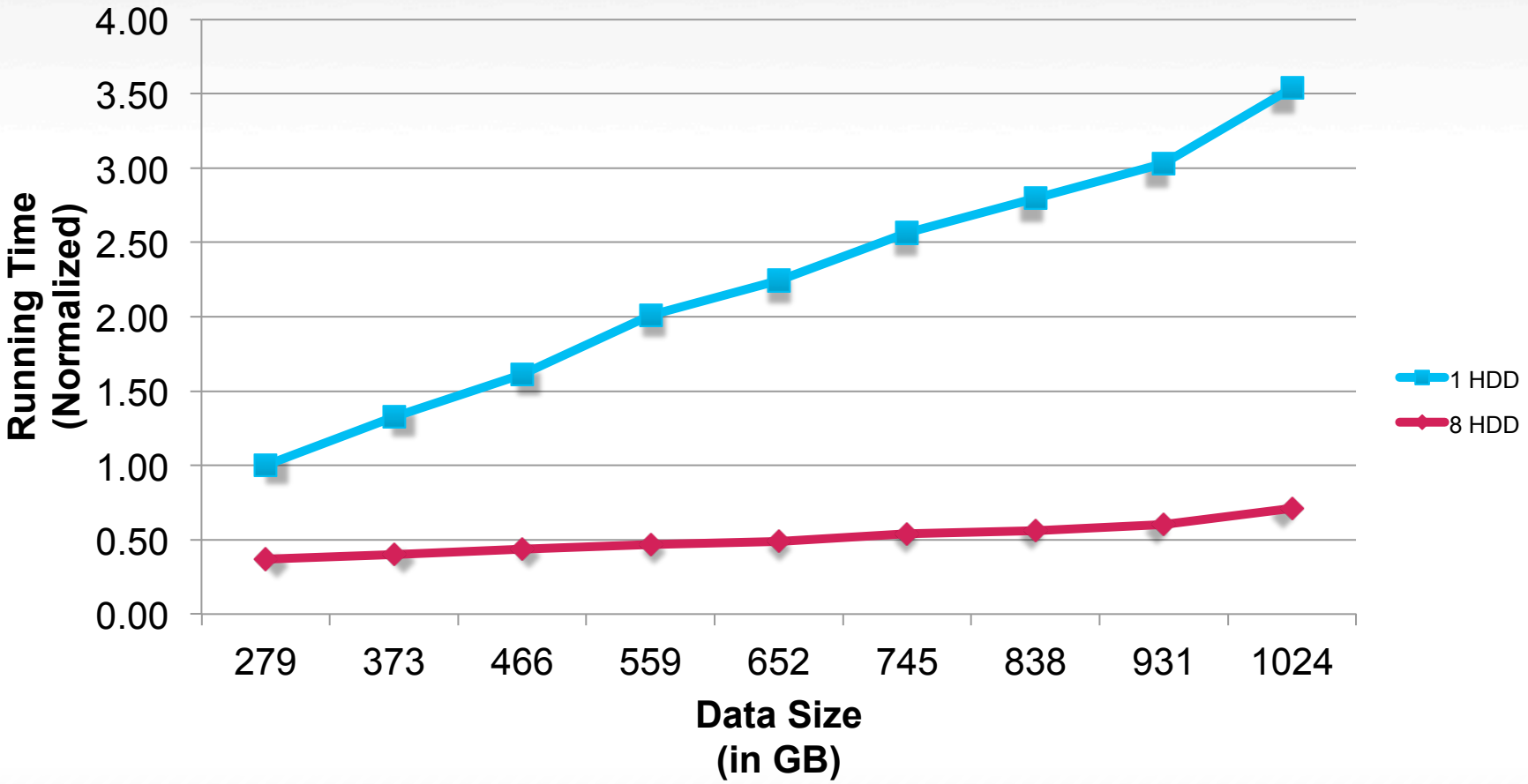
[1]MixApart: De-coupled Analytics for Shared Storage Servers.
Madalin Mihailescu, Gokul Soundararajan, and Cristiana Amza. In FAST '13

Scaling experiments

How does the NFS Connector scale with more storage and compute?

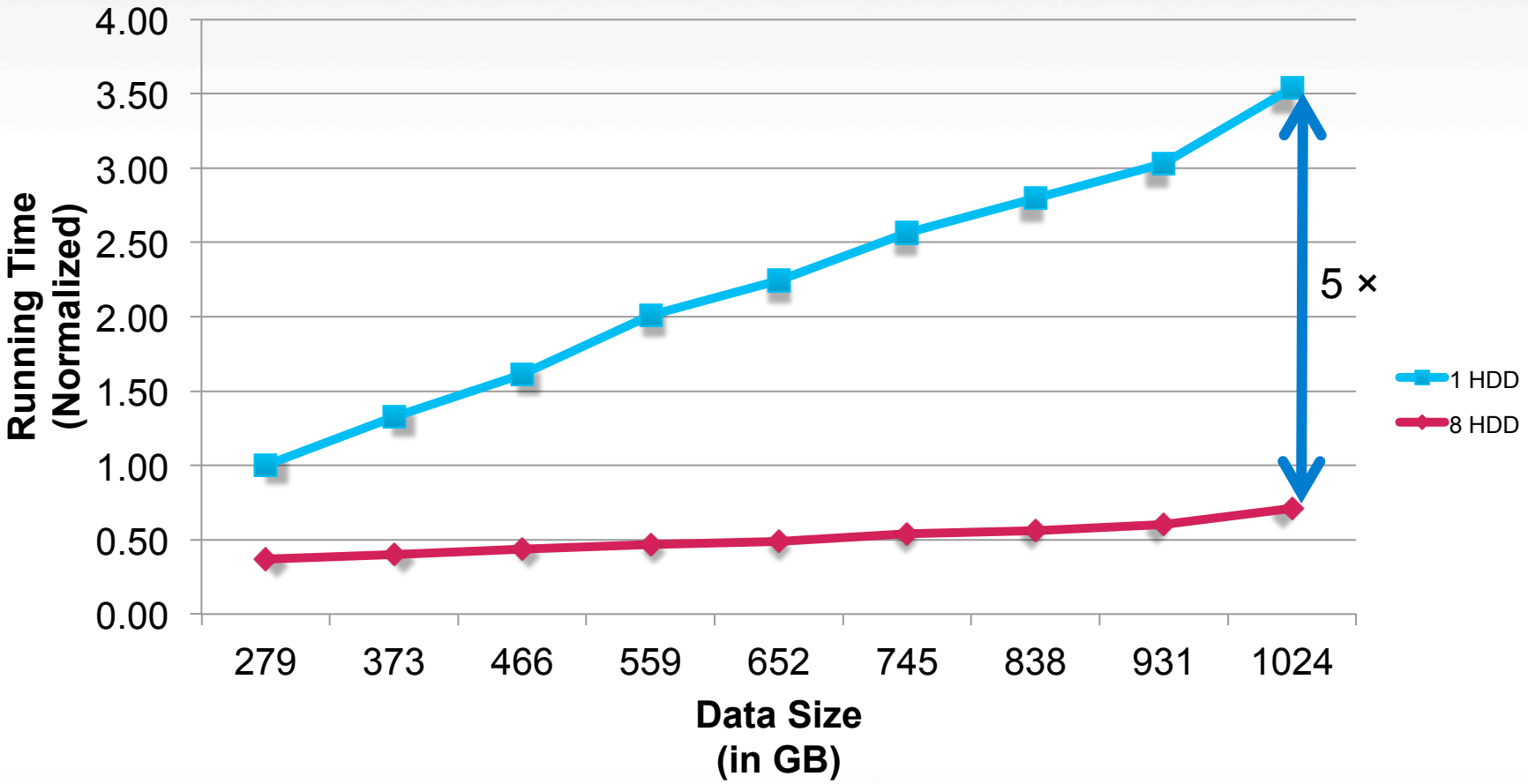


Scaling TeraGen



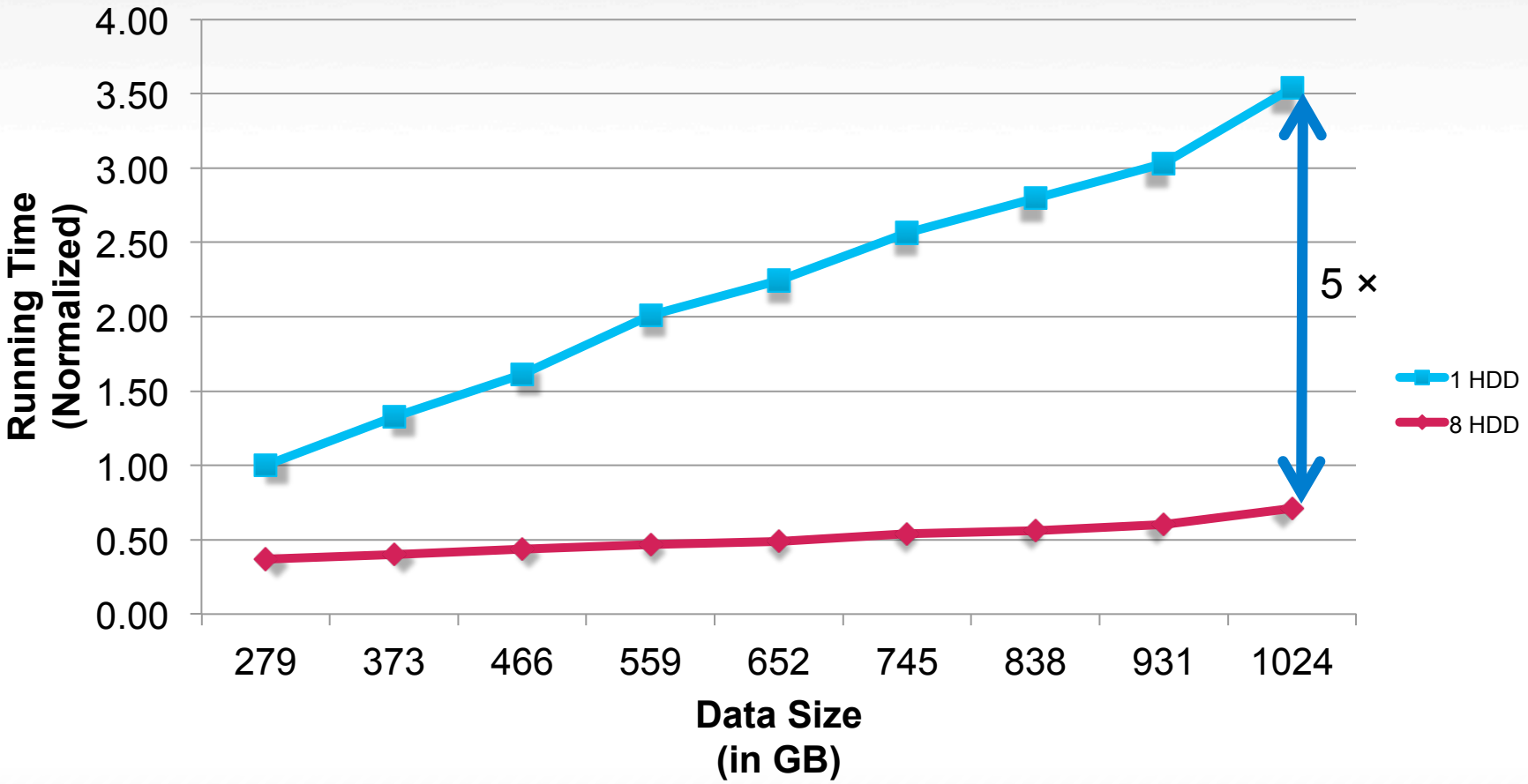
NetApp

Scaling TeraGen



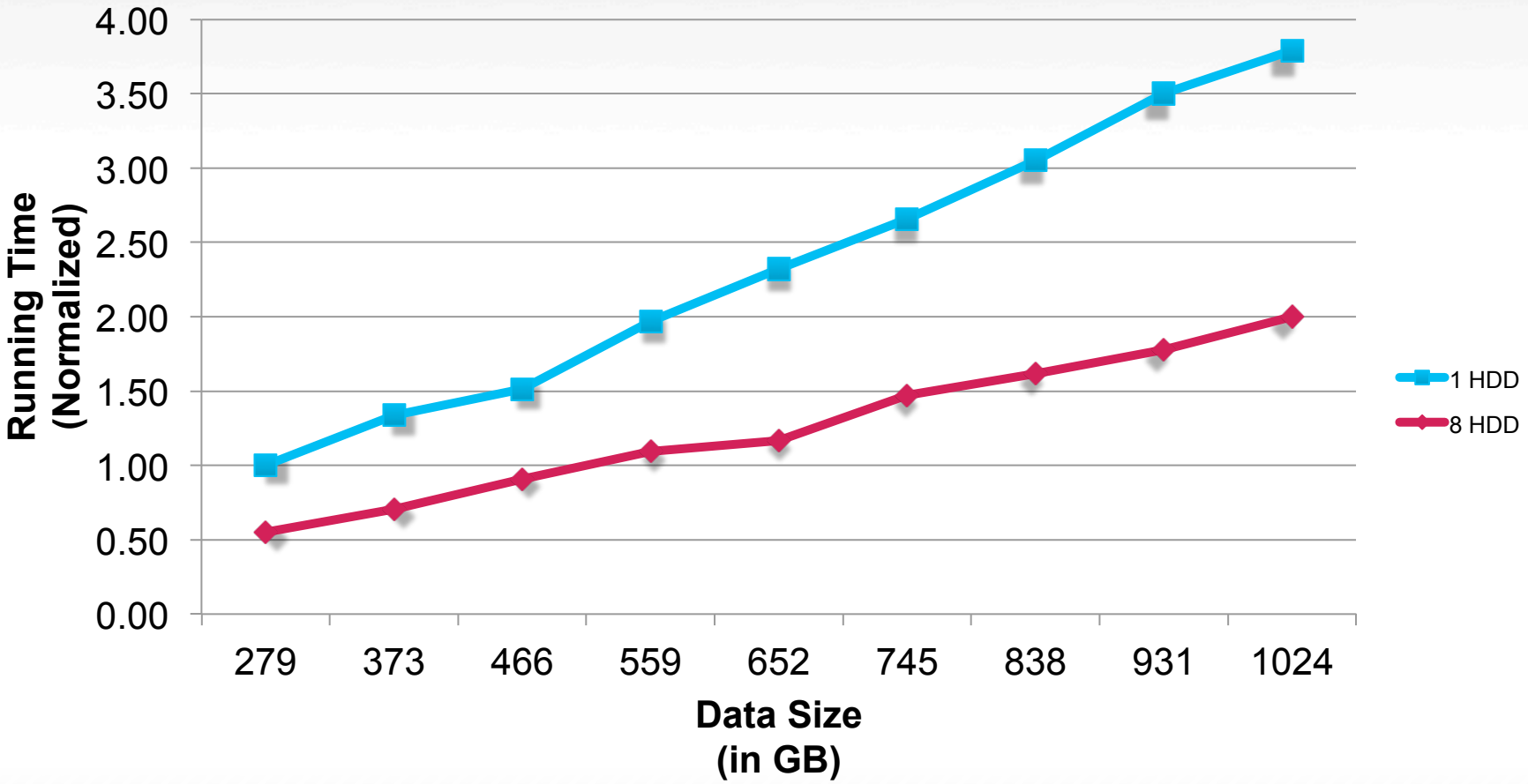
Scaling TeraGen

NFS connector scales well for large datasets.



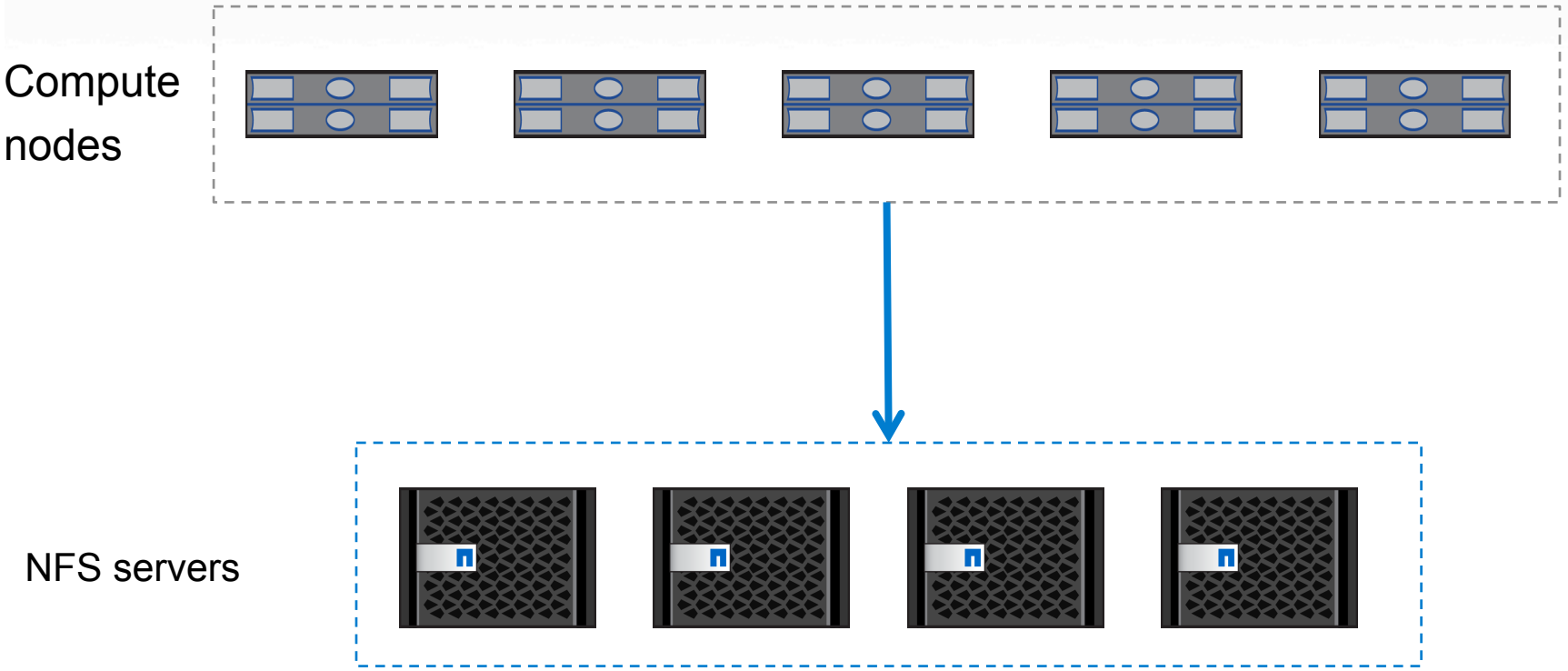
NetApp

Scaling TeraSort



Overcome NFS server bottleneck

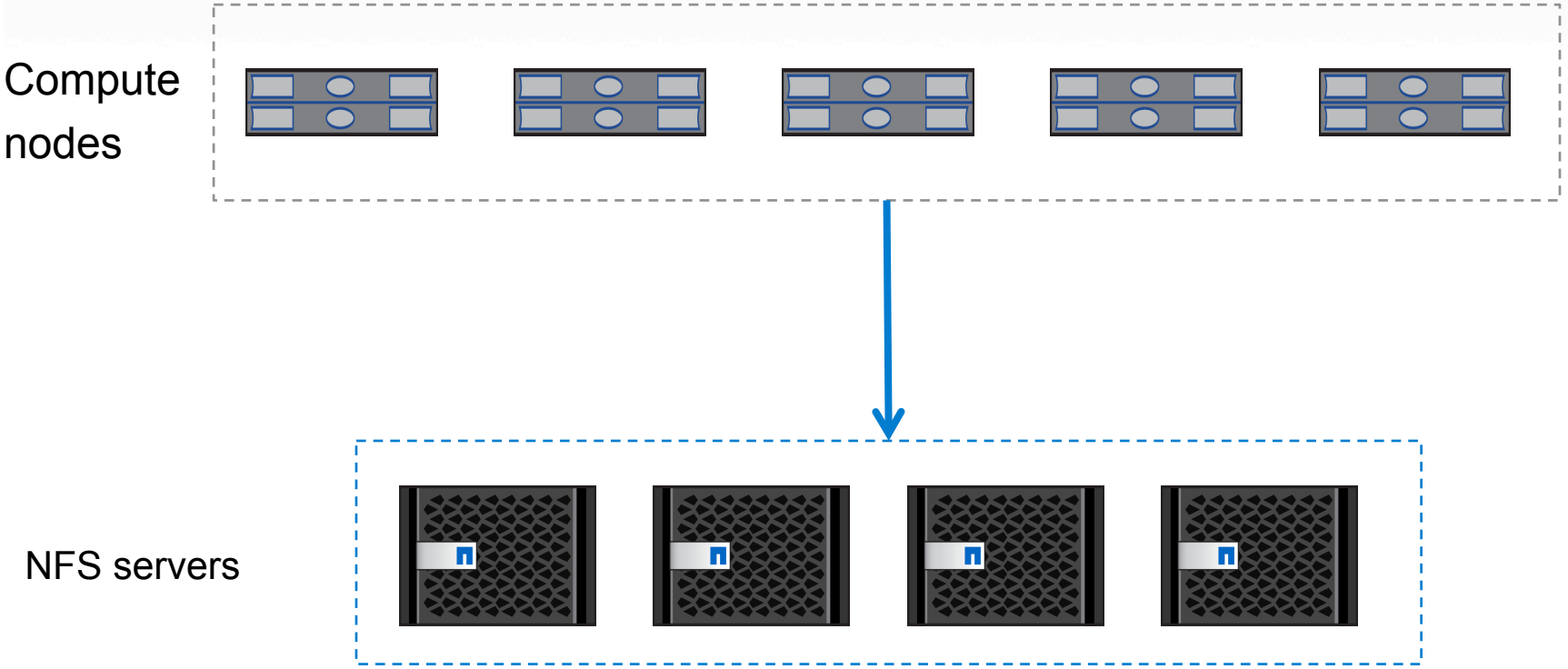
Optimize with Caching



Overcome NFS server bottleneck

Optimize with Caching

- Real workloads are cacheable[1]

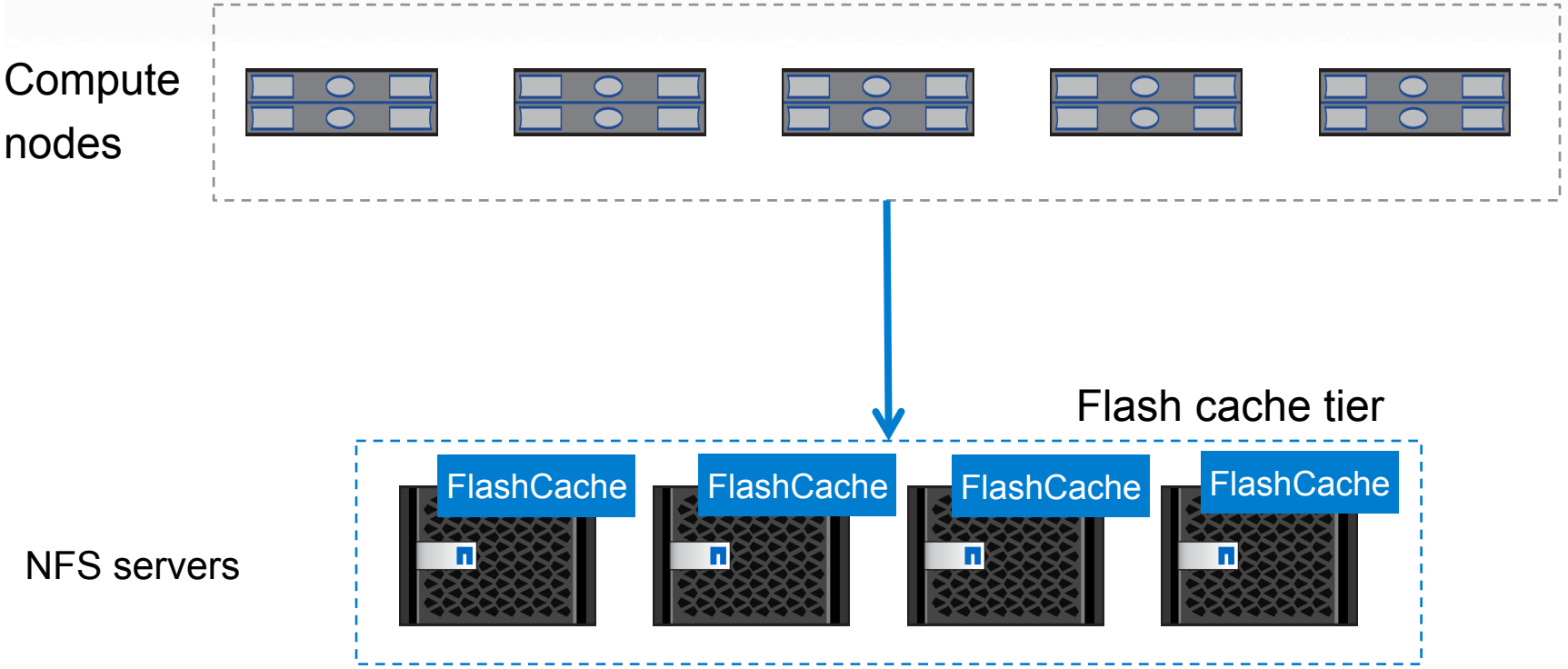


[1]MixApart: De-coupled Analytics for Shared Storage Servers.
Madalin Mihailescu, Gokul Soundararajan, and Cristiana Amza. In FAST '13

Overcome NFS server bottleneck

Optimize with Caching

- Real workloads are cacheable[1]

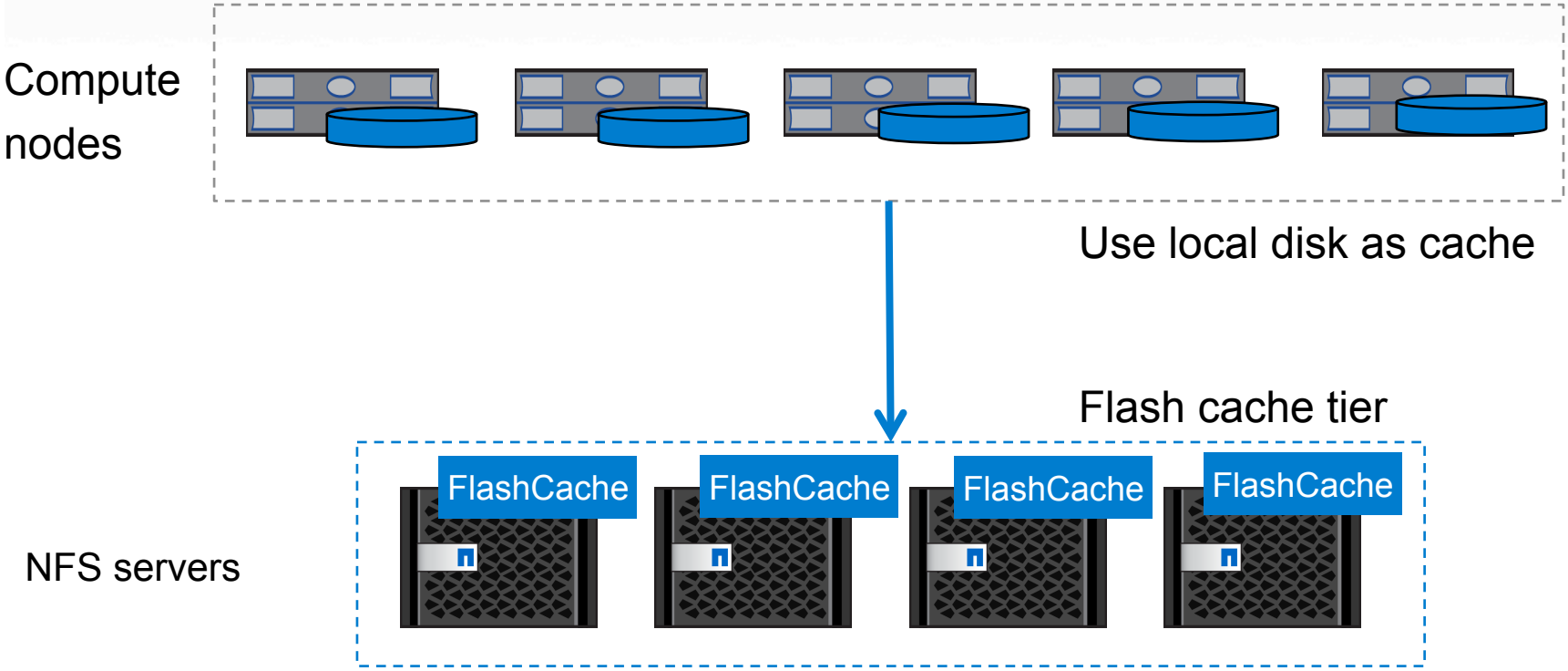


[1]MixApart: De-coupled Analytics for Shared Storage Servers.
Madalin Mihailescu, Gokul Soundararajan, and Cristiana Amza. In FAST '13

Overcome NFS server bottleneck

Optimize with Caching

- Real workloads are cacheable[1]

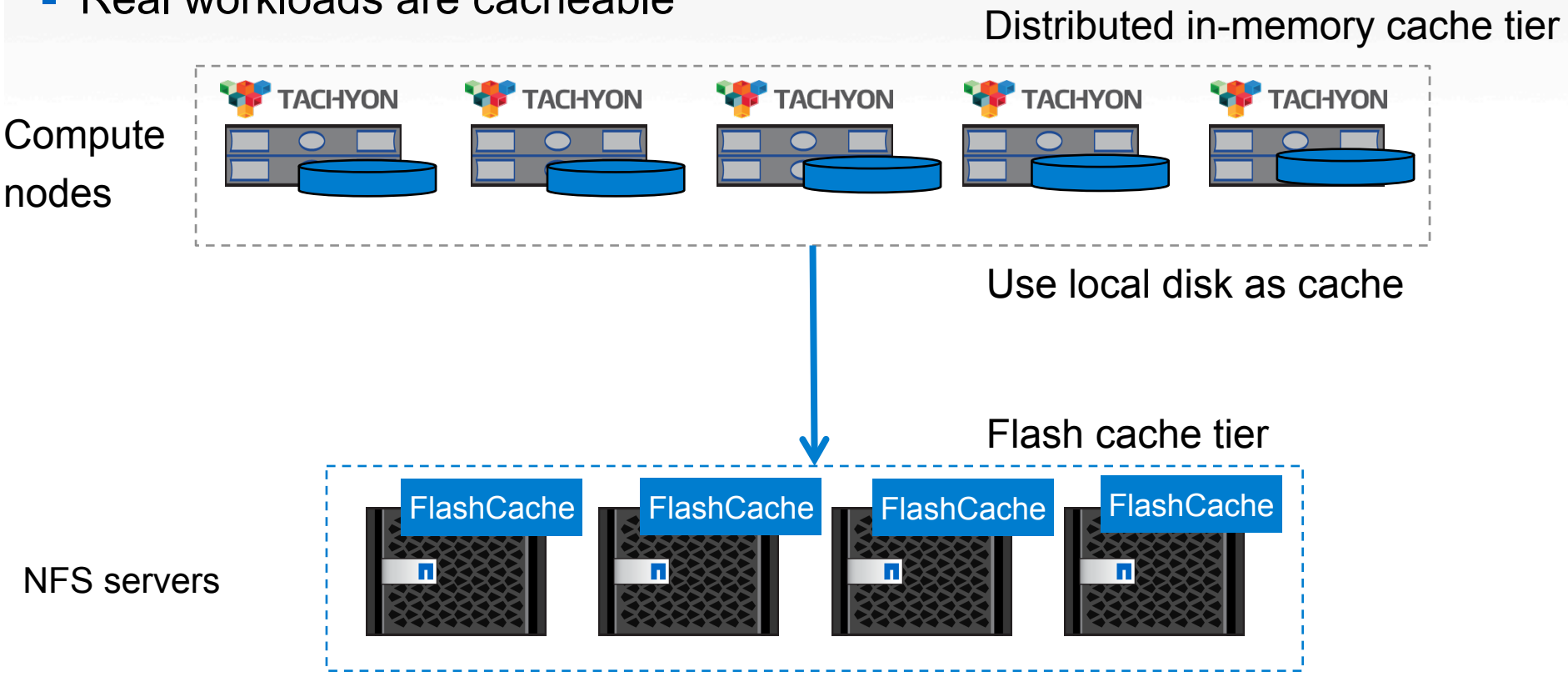


[1]MixApart: De-coupled Analytics for Shared Storage Servers.
Madalin Mihailescu, Gokul Soundararajan, and Cristiana Amza. In FAST '13

Overcome NFS server bottleneck

Optimize with Caching

- Real workloads are cacheable[1]



[1]MixApart: De-coupled Analytics for Shared Storage Servers.
Madalin Mihailescu, Gokul Soundararajan, and Cristiana Amza. In FAST '13

Next steps

We need your help.

Future Work

- Productization within NetApp
  - Support pNFS protocol
  - Security（Kerberos authentication)

Future Work

- Productization within NetApp
  - Support pNFS protocol
  - Security（Kerberos authentication)
- Integration tests with other frameworks
  - Tachyon, HBase, Spark, and etc.

Future Work

- Productization within NetApp
  - Support pNFS protocol
  - Security（Kerberos authentication)
- Integration tests with other frameworks
  - Tachyon, HBase, Spark, and etc.
- Production System Integration
  - NetApp Auto Support (ASUP) Team
  - Customer systems

We Need Your Help

- Anyone interested
  - Try it out and tell us how it works
  - Filing bugs
- Hadoop committers
  - Help to push NFS connector into Hadoop mainstream
- Help integration tests with other frameworks (Tachyon, HBase, etc)
- Help to improve the code at GitHub!

References

- Connector Information
  - http://www.netapp.com/us/solutions/big-data/nfs-connector-hadoop.aspx
- Public on GitHub:
  - https://github.com/NetApp/NetApp-Hadoop-NFS-Connector
- Technical Report:
  - http://www.netapp.com/us/media/tr-4382.pdf
- Paper at FAST'13
  - MixApart: De-Coupled Analytics for Shared Storage Servers
- If you have any question, please contact
  - Xing.Lin@netapp.com, Gokul.Soundararajan@netapp.com, Jingxin.Feng@netapp.com

Summary

- NetApp NFS connector for Hadoop
  - Allows analytics to use any NFS
  - An open implementation (no proprietary code) – contribute back to Hadoop
  - Works with Apache Hadoop, Apache Spark, Tachyon, and Apache HBase
  - In many cases, only configuration file change is needed (no source code changes)

Summary

- NetApp NFS connector for Hadoop
  - Allows analytics to use any NFS
  - An open implementation (no proprietary code) – contribute back to Hadoop
  - Works with Apache Hadoop, Apache Spark, Tachyon, and Apache HBase
  - In many cases, only configuration file change is needed (no source code changes)
- NetApp NFS connector for Hadoop is being deployed
  - Internal testing with other teams
  - Testing with select customers

Acknowledgements

- Madalin Mihailescu for starting down this path
- Kaladhar Voruganti, Scott Dawkins, Jeff Heller, AJ Mahajan, and Siva Jayasenan for supporting this effort
- Karthikeyan Nagalingam for validation and customer PoCs
- NetApp AutoSupport team for testing it in production
- NetApp NFS team for continuing the effort

Thank you

Mambo: analyze enterprise data in-place