



Technical Report

# NetApp In-Place Analytics Module

## Joint Reference Architecture

Karthikeyan Nagalingam, NetApp  
Ali Bajwa, Cloudera

September 2019 | TR-4715



### Abstract

This document introduces the NetApp® In-Place Analytics Module for Apache Hadoop. This module enables open-source analytics frameworks such as Hadoop to run natively on NFS storage. The topics covered in this report include the configuration, underlying architecture, primary use cases, integration with Hadoop, and benefits of using Hadoop with NetApp ONTAP® data management software. The NetApp In-Place Analytics Module allows analytics to run NetApp FAS and AFF with ONTAP software. It is easy to install and works with Apache Hadoop, Apache Spark, Tachyon, Apache HBase, and products from major Hadoop distributors, enabling data on NFSv3 to be analyzed.

## TABLE OF CONTENTS

<b>1</b>	<b>Introduction .....</b>	<b>5</b>
1.1	Overview .....	5
1.2	NetApp In-Place Analytics Module 3.0 Features.....	6
<b>2</b>	<b>Hortonworks Overview .....</b>	<b>6</b>
2.1	Hortonworks Data Platform Overview .....	7
2.2	Hortonworks Data Platform Products.....	7
<b>3</b>	<b>Benefits and Use Cases .....</b>	<b>8</b>
3.1	Deployment Options .....	10
3.2	Ease of Deployment.....	10
<b>4</b>	<b>NetApp In-Place Analytics Module: Architecture and Design .....</b>	<b>10</b>
4.1	High-Level Architecture.....	10
4.2	Technical Advantages.....	12
4.3	Design.....	12
<b>5</b>	<b>Solution Architecture .....</b>	<b>15</b>
5.1	Network Architecture.....	15
5.2	Storage Architecture .....	16
5.3	Key Components of This Validated Configuration.....	18
<b>6</b>	<b>Installation and Configuration.....</b>	<b>18</b>
6.1	Configure FAS/AFF Storage Controller.....	19
6.2	Configure Hadoop Cluster.....	19
6.3	Create JSON Configuration File.....	19
6.4	Modify Hadoop Configuration.....	19
6.5	Verification .....	20
<b>7</b>	<b>Product Validation .....</b>	<b>21</b>
7.1	Basic Hadoop Functionality Validation .....	21
<b>8</b>	<b>Hadoop TeraGen and TeraSort Validation .....</b>	<b>22</b>
8.1	Hardware Configuration .....	22
8.2	JSON File Used for Testing .....	22
8.3	TeraGen and TeraSort Validation with AFF A300 Storage Controller .....	23
<b>9</b>	<b>Hive Validation .....</b>	<b>25</b>
9.1	Hive with MapReduce .....	25
9.2	Hive with Tez .....	25

<b>10 Spark Validation.....</b>	<b>27</b>
<b>11 HBase Validation .....</b>	<b>28</b>
<b>12 MetroCluster Validation .....</b>	<b>28</b>
12.1 Test Scenarios .....	29
<b>13 Hortonworks Certification .....</b>	<b>30</b>
<b>14 Solutions for Error Messages .....</b>	<b>31</b>
14.1 AUTH_ERROR .....	31
14.2 Could Not Parse File .....	32
14.3 Unsupported Verifier AUTH_SYS .....	32
14.4 Split Metadata Size Exceeds 10,000,000 .....	33
14.5 java.lang.OutOfMemoryError: Java Heap Space .....	33
14.6 Temp Space.....	33
14.7 Root User.....	33
14.8 Class org.apache.hadoop.netapp.fs.nfs.NFSv3FileSystem Not Found .....	34
14.9 Could Not Get Root File Handle for Endpoint .....	34
14.10 No Such File or Directory.....	34
<b>15 Conclusion .....</b>	<b>34</b>
<b>16 Acknowledgments.....</b>	<b>35</b>
<b>Where to Find Additional Information .....</b>	<b>35</b>

## LIST OF TABLES

Table 1) Components used for NetApp In-Place Analytics Module tests. ....	18
Table 2) Parameters for Hadoop configuration.....	19
Table 3) Test details. ....	21
Table 4) Hardware configuration details. ....	22
Table 5) Ecosystem components. ....	30

## LIST OF FIGURES

Figure 1) NetApp In-Place Analytics Module. ....	5
Figure 2) NetApp In-Place Analytics Module plugs into Apache Hadoop. ....	6
Figure 3) Cross-data center deployments. ....	9
Figure 4) Duplicate Hadoop cluster using NetApp FlexClone® technology.....	9
Figure 5) High-level architecture of NetApp In-Place Analytics Module with application execution sequence. ....	11
Figure 6) Components of the NetApp In-Place Analytics Module.....	12

Figure 7) Connection pool. ....	13
Figure 8) NFS input and output streams.....	14
Figure 9) LRU cache. ....	14
Figure 10) Network connectivity. ....	15
Figure 11) NFS volume from a single NetApp FAS/AFF storage controller.....	17
Figure 12) NFS volumes from two FAS/AFF storage controllers.....	17
Figure 13) TeraGen, TeraSort, and TeraValidate report. ....	24
Figure 15) Tested architecture. ....	29

# 1 Introduction

Big data is defined broadly as large datasets containing a mix of structured, unstructured, and semistructured data that cannot be processed by traditional database systems. Businesses have turned to big-data frameworks that support analytical tools such as Apache Hadoop to help store and analyze these datasets. Apache Hadoop software is a framework that enables the distributed processing of large and varied datasets across clusters of computers by using programming models. The Hadoop Distributed File System (HDFS) provides high throughput processing of application data.

Historically, Hadoop has been used primarily on incoming external data. However, there's also a need to use Hadoop on existing internal data, which is typically stored in NAS. This use case requires setting up another storage silo to host HDFS and then running Hadoop analytics on that storage. This process results in additional data management, reduced efficiency, and increased costs for moving the data between NAS and HDFS.

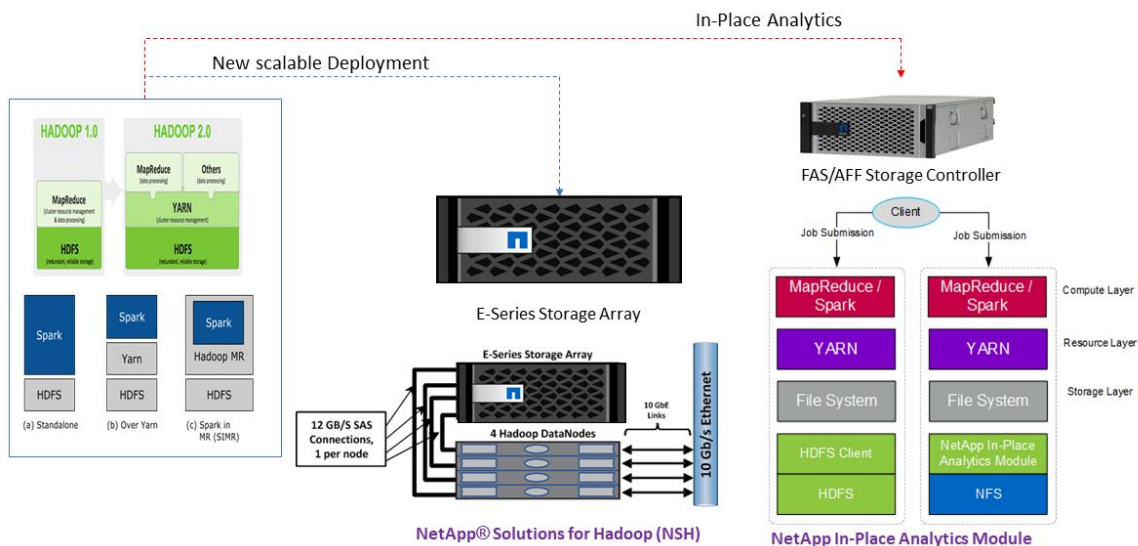
In this document, NetApp introduces the NetApp® In-Place Analytics Module. This module enables analytics software such as Apache Hadoop and Apache Spark to access and analyze data by using NetApp ONTAP® data management software and the NFS protocol. By using ONTAP software, the NetApp In-Place Analytics Module decouples analytics from storage, leveraging the benefits of NAS to share data.

This report also covers installation, the underlying architecture, use cases, integration with Hadoop, sizing information, benefits of using analytics with ONTAP software, Hortonworks certification, NetApp MetroCluster™ validation, and solutions for error messages.

## 1.1 Overview

NetApp provides NetApp E-Series and FAS arrays that are ideal for scalable Hadoop deployments. Both systems are complementary and target different use cases, as highlighted in Figure 1. This document focuses on the NetApp In-Place Analytics Module. For detailed information about the E-Series solution, see [TR-3969: NetApp Open Solution for Hadoop Solutions](#).

Figure 1) NetApp In-Place Analytics Module.



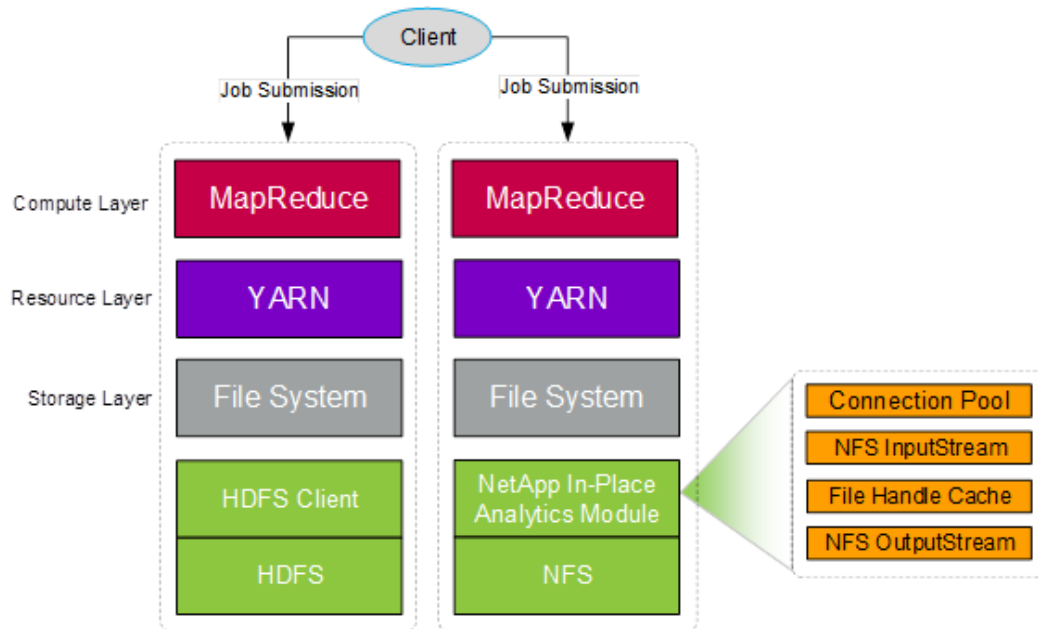
As mentioned, the NetApp In-Place Analytics Module enables analytics software to run on data stored with ONTAP software. It works with Apache Hadoop and Apache Spark by using a simple configuration file change that enables data on NFSv3 storage to be analyzed. By using ONTAP, the NetApp In-Place

Analytics Module decouples analytics from storage, taking advantage of NAS benefits. ONTAP data management software enables high availability and storage efficiency. For even better performance, the NetApp In-Place Analytics Module can be combined with Tachyon to build a scale-out caching tier that is backed by ONTAP software.

The NetApp In-Place Analytics Module is an implementation of the file system interface defined by Apache Hadoop that is based on NFSv3. Currently, it supports the `AUTH_NONE` and `AUTH_SYS` security policies, which are UNIX authentication mechanisms.

Figure 2 shows how the NetApp In-Place Analytics Module plugs into Apache Hadoop. This design has two deployment options, which are covered in section 3.1 “Deployment Options.”

**Figure 2) NetApp In-Place Analytics Module plugs into Apache Hadoop.**



## 1.2 NetApp In-Place Analytics Module 3.1 Features

NetApp In-Place Analytics Module 3.1 has the following features:

- **Performance enhancements.** Network load is distributed across multiple network interfaces.
- **Hortonworks certification.** Hortonworks certification is provided for the Hortonworks Data Platform (HDP).
- **Apache Ambari module for management.** Manage installation and configuration from the Ambari UI.
- **Simplicity enhancements.** User, access, and configuration management are included.
- **Azure integration with HDInsight application.** Run an Azure HDInsight cluster on NFS data.
- **Kerberos and Ranger support included.**

## 2 Hortonworks Overview

In 2011, Rob Bearden partnered with Yahoo to establish Hortonworks with 24 engineers from the original Hadoop team, including founders Alan Gates, Arun Murthy, Devaraj Das, Mahadev Konar, Owen O'Malley, Sanjay Radia, and Suresh Srinivas. Under the leadership of Arun Murthy, chief products officer, this core product team has been enriched with enterprise software talent from the likes of Oracle, IBM,

HP, VMware, and others to help ensure that HDP and HDF meet the enterprise-grade requirements our customers expect. Hortonworks is headquartered in Santa Clara, California, and the business model is based on open-source software support subscriptions, services, solutions, training, and consulting services.

Hortonworks operates in 19 countries with approximately 1,110 employees. There are 29 out of 114 Apache Hadoop committers from Hortonworks and 208 committer seats across 20+ Apache projects, and they focus on the data access, security, operations, and governance needs of the enterprise Hadoop market.

Hortonworks is a leading innovator in the industry, creating, distributing, and supporting enterprise-ready open data platforms and modern data applications. Its mission is to manage the world's data. It has a single-minded focus on driving innovation in open-source communities such as Apache Hadoop, NiFi, and Spark. Hortonworks, along with its 1,600+ partners, provides the expertise, training, and services that allow customers to unlock transformational value for their organizations across lines of business. Hortonworks's connected data platforms power modern data applications that deliver actionable intelligence from all data: data in motion and data at rest.

## **2.1 Hortonworks Data Platform Overview**

HDP is a secure, enterprise-ready, open-source Apache Hadoop distribution based on a centralized architecture (YARN). HDP addresses the complete needs of data at rest, powers real-time customer applications, and delivers robust big data analytics that accelerate decision making and innovation.

## **2.2 Hortonworks Data Platform Products**

HDP products are divided into six categories: data management, data access, data governance and integration, security, operations, and cloud.

### **Data Management**

YARN and HDFS are the cornerstone components of HDP for data at rest. Whereas HDFS provides scalable, fault-tolerant, cost-efficient storage for your data lake, YARN provides the centralized architecture that enables you to process multiple workloads simultaneously. YARN provides resource management and a pluggable architecture for a wide variety of data-access methods.

### **Data Access**

HDP includes a versatile range of processing engines that allow you to interact with the same data in multiple ways at the same time. Therefore, applications for big data analytics can interact with the data in an optimal way, from batch to interactive SQL or low latency access with NoSQL. Emerging use cases for data science, search, and streaming are also supported with Apache Spark, Storm, and Kafka.

### **Data Governance and Integration**

HDP extends data access and management with powerful tools for data governance and integration. These tools provide a reliable, repeatable, and simple framework for managing the flow of data in and out of Hadoop. This control structure, along with a toolset to ease and automate the application of schemas or metadata on sources, is critical for successful integration of Hadoop into your modern data architecture. Hadoop has engineering relationships with many leading data management providers that enable their tools to work and integrate with HDP. Data governance and integration are provided by Atlas, Falcon, Oozie, Sqoop, Flume, and Kafka components.

### **Security**

Security is woven and integrated into HDP in multiple layers. Critical features for authentication, authorization, accountability, and data protection help secure HDP across these key requirements.

Consistent with this approach throughout all the enterprise Hadoop capabilities, HDP also helps you integrate and extend your current security solutions to provide a single, consistent, secure umbrella over your modern data architecture. Authentication, authorization, and data protection are provided by Knox and Ranger components.

## Operations

Operations teams deploy, monitor, and manage Hadoop clusters within their broader enterprise data ecosystem, and Apache Ambari simplifies this experience. Ambari is an open-source management platform for provisioning, managing, monitoring, and securing the HDP that integrates Hadoop seamlessly into your enterprise environment. Ambari and Zookeeper manage a Hadoop cluster.

## Cloud

Cloudbreak, a part of HDP powered by Ambari, allows simplified provisioning and Hadoop cluster management in any cloud environment, including Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform, and OpenStack. It optimizes the use of cloud resources as workloads change.

## 3 Benefits and Use Cases

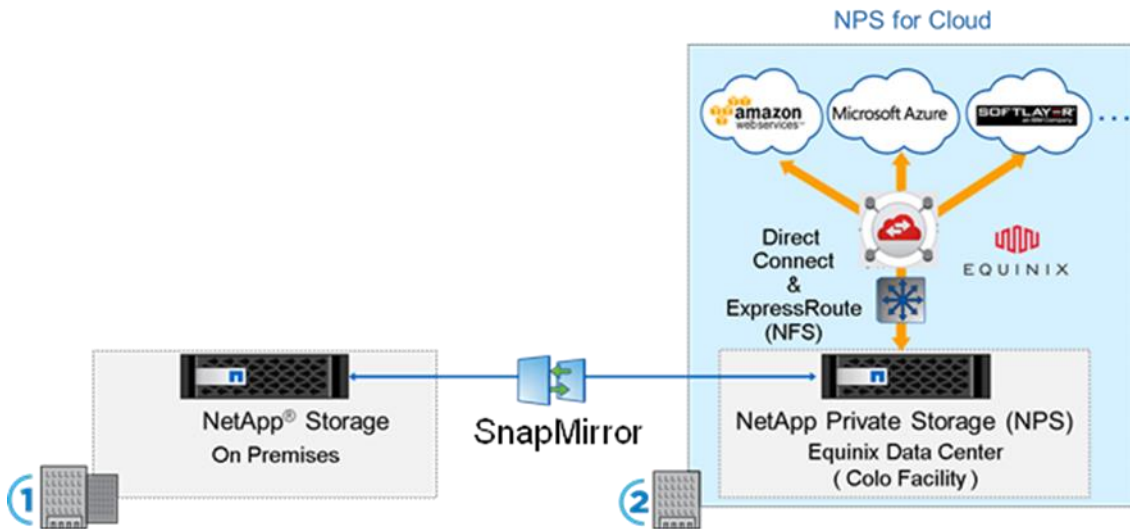
The NetApp In-Place Analytics Module allows you to analyze data stored on enterprise storage systems. The decoupled design of the NetApp In-Place Analytics Module provides high functional value in the following scenarios:

- **Analyzing data on enterprise storage.** Companies can use their existing investment in enterprise storage and enable analytics incrementally. Many file-based data sources exist, such as source-code repositories, emails, and log files. These files are generated by traditional applications but currently require a workflow to ingest the data into a separate analytics file system. The NetApp In-Place Analytics Module allows a single storage back end to manage and service data for both enterprise and analytics workloads. Data analytics that use the same file system namespace can analyze enterprise data with no additional ingest workflows.
- **Cross-data center deployments.** The decoupled design of the NetApp In-Place Analytics Module also allows independent scaling of compute and storage layers. As shown in Figure 3, this feature gives the flexibility of placing the analytics compute tier on cloud infrastructures, such as Amazon EC2, while keeping the data on premises. In this scenario, up-front hardware purchases are replaced by the pay-as-you-go cloud model. Cross-data center deployments benefit from products such as AWS Direct Connect and NetApp Private Storage (NPS) that enable high-bandwidth connections between private data centers and public clouds.

NPS enables enterprise customers to take advantage of the performance, availability, security, and compliance of NetApp storage with the economics, elasticity, and time-to-market benefits of the public cloud.



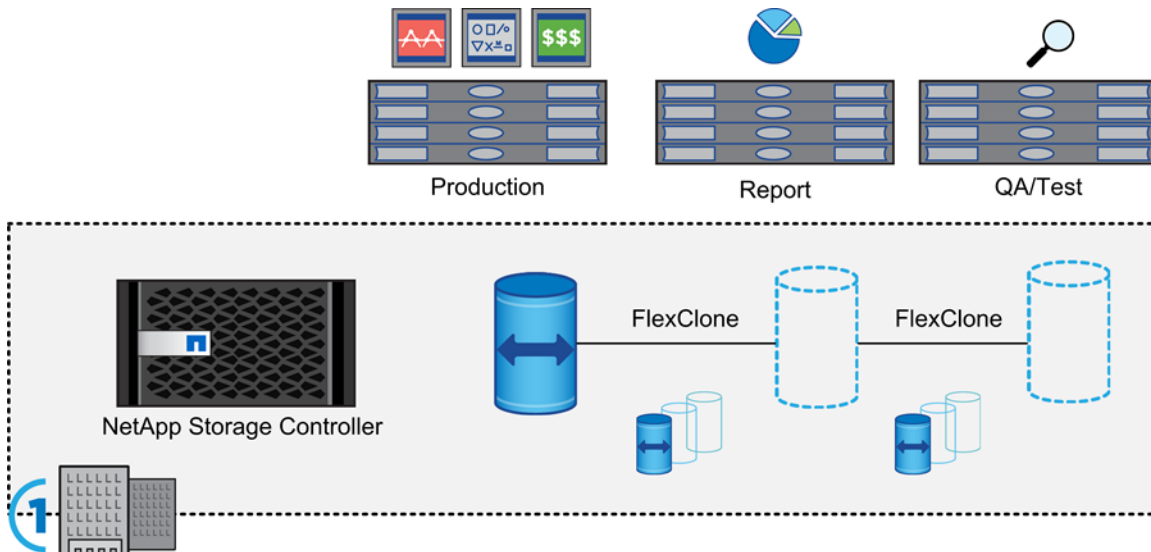
Figure 3) Cross-data center deployments.



The NetApp In-Place Analytics Module is optimal for the following use cases:

- **Analyze data on existing NFS storage.** The NetApp In-Place Analytics Module enables analytics on existing workflows and applications that write files to NFS, code repositories on NFS, and data in NetApp SnapLock® volumes.
- **Build testing and QA environments by using clones of existing data.** As shown in Figure 4, the NetApp In-Place Analytics Module enables developers to run variations of analytics code on shared datasets. If the production data is on a NetApp FlexVol® volume, the volume can be cloned and used for testing.

Figure 4) Duplicate Hadoop cluster using NetApp FlexClone technology.



- **Exploit storage-level caching for iterative machine-learning algorithms.** The iterative machine-learning algorithms are cache-friendly and compute-intensive. The NetApp In-Place Analytics Module can use storage caches, such as NetApp Flash Cache™ caches, for acceleration.

- **Use a backup site for analytics.** When data is stored near the cloud (NPS), the NetApp In-Place Analytics Module allows the use of cloud resources such as Amazon EC2 or Microsoft Azure for analytics while managing data with NetApp ONTAP software.

#### Additional Information

For a detailed list of data protection use cases, see [TR-4657: NetApp Hybrid Data Protection Solutions for Hadoop and Spark: Customer Use Case-Based Solutions](#).

### 3.1 Deployment Options

The NetApp In-Place Analytics Module allows users to run one of two deployment options:

- Run HDFS as the primary file system and use the NetApp In-Place Analytics Module to analyze data on the NFS storage systems.
- Deploy NFS as the primary storage or default file system.

You can base your deployment choice on the use cases and applications used in Apache Hadoop and Apache Spark.

#### Deploy NFS as Primary Storage or Default System

Even though some customers are running this option from the CLI, NetApp does not currently support this option in the NetApp In-Place Analytics Module 3.0 or later release. However, NetApp is actively working with Hadoop distributors to support this option in future versions.

The NetApp In-Place Analytics Module allows analytics to use existing technologies such as NetApp Snapshot™ copies, NetApp RAID DP® technology, NetApp SnapMirror® data replication, storage efficiency, and NetApp FlexClone® technology.

### 3.2 Ease of Deployment

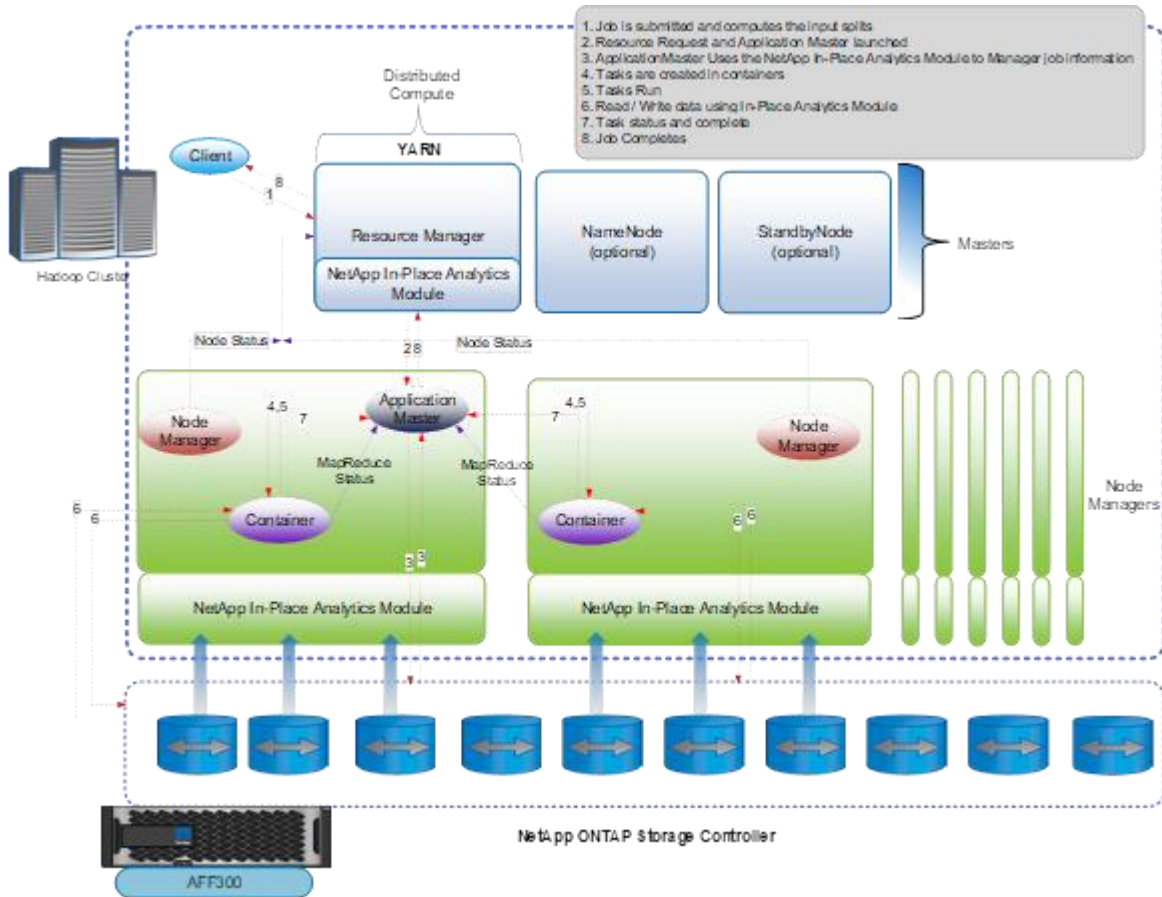
Installing the NetApp In-Place Analytics Module is simple. For Apache Hadoop, install the NetApp In-Place Analytics Module Java Archive (JAR) file and modify the `core-site.xml` file. A similar change is needed for Apache HBase: the `hbase-site.xml` file must be changed. After this modification, applications that use HDFS as their storage system can then use NFS.

## 4 NetApp In-Place Analytics Module: Architecture and Design

### 4.1 High-Level Architecture

Figure 5 shows a high-level architecture for the NetApp In-Place Analytics Module and the application execution sequence.

Figure 5) High-level architecture of NetApp In-Place Analytics Module with application execution sequence.



The high-level architecture of the NetApp In-Place Analytics Module can be explained through a client's application execution sequence:

1. The client program submits the application (called a MapReduce job). This application includes the specifications necessary to launch the application-specific ApplicationMaster. The program first computes the input splits. Then the ApplicationMaster coordinates and manages the lifetime of the job execution.
2. The ResourceManager assumes the responsibility to negotiate for a container in which to start the ApplicationMaster and then launches the ApplicationMaster.
3. The ApplicationMaster uses the NetApp In-Place Analytics Module to manage job information, such as status and logs. The ApplicationMaster requests containers for either its map or reduce tasks.
4. For each input split, the ApplicationMaster requests a container for the task analyzing the split and initiates the task in the newly created container.
5. The task, either map or reduce, runs in the container.
6. By using the NetApp In-Place Analytics Module, the task reads and/or writes data stored on NFS. As the task executes, its progress and status are updated periodically to the ApplicationMaster.
7. After the task is complete, it updates its completion status with the ApplicationMaster and exits. The container used by the task is given back to the ResourceManager.
8. After all tasks are complete, the ApplicationMaster updates various statistics and finishes the job. The client is notified of job completion.

## 4.2 Technical Advantages

The NetApp In-Place Analytics Module has the following technical advantages:

- It works with Apache Hadoop, Spark, HBase, Hive, and Mahout. It also works with Tachyon.
- No changes are needed to existing applications.
- No changes are needed to existing deployments; only the configuration files (`core-site.xml`, `hbase-site.xml`, and so on) are modified.
- You can modify and upgrade data storage nondestructively by using NetApp ONTAP software.
- It supports the latest networks (10/40GbE) and multiple NFS connections.
- The connector enables high availability and nondisruptive operations by using ONTAP software.

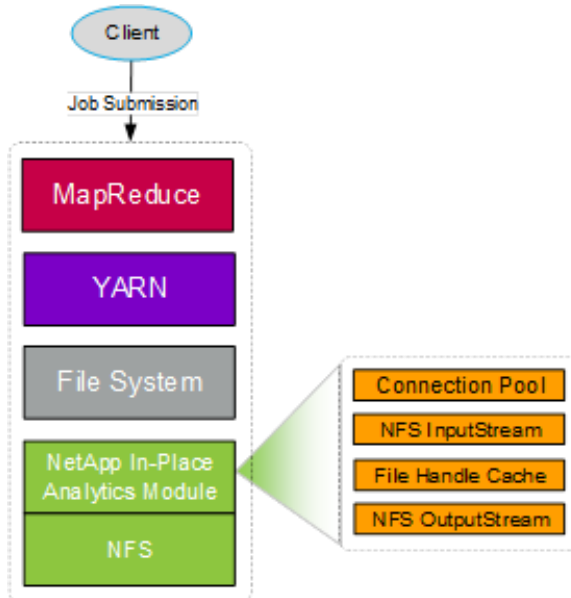
## 4.3 Design

Figure 6 shows the four main components of the NetApp In-Place Analytics Module:

- Connection pool
- NFS InputStream
- File handle cache
- NFS OutputStream

The other minor component is authentication.

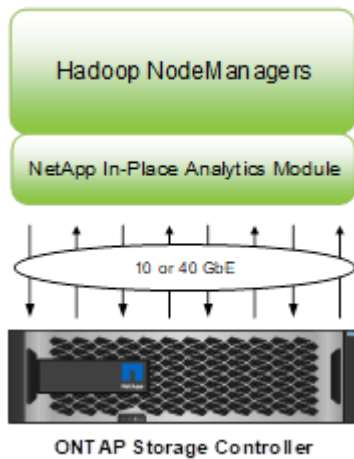
Figure 6) Components of the NetApp In-Place Analytics Module.



### Connection Pool

When the NetApp In-Place Analytics Module is loaded by Apache Hadoop, it creates a connection pool consisting of several connections to the NFS server. The size of the pool is dynamic. Later, all NFS traffic uses one of the connections available from the pool and multiplexes among them. This approach allows the NetApp In-Place Analytics Module to take advantage of high-speed (10/40GbE) networking and utilize aggregate bandwidth.

Figure 7) Connection pool.

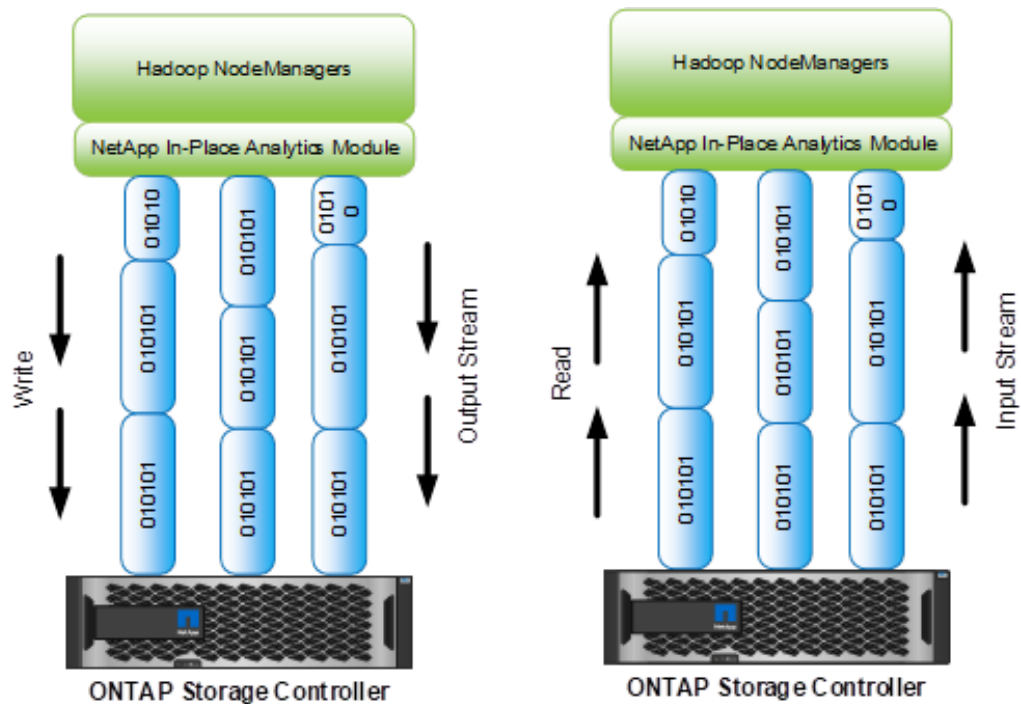


## NFS InputStream

InputStream is the method used by Apache Hadoop to read data from files. The NetApp In-Place Analytics Module is optimized to take full advantage of Hadoop and the underlying network and file system in the following ways:

- **Large sequential reads.** Applications use InputStream to read data. They read it in bytes required by the application, ranging from a single byte to several kilobytes of data. However, this method is not optimal for NFS. The connector modifies the I/O size issued to NFS to optimize for the underlying network. The default read value is 1MB, but you can configure it by setting the `nfsReadSizeBits` option. If the block size is larger than the maximum read request that the NFS server can support, then the NetApp In-Place Analytics Module automatically switches to using the smaller of the two.
- **Multiple outstanding I/Os.** The NetApp In-Place Analytics Module uses a temporary cache and issues multiple I/O requests in parallel. This method allows the amortization of the I/O time and enables the system to prefetch aggressively.
- **Prefetching.** Prefetching is used to improve the performance of streaming reads. When an on-demand read request is received, prefetching for the next 128 blocks is issued. To avoid unnecessary prefetching for Hadoop jobs, a heuristic is implemented. When a seek request is received, it sets the last block to prefetch, based on the offset of the seek request and the split size. The connector stops prefetching when it reaches that block. It never prefetches beyond the boundary of a file split. In the other case, in which the last block to prefetch is not set, the NetApp In-Place Analytics Module continues prefetching. The split size is configurable with the `nfsSplitSizeBits` option.

Figure 8) NFS input and output streams.



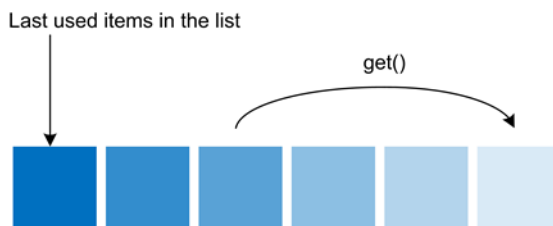
## File Handle Cache

A least recently used (LRU) cache is implemented to cache recently used file handles. This approach lowers the need to issue frequent lookup operations. It works as follows:

1. To get a file handle for a path, the file handle cache is checked.
2. If a handle is found in the cache, a lookup request is sent to the NFS server to check whether the file handle is valid or stale.
3. The handle is returned if it is valid. Otherwise, the same procedure is called to get a valid handle for the parent directory. This process is recursive, and it stops either when a valid handle is found for one of the ancestor directories or when the mount root directory is reached.
4. A lookup request for the file or directory in that parent directory is called.

The process repeats until it reaches the desired path.

Figure 9) LRU cache.



Calling `get()` for an item, moves it to the top of the cache

File Handle Cache

## NFS OutputStream

Similar to `InputStream`, Apache Hadoop uses `OutputStream` to write data to files. You can use similar optimizations such as batching writes for large I/Os and taking advantage of Hadoop's consistency semantics:

- A write buffer is maintained in the NetApp In-Place Analytics Module to store write requests. When the buffer becomes full, requests are sent to the NFS server. The size of the write buffer is configurable with the `nfsWriteSizeBits` option.
- The default mode used in write requests sent to the NFS server is nonoptimal because it requires the NFS server to make each write request durable in disk. Instead, the NetApp In-Place Analytics Module sends each write request to the NFS server as nondurable. A commit request is sent to the NFS server to flush all write requests only when the output stream is closed. This approach is one of the optimizations introduced specifically for running Hadoop jobs. There is no need to flush data to disk unless a task succeeds. Failed tasks are automatically restarted by Hadoop.

## Authentication

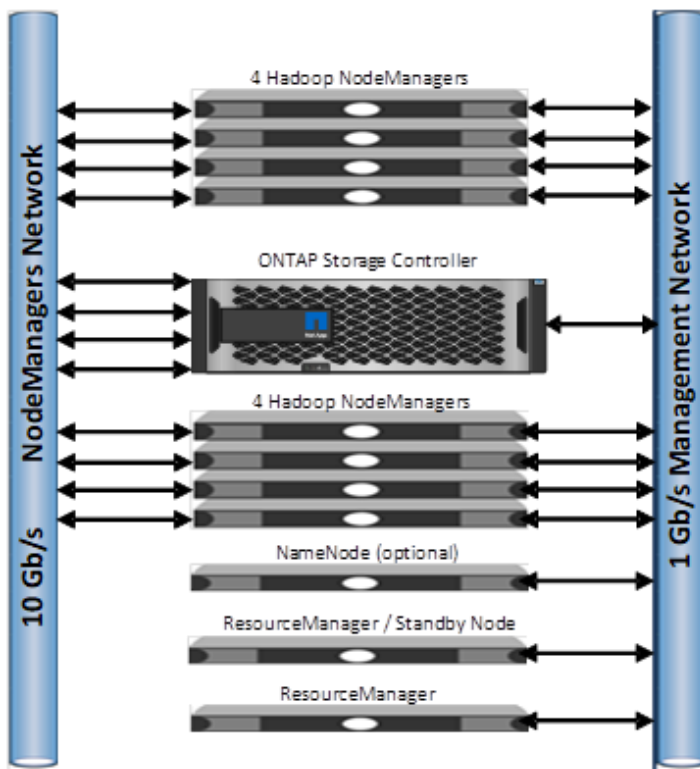
Currently, NetApp supports two types of authentication: none and UNIX. Authentication is configurable with the `nfsAuthScheme` option. NetApp is in the process of adding tighter integration with other authentication schemes such as NIS, Kerberos, and Ranger.

## 5 Solution Architecture

### 5.1 Network Architecture

Figure 10 shows the connectivity between the servers and the NetApp FAS/AFF storage controller.

Figure 10) Network connectivity.





## Recommended Network Topology

Apache Hadoop with the NetApp In-Place Analytics Module has the following network topology:

- NodeManagers (two or more 10/40GbE ports):
  - Primary purpose: data ingest and movement in a cluster.
  - Private interconnection between all NodeManagers.
  - Dedicated nonroutable virtual LAN (VLAN).
- Management network (GbE):
  - Purpose: system administration network.
  - Publicly routable subnet.
  - NodeManagers and ResourceManager (in Hadoop 2.0) also use GbE.
  - GbE interface for administration and data-transfer purposes on all nodes.
  - Two GbE ports required by FAS/AFF storage systems are for management purposes only.

## 5.2 Storage Architecture

The NetApp FlexVol flexible volume technology decouples the physical connection between data containers and their associated physical disks. The result is a significant increase in flexibility and storage utilization. This technology can shrink or grow data containers based on immediate needs. You can add disks or change the size of data containers as needed, without disrupting the system and associated applications.

- On a NetApp ONTAP system, NAS clients access flexible volumes through a storage virtual machine (SVM, previously called a Vserver). SVMs abstract storage services from their underlying hardware.
- Flexible volumes containing NAS data are junctioned into the owning SVM in a hierarchy. This hierarchy presents NAS clients with a unified view of the storage, regardless of the physical location of the flexible volumes inside the cluster.
- When a flexible volume is created in the SVM, the administrator specifies the junction path for the flexible volume. The junction path is a directory location under the root of the SVM where the flexible volume can be accessed. A flexible volume's name and junction path do not need to be the same.
- Junction paths allow each flexible volume, for example, a directory or folder, to be browsable. NFS clients can access multiple flexible volumes using a single mount point. CIFS clients can access multiple flexible volumes using a single CIFS share.
- A namespace consists of a group of volumes connected using junction paths. It is the hierarchy of flexible volumes in a single SVM as presented to NAS clients.
- The storage architecture can consist of one or more FAS/AFF storage controllers. Figure 11 shows a single NetApp FAS/AFF controller with a volume mounted under its default or own namespace.



Figure 11) NFS volume from a single NetApp FAS/AFF storage controller.

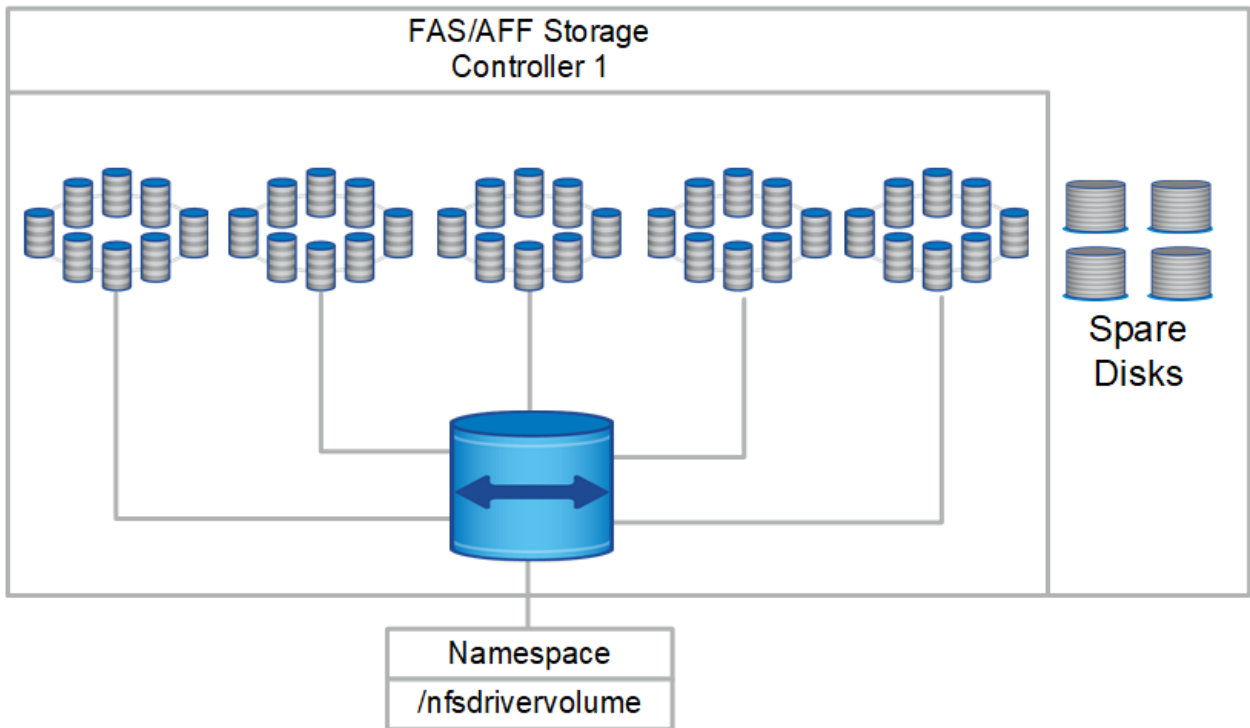
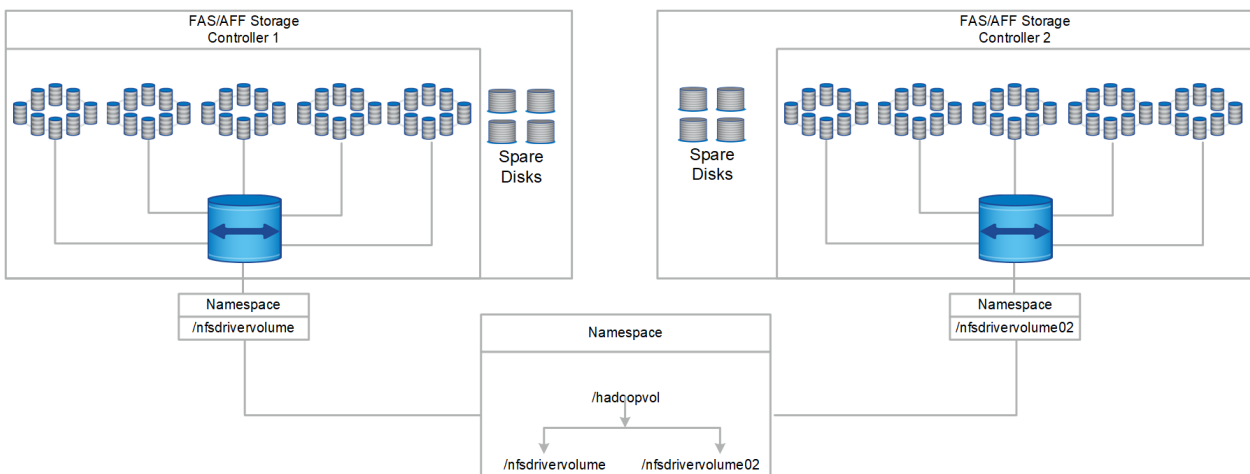


Figure 12 shows a FAS/AFF controller, in which each volume from each controller is mounted under one common namespace. The common namespace is a dummy volume that can be from any one of the controllers, and it's configured as `nfsExportPath` in `core-site.xml` in the Hadoop cluster.

#### Best Practice

NetApp recommends creating a volume with NetApp RAID DP, adding more disks for better performance, and keeping two disks as global hot spares for up to 100 disks of the same type.

Figure 12) NFS volumes from two FAS/AFF storage controllers.



In a dual-controller setup, you can distribute the load among the controllers during job execution. For example, you can store TeraGen map job results and perform a TeraSort map operation on one volume

followed by a reduce operation, based on TeraGen results, on another volume. You can also perform a TeraGen operation on one volume while a TeraSort operation is running on another.

#### Best Practice

NetApp recommends having an application cache (`usercache`) in a NetApp storage volume and creating one volume per NodeManager.

### 5.3 Key Components of This Validated Configuration

Table 1 lists the products used for the validation of the NetApp In-Place Analytics Module. You can configure the storage based on the [NetApp sizer](#) recommendations for your bandwidth (in Mbps), IOPS, and capacity requirements.

Table 1) Components used for NetApp In-Place Analytics Module tests.

Component	Product or Solution	Details
Storage	NetApp AFF A300 storage array with ONTAP 9.3	<ul style="list-style-type: none"><li>2 controllers (HA pair) with 24 x 900GB solid-state drives (SSDs)</li><li>1 hot spare per disk shelf</li><li>1 data aggregate (23 drives shared with both controllers) per controller</li><li>NetApp ONTAP FlexGroup and some test cases with 2 aggregates and 2 volumes</li><li>2 x 4 x 10GbE</li><li>RAID DP</li></ul>
Servers	PRIMERGY RX2540 M1 Intel Xeon CPU E5-2670 v3 at 2.30GHz	<ul style="list-style-type: none"><li>10 servers, each with two 2.4GHz (6-core) or 2.3GHz (8-core) processors (40 cores)</li><li>12 DIMMs (up to 192GB), up to 1600MHz (512GB/node)</li><li>1 x 1Gbps Ethernet port, 2 x 10GbE network ports</li></ul>
Networking	<ul style="list-style-type: none"><li>10GbE nonblocking network switch</li><li>1GbE network switch</li></ul>	A Cisco Nexus 5000 switch was used for testing. Any compatible 10GbE network switch can also be used.
Server operating system	Red Hat Enterprise Linux Server 7.4 (x86_64) or later	Hadoop typically requires a Linux distribution.
Hadoop distribution	Hortonworks Data Platform 3.1	Apache Ambari 2.7

## 6 Installation and Configuration

NetApp In-Place Analytics Module installation is simple and consists of three parts:

1. Configure the FAS/AFF storage controller.
2. Configure the Hadoop cluster.
3. Create the JSON configuration file.

The first step is to download the NetApp In-Place Analytics Module software and installation guide from the [NetApp website](#).

## 6.1 Configure FAS/AFF Storage Controller

To configure the FAS/AFF storage controller, complete the following steps:

1. Create an SVM with NFS access and then disable both `nfs-rootonly` and `mount-rootonly`.
2. In the SVM, change the `guid` from 1 to 0 by using the `unix-user modify` command.
3. In the SVM, check that the `export-policy` rule has access (`ip range`) for the Hadoop worker nodes and that the `superuser security type` is set to `sys`.
4. Create a volume and one or more logical network interfaces for the data role.
5. Change the NFS read and write size to 1MB by using `tcp-max-xfer-size` or `v3-tcp-max-write/read-size`.

## 6.2 Configure Hadoop Cluster

To configure the Hadoop cluster for Hortonworks, NetApp recommends using the [NetApp Ambari UI Service plug-in](#).

## 6.3 Create JSON Configuration File

The NetApp In-Place Analytics Module requires the creation of a configuration file, which is then distributed to all the nodes on your Hadoop cluster. The location of the file can be anywhere on your Hadoop node as long as it is accessible to all the Hadoop users in the system. See the [Installation Guide](#) for a detailed explanation of the JSON file and its parameters.

## 6.4 Modify Hadoop Configuration

Modify the `core-site.xml`, `hbase-site.xml`, and `hive-site.xml` files for Hadoop, Spark, and Hive. Override the following existing parameters for Hortonworks using the Ambari framework.

Table 2) Parameters for Hadoop configuration.

Parameter	Value	Description
<code>fs.nfs.prefetch</code>	<code>True</code>	Enables prefetch for InputStream.
<code>fs.defaultFS</code>	<code>nfs://192.168.120.207:2049</code>	Name of the default file system specified as a URL (this configuration works for some Hadoop ecosystems). The current version of the NetApp In-Place Analytics Module does not support this option.
<code>fs.AbstractFileSystem.nfs.impl</code>	<code>org.apache.hadoop.netapp.fs.nfs.NFSv3AbstractFileSystem</code>	Allows Hadoop 2.0 and later to create connectivity for NFS.
<code>fs.nfs.impl</code>	<code>org.apache.hadoop.netapp.fs.nfs.NFSv3FileSystem</code>	Allows Hadoop to find the NetApp NAS NFS connector.
<code>fs.nfs.configuration</code>	<code>/etc/NetApp/conf/nfs-mapping.json</code>	Defines the cluster architecture for the NetApp In-Place Analytics Module. Make sure the JSON file exists before restarting the required services in Hadoop.

## Best Practices

- Create separate networks for your Hadoop jobs between the NetApp NFS volumes and NodeManager for improved network bandwidth.
- Spread the NetApp NFS volumes equally across the controllers to distribute the load or create FlexGroup volumes.

## Example Configuration File (Version 3.x)

A configuration file for the `nfs-mapping.json` file is shown in the following example:

```
[root@node1 ~]# cat /etc/NetAppNFSConnector/conf/nfs-mapping.json
{
  "spaces": [
    {
      "endpoints": [
        {
          "exportPath": "/iam_volume2",
          "hosts": [
            "nfs://10.63.150.213:2049/",
            "nfs://10.63.150.127:2049/",
            "nfs://10.63.150.128:2049/",
            "nfs://10.63.150.211:2049/"
          ],
          "path": "/iam_volume2"
        }
      ],
      "name": "iam_validation",
      "options": {
        "nfsAuthScheme": "AUTH_SYS",
        "nfsExportPath": "/iam_volume2",
        "nfsMountPort": -1,
        "nfsPort": 2049,
        "nfsReadSizeBits": 20,
        "nfsRpcbindPort": 111,
        "nfsSplitSizeBits": 30,
        "nfsWriteSizeBits": 20
      },
      "uri": "nfs://10.63.150.213:2049/"
    }
  ]
}
```

To implement NIPAM with Ranger, see [TR-4744: Secure Hadoop using Apache Ranger with NetApp In-Place Analytics Module](#).

To implement NIPAM with Kerberos, see [TR-4797: NetApp In Place Analytics Module with Kerberos](#).

## 6.5 Verification

To verify installation and configuration of the NetApp In-Place Analytics Module, complete the following steps:

1. Use `copyFromLocal` to test the NetApp In-Place Analytics Module.

```
[root@node1 ~]#hadoop fs -ls  nfs://10.63.150.213:2049/
Found 4 items
drwxrwxrwx - root root      4096 2018-05-11 04:31 nfs://10.63.150.213:2049/.snapshot
drwxrwxrwx - root root      4096 2018-05-11 06:28 nfs://10.63.150.213:2049/tg
drwxrwxrwx - root root      4096 2018-05-11 06:29 nfs://10.63.150.213:2049/ts
drwxrwxrwx - root root      4096 2018-05-11 06:30 nfs://10.63.150.213:2049/tv
[root@node1 ~]# hadoop fs -copyFromLocal /usr/share/doc/util-linux-ng-2.17.2/README
nfs://10.63.150.213:2049/testfile
```

2. Validate that the `copyFromLocal` operation was successful in the Hadoop file system.

```
[root@node1 ~]#hadoop fs -ls nfs://10.63.150.213:2049/
Found 5 items
drwxrwxrwx - root root      4096 2018-05-11 04:31 nfs://10.63.150.213:2049/.snapshot
-rw-r--r-- 1 root root      2865 2018-05-15 13:31 nfs://10.63.150.213:2049/testfile
drwxrwxrwx - root root      4096 2018-05-11 06:28 nfs://10.63.150.213:2049/tg
drwxrwxrwx - root root      4096 2018-05-11 06:29 nfs://10.63.150.213:2049/ts
drwxrwxrwx - root root      4096 2018-05-11 06:30 nfs://10.63.150.213:2049/tv
[root@node1 ~]#
```

## 7 Product Validation

To validate the NetApp In-Place Analytics Module for real-world deployments, NetApp used the TeraGen tool to generate a Hadoop dataset. NetApp then used the TeraSort tool to conduct a MapReduce/YARN process to verify that the configuration worked as expected.

The following subsections present the details of each test NetApp conducted to validate the NetApp In-Place Analytics Module with Hadoop 3.1.

### 7.1 Basic Hadoop Functionality Validation

#### Setup Procedure

To set up basic validation testing of Hadoop functionality, we completed the following steps:

1. We removed any previous NFS artifacts before each run.
2. We verified that all components were restored to a nonfaulted condition.

#### Run Procedure

To run the validation testing, we used the included Apache TeraGen and TeraSort utilities, starting from the ResourceManager node.

**Note:** We used the same TeraGen and TeraSort parameters for all iterations.

Table 3 summarizes the test details.

Table 3) Test details.

Test Information	Details
Test type	Initial tuning and full function
Execution type	Automated
Configuration	<ul style="list-style-type: none"><li>• Memory configured in the command prompt <code>mapreduce.map.memory.mb = 32768</code></li><li>• TeraGen options <code>-Dmapreduce.job.maps=128</code></li><li>• TeraSort options <code>-Dmapreduce.job.reduces=360</code></li><li>• TeraValidate <code>-Dmapreduce.job.maps=360</code></li></ul>
Duration	Multiple runs, one day total
Description	This test runs a TeraGen job with a duration of more than 10 minutes to generate a substantial dataset. It then runs a TeraSort job on the dataset created by TeraGen.

Test Information	Details
Prerequisites	The NodeManager components were started.
Test results	<ul style="list-style-type: none"> <li>• Proper output results were received from the TeraSort reduce stage.</li> <li>• No tasks on individual task nodes (NodeManagers) failed.</li> <li>• The file system (NFS) maintained integrity and was not corrupted.</li> <li>• All test environment components were still running.</li> </ul>
Notes	<ul style="list-style-type: none"> <li>• We used integrated web and UI tools to monitor the Hadoop tasks and file system.</li> <li>• We used Ganglia to monitor the Linux server in general.</li> </ul>

## 8 Hadoop TeraGen and TeraSort Validation

### 8.1 Hardware Configuration

Table 4 summarizes the configuration details for the hardware.

Table 4) Hardware configuration details.

Component	Product or Solution	Details
Storage	NetApp AFF A300 storage array with NetApp ONTAP 9.3	<ul style="list-style-type: none"> <li>• 2 controllers (HA pair) with 24 x 900GB SSDs</li> <li>• 1 hot spare per disk shelf</li> <li>• 1 data aggregate (23 drives, shared to both controllers) per controller</li> <li>• FlexGroup volumes</li> <li>• 8 x 10GbE</li> </ul>
Servers	PRIMERGY RX2540 M1 Intel Xeon CPU E5-2670 v3 at 2.30GHz	<ul style="list-style-type: none"> <li>• 10 servers, each with two 2.4GHz (6-core) or 2.3GHz (8-core) processors (40 cores)</li> <li>• 12 DIMMs (up to 192GB), up to 1600MHz (512GB/node)</li> <li>• 1 x 1Gbps Ethernet port, 2 x 10GbE network ports</li> </ul>
Networking	<ul style="list-style-type: none"> <li>• 10GbE nonblocking network switch</li> <li>• 1GbE network switch</li> </ul>	We used a Cisco Nexus 5000 switch for testing. Any compatible 10GbE network switch can also be used.
Server operating system	Red Hat Enterprise Linux Server 7.4 (x86_64) or later	Hadoop typically requires a Linux distribution.
Hadoop distribution used in the testing	Hortonworks Data Platform 3.1	Apache Ambari 2.7

### 8.2 JSON File Used for Testing

We used the following JSON file for testing.

```
[root@stlrx2540m1-36 ~]# cat /etc/NetAppNFSCConnector/conf/nfs-mapping.json
{
  "spaces": [{
    "name": "santanderpoc1",
```

```

"uri": "nfs://10.63.150.65:2049/",
"options": {
  "nfsAuthScheme": "AUTH_SYS",
  "nfsExportPath": "/pocfg",
  "nfsMountPort": -1,
  "nfsPort": 2049,
  "nfsRpcbindPort": 111,
  "nfsReadSizeBits": 22,
  "nfsWriteSizeBits": 22,
  "nfsReadMinPoolThreads": 64,
  "nfsReadMaxPoolThreads": 256,
  "nfsWriteMinPoolThreads": 64,
  "nfsWriteMaxPoolThreads": 256,
  "nfsSplitSizeBits": 30
},
"endpoints": [
  {
    "hosts": [
      "nfs://10.63.150.58:2049/", "nfs://10.63.150.65:2049/"
    ],
    "path": "/pocfg",
    "exportPath": "/pocfg"
  }
]
}]
}

[root@stlrx2540m1-36 ~]#

```

### 8.3 TeraGen and TeraSort Validation with AFF A300 Storage Controller

In addition to the basic functionality and fault injection testing described in section 7.1, “Basic Hadoop Functionality Validation,” NetApp used the TeraGen and TeraSort tools to measure how well the Hadoop configuration performed when generating and processing considerably larger datasets. In these tests, we used TeraGen to create datasets that ranged in size from 100GB and 500GB to 1TB. We then used TeraSort to conduct a MapReduce function on each dataset using 10 nodes in the Hadoop cluster. We recorded the elapsed time required to complete the process and observed that the duration (in minutes) of TeraGen and TeraSort responses was directly proportional to the size of the datasets.

**Note:** For these tests, we did not attempt to maximize the performance of TeraGen and TeraSort. NetApp believes that the performance can be improved with additional tuning.

Figure 13 shows the elapsed time to create the different datasets with TeraGen. Creating a 1TB dataset took over 4 minutes, and no issues were logged during the TeraGen operations. Also, the time required to generate the datasets increased proportionally with the size of the dataset, indicating that the cluster maintained its data ingest rates over time.

**Note:** NetApp used one FlexGroup volume from the AFF A300 storage controllers for this testing with TeraGen, TeraSort, and TeraValidate.

See the details for the performance validation in Table 3.

Figure 13) TeraGen, TeraSort, and TeraValidate report.

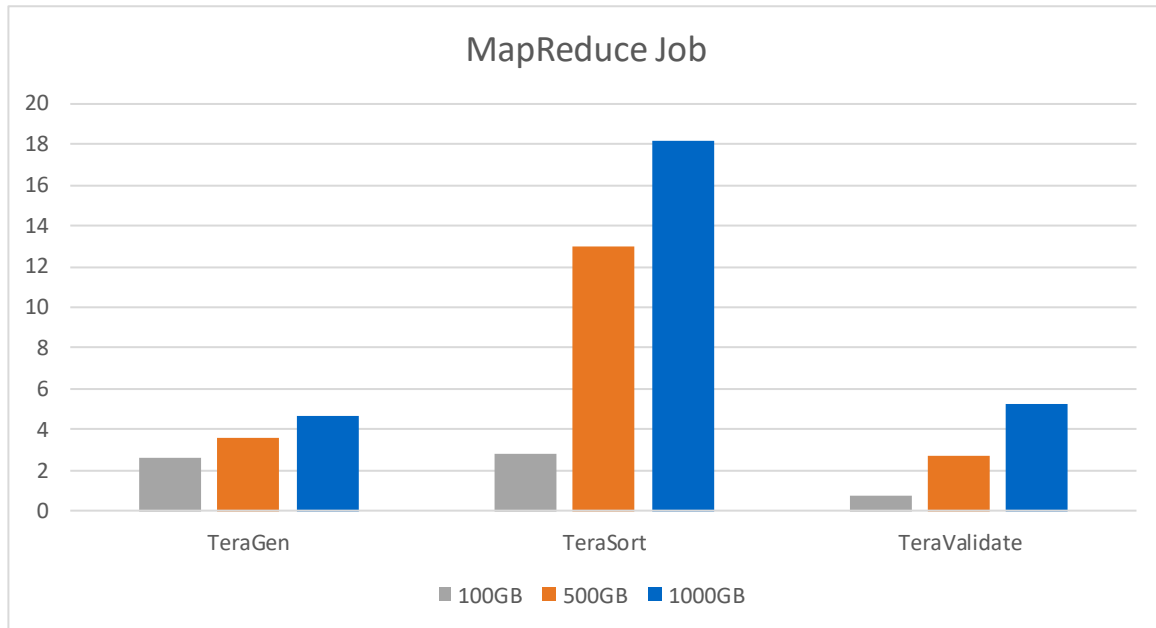


Figure 13 shows the time required to complete a TeraSort job on each of the increasingly larger datasets described in the preceding paragraphs. The 1TB dataset required 18 minutes to complete the process, and no issues were logged during the TeraSort operations. These results demonstrate that the Hadoop cluster maintained comparable processing rates as the size of the dataset increased. They also demonstrate the stability of the overall Hadoop cluster.

The tests were based on four Hadoop NodeManagers and one AFF A300 HA pair with two storage controllers. During the test, we observed that the storage controllers and disk utilization were less than 30%, which indicated that there was a lot of headroom in the storage system to perform additional operations.

Some Hadoop and Spark distributions must have the NetApp In-Place Analytics Module JAR files in the SQL Engines library path, and their XML files must be updated for the NetApp In-Place Analytics Module. You can use Ambari to update the configuration by following the path HDFS > configuration > custom core-site.xml > add sign.

The following configuration specifies that the Hadoop cluster is using NFS in parallel with the default (primary) file system.

```
<configuration>
<property>
  <name>fs.AbstractFileSystem.nfs.impl</name>
  <value>org.apache.hadoop.netapp.fs.nfs.NFSv3AbstractFileSystem</value>
</property>
<property>
  <name>fs.nfs.impl</name>
  <value>org.apache.hadoop.netapp.fs.nfs.NFSv3FileSystem</value> </property>
<property>
  <name>fs.nfs.configuration</name>
  <value>/<path-to-nfs-mapping-file>/nfs-mapping.json</value> </property>
</configuration>
<property>
  <name>fs.nfs.prefetch</name>
  <value>>false</value>
</property>
```

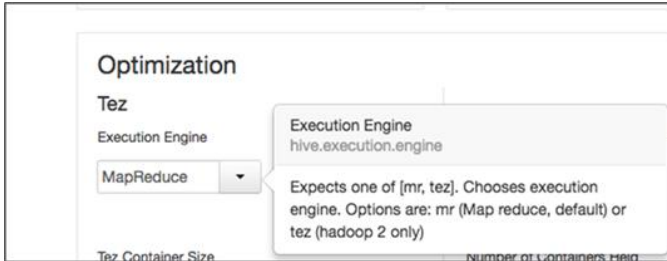


## 9 Hive Validation

Hive is similar to a SQL query, and it is often used for Apache Hadoop data warehouses. This section provides details about using the NetApp In-Place Analytics Module with Hive, MapReduce, and the Tez execution engine.

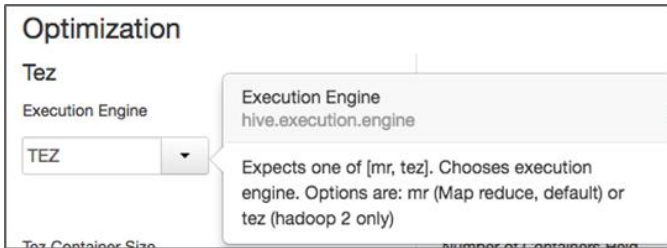
### 9.1 Hive with MapReduce

1. Select MapReduce from the Execution Engine drop-down list. By default, it points to Tez.



### 9.2 Hive with Tez

1. Select Tez from the Execution Engine drop-down list.

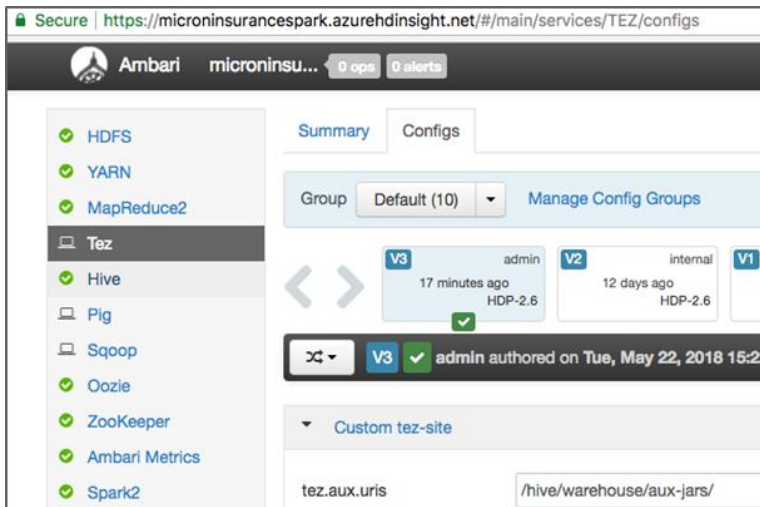


2. Check that `nfs export (nfs volume)` has Hive user and group permissions. Update it if needed.
3. Copy the NetApp In-Place Analytics Module JAR files to the `defaultFS` as configured in Ambari. For example, if `fs.defaultFS` points to HDFS, then copy the JAR files to the HDFS location. In HDInsight, if `fs.defaultFS` points to WASB, then copy the JAR files to the WASB location.

In the following example, the JAR files were copied to `/hive/warehouse`.

```
hdfs dfs -mkdir /hive/warehouse/aux-jars
hdfs dfs -copyFromLocal /<parentfolder>/hadoop-nfs-2.7.1.jar /hive/warehouse/aux-jars
hdfs dfs -copyFromLocal /<parent folder>/hadoop-nfs-connector-3.1.0.jar /hive/warehouse/aux-jars
```

4. Add the new auxiliary JAR location to the `tez-site.conf` file. The following screenshot shows an Ambari example.



The following example shows a `Tez-site.conf` manual edit.

```
<tez.aux.uris>
/{path to desired location}/aux-jars
</tez.aux.uris>
```

## 5. Restart the Tez, Hive, Oozie, and other services.

- The following example shows the `nfs-mapping.json` file used for the Hive test.

```
root@hn0-micron:~# cat /netappnfs/nfs-mapping.json
{
  "spaces": [
    {
      "name": "nfsserver",
      "uri": "nfs://172.18.10.68:2049/",
      "options": {
        "nfsExportPath": "/iamntapcloudvolume",
        "nfsReadSizeBits": 20,
        "nfsWriteSizeBits": 20,
        "nfsSplitSizeBits": 30,
        "nfsAuthScheme": "AUTH_SYS",
        "nfsUserConfigFile": "/netappnfs/users.json",
        "nfsGroupConfigFile": "/netappnfs/groups.json",
        "nfsUsername": "root",
        "nfsGroupname": "root",
        "nfsUid": 0,
        "nfsGid": 0,
        "nfsPort": 2049,
        "nfsMountPort": -1,
        "nfsRpcbindPort": 111
      },
      "endpoints": [
        {
          "path": "/iamntapcloudvolume",
          "exportPath": "/iamntapcloudvolume",
          "hosts": [
            "nfs://172.18.10.68:2049/"
          ]
        },
        {
          "path": "/cloudvolumetesting",
          "exportPath": "/cloudvolumetesting",
          "hosts": [
            "nfs://172.18.10.68:2049/"
          ]
        }
      ]
    }
  ]
}
```

```
]
}
root@hn0-micron:~#
```

See the Appendix section of the document [TR-4382: NetApp In-Place Analytics Module Best Practices](#) for a detailed Hive example.

## 10 Spark Validation

Apache Spark is a fast, general-purpose cluster computing system. It provides high-level APIs in Java, Scala, Python, and R and an optimized engine that supports general execution graphs. It also supports a rich set of higher-level tools, including [Spark SQL](#) for SQL and structured data processing, [MLlib](#) for machine learning, [GraphX](#) for graph processing, and [Spark Streaming](#).

For a Hortonworks distribution and Apache Spark, append `SPARK_CLASSPATH` or `SPARK_DIST_CLASSPATH` to the NetApp In-Place Analytics Module JAR files location in `spark-env` or `spark2-env`. See the following example:

```
export SPARK_CLASSPATH=/usr/hdp/3.1.0.0-78/spark/lib/*:/usr/hdp/3.1.0.0-78/oozie/share/lib/spark/*:/usr/hdp/3.1.0.0-78/hadoop/*:/usr/hdp/3.1.0.0-78/hadoop/client/*

export SPARK_DIST_CLASSPATH=$SPARK_DIST_CLASSPATH:/usr/hdp/current/spark2-client/jars/*:/usr/lib/hdinsight-datalake/*:/usr/hdp/current/spark_llap/*:/usr/hdp/current/spark2-client/conf:/usr/hdp/3.1.0.0-78/spark/lib/*:/usr/hdp/3.1.0.0-78/oozie/share/lib/spark/*:/usr/hdp/3.1.0.0-78/hadoop/*:/usr/hdp/3.1.0.0-78/hadoop/client/*
```

We tested Spark with the NetApp In-Place Analytics Module. See the following sample:

```
sshuser@hn0-micron:~$ sudo su -
root@hn0-micron:~# spark-shell
scala> val file =
sc.textFile("nfs://172.18.10.68:2049/iamntapcloudvolume/hivetest/MOCK_DATA.csv")
scala> val counts = file.flatMap(line => line.split(" ")).map(word => (word, 1)).reduceByKey(_ + _)
scala> counts.saveAsTextFile("nfs://172.18.10.68:2049/iamntapcloudvolume/hivetest/result")
scala> counts.count()
scala> :quit
root@hn0-micron:~# hadoop fs -ls nfs://172.18.10.68:2049/iamntapcloudvolume/hivetest
Found 4 items
-rwxrwxrwx   1 root root          56 2018-05-11 21:56
nfs://172.18.10.68:2049/iamntapcloudvolume/hivetest/000000_0
-rw-r--r--   1 root root       61981 2018-05-11 21:49
nfs://172.18.10.68:2049/iamntapcloudvolume/hivetest/MOCK_DATA.csv
drwxrwxrwx   - root root        4096 2018-05-11 22:52
nfs://172.18.10.68:2049/iamntapcloudvolume/hivetest/out
drwxrwxrwx   - root root        4096 2018-05-11 22:57
nfs://172.18.10.68:2049/iamntapcloudvolume/hivetest/result
root@hn0-micron:~# hadoop fs -ls nfs://172.18.10.68:2049/iamntapcloudvolume/hivetest/result
Found 3 items
-rw-r--r--   1 root root           0 2018-05-11 22:57
nfs://172.18.10.68:2049/iamntapcloudvolume/hivetest/result/_SUCCESS
-rw-r--r--   1 root root       33125 2018-05-11 22:57
nfs://172.18.10.68:2049/iamntapcloudvolume/hivetest/result/part-00000
-rw-r--r--   1 root root       32917 2018-05-11 22:57
nfs://172.18.10.68:2049/iamntapcloudvolume/hivetest/result/part-00001
root@hn0-micron:~#
```

See [TR-4570: NetApp Storage Solutions for Apache Spark](#) for detailed Spark validation and performance results.

## 11 HBase Validation

HBase, which is written in [Java](#), is an open-source, nonrelational, distributed database modeled after Google's [Bigtable](#). It was developed as a part of Apache Software Foundation's Apache Hadoop project and runs on top of [HDFS](#) and non-HDFS protocols such as NFS and S3, providing Bigtable-like capabilities for Hadoop. That is, it provides a fault-tolerant way of storing large quantities of [sparse](#) data (small amounts of information caught in a large collection of empty or unimportant data), such as finding the 50 largest items in a group of 2 billion records or finding the nonzero items representing less than 0.1% of a huge collection.

We performed the following two validations:

- We generated the HBase Hfiles with the test table and ran the HBase load process.
- We evaluate the HBase performance using the HBase performance evaluation.

See [TR-4382: NetApp In-Place Analytics Module Best Practices](#) for these two validations and performance results.

## 12 MetroCluster Validation

Based on requests from customers for active-active data protection across a site for the Hadoop cluster, we tested NetApp MetroCluster in our lab. This section covers the testing details.

The architecture shown in Figure 14 was used for the validation.

[illegible]

We completed the following test scenarios for the MetroCluster validation:









- See [TR-4382: NetApp In-Place Analytics Module Best Practices](#) for detailed test scenarios.







To view a detailed video about the MetroCluster validation, contact the author of this technical report.

## 13 Cloudera Certification

We certified the NetApp In-Place Analytics Module with Hortonworks for the ecosystem components described in Table 5 with Kerberos.

Table 5) Ecosystem components.

Component	Description
ZooKeeper 	Zookeeper provides operational services for a Hadoop cluster. It also provides a distributed configuration service, a synchronization service, and a naming registry for distributed systems. Distributed applications such as HBase and Hive use Zookeeper to store and mediate updates to important configuration information.
YARN MapReduce 	YARN is the architectural center of Hadoop that allows multiple data processing engines, such as interactive SQL, real-time streaming, data science, and batch processing, to handle data stored in a single platform, unlocking an entirely new approach to analytics. MapReduce is the original framework for writing applications that process large amounts of structured and unstructured data stored in the Hadoop Distributed Filesystem or Hadoop Compatible Distributed Filesystem.
HiveServer2, Hive, HiveServer2 Concur Hive LLAP 	Hive is the standard for interactive SQL queries over petabytes of data in Hadoop. It integrates with other data center technologies by using the Java Database Connectivity interface. <b>Note:</b> Hive 3.1.0 is included with HDP 3.1.
Hbase 	Hbase is a nonrelational (NoSQL) database that runs on top of HDFS. It provides real-time read/write access to large datasets. HBase is natively integrated with Hadoop and works seamlessly alongside data-access engines through YARN.
Pig 	Pig is a scripting platform for processing and analyzing the largest datasets. Pig translates the Pig Latin script into MapReduce so that it can be executed in YARN for access to a single dataset stored in HDFS. <b>Note:</b> We found an issue in Pig during certification. See details <a href="#">here</a> .
Spark, Hive on Spark 	Apache Spark adds in-memory compute for extract, transform, and load (ETL), machine learning, and data science workloads to Hadoop. Spark is a fast, in-memory data processing engine with elegant and expressive development APIs to allow data workers to efficiently execute streaming, machine learning, or SQL workloads that require fast iterative access to datasets. <b>Note:</b> Spark 2.3.2 is included with HDP 3.1.
Sqoop 	Apache Sqoop efficiently transfers bulk data between Apache Hadoop and structured datastores such as relational databases. Sqoop can also be used to extract data from Hadoop and export it into external structured datastores. Sqoop works with relational databases such as Teradata, Netezza, Oracle, MySQL, Postgres, and HSQLDB.
Phoenix, Phoenix Query Server 	Apache Phoenix is an open-source, massively parallel relational database engine that supports online transaction processing for Hadoop by using Apache HBase as its backing store.

Component	Description
Apache Oozie 	Apache Oozie is a Java web application used to schedule Apache Hadoop jobs. Oozie combines multiple jobs sequentially into one logical unit of work. It is integrated with the Hadoop stack and with YARN as its architectural center. It also supports Hadoop jobs for Apache MapReduce, Apache Pig, Apache Hive, and Apache Sqoop. Oozie can schedule jobs specific to a system, such as Java programs or shell scripts.
Apache Storm 	Apache Storm adds reliable real-time data processing capabilities to Enterprise Hadoop. Storm on YARN is powerful for scenarios that require real-time analytics, machine learning, and continuous monitoring of operations.
Ranger 	Apache Ranger is a framework to enable, monitor, and manage comprehensive data security across the Hadoop platform. With the advent of Apache YARN, the Hadoop platform can now support a true data lake architecture. Enterprises can potentially run multiple workloads in a multitenant environment. Data security within Hadoop needs to evolve to support multiple use cases for data access, while also providing a framework for central administration of security policies and monitoring of user access.
Ranger – NFS	Ranger with NIPAM for Network File Systems (NFS) is validated and certified with HDP 3.1.
Tez 	The Apache TEZ project provides an application framework to build a complex directed, acyclic graph of tasks for processing data.
Atlas 	Atlas is a scalable and extensible set of core foundational governance services that enable enterprises to meet their compliance requirements within Hadoop with complete integration into the enterprise data ecosystem. Apache Atlas provides open metadata management and governance for organizations to build a catalog of their data assets. They can then classify and govern these assets and enable collaboration for data scientists, analysts, and the data governance team.
Zeppelin 	Web-based notebook that enables interactive, data-driven analytics and collaborative documents with SQL, Scala, and more.

See the [Hortonworks](https://hortonworks.com) website for certification details. The old product name, NetApp FAS NFS Connector for Hadoop, was used during the certification process. Contact the author of this technical report for more information about the ecosystem components and their validation.

**Note:** WebHDFS, and Knox (because WebHDFS is not supported) are unsupported components for HDP 3.1. We are currently working on these unsupported components for the upcoming release of the NetApp In-Place Analytics Module.

**Note:** NIPAM is not supported or certified with MapR.

## 14 Solutions for Error Messages

This section provides solutions for error messages that NetApp encountered during the validation process.

### 14.1 AUTH\_ERROR

See the following example of the AUTH\_ERROR message:



```
Error: 18/01/25 12:48:19 ERROR rpc.RpcClient: RPC: xid=8000001 RpcReply request denied:
xid:8000001,messageType:RPC_REPLYverifier_flavor:AUTH_NONerejectState:AUTH_ERROR
18/01/25 12:48:19 ERROR mount.MountClient: Mount MNT operation failed with RpcException RPC:
xid=8000001 RpcReply request denied:
xid:8000001,messageType:RPC_REPLYverifier_flavor:AUTH_NONerejectState:AUTH_ERROR
```

## Solution

Disable mount-rootonly and nfs-rootonly on the NetApp SVM.

```
Cluster::> vserver nfs modify -vserver Hadoop_SVM -nfs-rootonly disabled
Cluster::> vserver nfs modify -vserver Hadoop_SVM -mount-rootonly disabled
Hadoop-AFF8080::> vserver nfs show -vserver Hadoop_SVM -fields nfs-rootonly
vserver      nfs-rootonly
-----
Hadoop_SVM disabled
Hadoop-AFF8080::> vserver nfs show -vserver Hadoop_SVM -fields mount-rootonly
vserver      mount-rootonly
-----
Hadoop_SVM disabled
```

## 14.2 Could Not Parse File

See the following example of the could not parse file message:

```
Could not parse config file /etc/NetAppNFSCConnector/conf/nfs-mapping.json
```

## Solution

Check <https://jsonlint.com/> to see if the JSON file is valid, because it might have a syntax error.

## 14.3 Unsupported Verifier AUTH\_SYS

See the following example of the Unsupported verifier AUTH\_SYS message:

```
ERROR rpc.RpcClientHandler: NfsConnectorV3.1.0 RPC: Got an exception
java.lang.UnsupportedOperationException: Unsupported verifier flavorAUTH_SYS
```

## Solution

Modify superuser security types to sys in the NetApp SVM.

```
stlaff300-1and2::*> export-policy rule show -vserver hadoopsvm -policyname default -instance

          Vserver: hadoopsvm
          Policy Name: default
          Rule Index: 1
          Access Protocol: cifs, nfs, flexcache
List of Client Match Hostnames, IP Addresses, Netgroups, or Domains: 0.0.0.0/0
          RO Access Rule: any
          RW Access Rule: any
User ID To Which Anonymous Users Are Mapped: 65534
          Superuser Security Types: any
          Honor SetUID Bits in SETATTR: true
          Allow Creation of Devices: true
          NTFS Unix Security Options: fail
Vserver NTFS Unix Security Options: use_export_policy
          Change Ownership Mode: restricted
Vserver Change Ownership Mode: use_export_policy

stlaff300-1and2::*>

stlaff300-1and2::*> export-policy rule modify -vserver hadoopsvm -policyname default -superuser
sys -ruleindex *
1 entry was modified.
```



```
stlaff300-1and2::~*> export-policy rule show -vserver hadoopsvm -policyname default -fields
superuser
vserver    policyname ruleindex superuser
-----
hadoopsvm default      1          sys

stlaff300-1and2::~*>
```

## 14.4 Split Metadata Size Exceeds 10,000,000

If the split metadata size exceeds 10,000,000, then change  
`mapreduce.job.split.metainfo.maxsize=-1` in XML or change it in the CLI by using  
`-Dmapreduce.job.split.metainfo.maxsize=-1`.

## 14.5 java.lang.OutOfMemoryError: Java Heap Space

See the following example of the `java.lang.OutOfMemoryError` message:

```
java.lang.OutOfMemoryError: Java heap space
18/04/24 16:53:38 ERROR rpc.RpcClientHandler: NfsConnectorV3.1.0 RPC: Got an exception
java.lang.OutOfMemoryError: GC overhead limit exceeded
Apr 24, 2018 4:53:38 PM org.jboss.netty.channel.DefaultChannelPipeline
WARNING: An exception was thrown by a user handler while handling an exception event ([id:
0x01029bf6, /192.168.30.175:54688 => /192.168.30.111:2049] EXCEPTION: java.lang.OutOfMemoryError:
Java heap space)
java.lang.OutOfMemoryError: GC overhead limit exceeded

Apr 24, 2018 4:53:38 PM org.jboss.netty.channel.socket.nio.AbstractNioSelector
WARNING: Unexpected exception in the selector loop.
java.lang.OutOfMemoryError: GC overhead limit exceeded
```

### Solution

`JAVA_OPTIONS` is the standard environment variable that some servers and other Java apps append to the call that executes the `JAVA` command. For example, allocate 240GB for heap size with the `JAVA` command and export `_JAVA_OPTIONS` with the following settings:

```
[root@hdp1 ~]# export _JAVA_OPTIONS=-Xmx240000M
[root@hdp1 ~]# echo $_JAVA_OPTIONS
-Xmx240000M
```

## 14.6 Temp Space

If temp space in the NodeManager is not equal to or greater than the dataset size, then run the `yarn.nodemanager.localizer.cache.target-size-mb` command to control the space.

## 14.7 Root User

See the following example of the root user error:

```
permission denied
```

### Solution

Change the root user in the SVM from 1 to 0 by running the `unix-user modify` command.

```
Hadoop_SVM:::> unix-user show -instance -user root

      User Name: root
      User ID: 0
Primary Group ID: 1
User's Full Name:
Hadoop_SVM:::> unix-user modify -user root -id 0 -primary-gid 0
```

```
Hadoop_SVM:> unix-user show -instance -user root

      User Name: root
      User ID: 0
Primary Group ID: 0
User's Full Name:

Hadoop_SVM:>
```

## 14.8 Class org.apache.hadoop.netapp.fs.nfs.NFSv3FileSystem Not Found

This error message sometimes displays in Hive:

```
FAILED: Execution Error, return code 1 from org.apache.hadoop.hive ql.exec.DDLTask.
MetaException(message:java.lang.RuntimeException:java.lang.ClassNotFoundException: Class
org.apache.hadoop.fs.nfs.NFSv3FileSystem not found
```

### Solution

If Hive cannot find the NetApp In-Place Analytics Module JAR file in the Hive prompt, then add the JAR files to Hive by using the `add jar` command.

```
hive> add jar /usr/hdp/2.5.3.51-3/hadoop/hadoop-nfs-2.7.3.2.5.3.51-3.jar;
Added [/usr/hdp/2.5.3.51-3/hadoop/hadoop-nfs-2.7.3.2.5.3.51-3.jar] to class path
Added resources: [/usr/hdp/2.5.3.51-3/hadoop/hadoop-nfs-2.7.3.2.5.3.51-3.jar]
Added [/usr/hdp/2.5.3.51-3/hadoop/hadoop-nfs-connector-2.0.0.jar] to class path
Added resources: [/usr/hdp/2.5.3.51-3/hadoop/hadoop-nfs-connector-2.0.0.jar]
```

## 14.9 Could Not Get Root File Handle for Endpoint

See this example of a could not get root file handle for endpoint message:

```
ERROR nfs.NFSv3FileSystemStore: NfsConnectorV3.1.0 Could not get root file handle for endpoint
ep=Endpoint: hosts=[nfs://10.63.150.118:2049/] export=/faiz_vol path=/
ls: Could not establish channel to any interface in vserver. Check network configuration
```

### Solution

- Check the `nfs` server export policy rules in the NetApp SVM that the Hadoop servers access.
- The new SVMs normally don't have the rules in the default export policy. Therefore, create a rule with `nfsv3` access, `unix` permissions.
- Check that the 2049 port is listening for the `nfs` server IP/LIF.
- Check an online website such as [jsonlint.com](https://jsonlint.com) to verify that the JSON syntax is correct.

## 14.10 No Such File or Directory

See the following example of a no such file or directory error:

```
mount.nfs: mounting 10.63.150.118:/faiz_vol1 failed, reason given by server: No such file or
directory
```

### Solution

In the NetApp SVM namespace configuration, make sure that `path` and `storage object` are the same.

## 15 Conclusion

The NetApp In-Place Analytics Module is easy to deploy. It runs analytics natively on existing NAS storage (NFS or CIFS) with high availability and only one copy of Hadoop data. The performance of this

software scales in proportion to the dataset size. In addition, it supports key Hadoop ecosystem projects such as Apache Hadoop, Apache Spark, Apache HBase.

## 16 Acknowledgments

We would like to thank the following people for their invaluable help in completing this document:

- Prasad Menon, QATS Manager, Partner Certification Team, Cloudera
- Nilesh Bagad, Senior Product Manager, NetApp
- Santosh Rao, Senior Technical Director, NetApp
- Lee Dorrier, Director, NetApp
- Aditya Sirna, Software Development Engineer in Test, Cloudera
- Manoj Kumar, Program Manager, Cloudera
- Harsh Shah, Partner Solution Engineer, Cloudera
- Archana Talamani, QATS Manager, Cloudera

## Where to Find Additional Information

To learn more about the information described in this document, refer to the following documents and websites:

- NetApp Solutions for Hadoop Reference Architecture  
[www.netapp.com/us/media/wp-7196.pdf](http://www.netapp.com/us/media/wp-7196.pdf)
- NetApp In-Place Analytics Module Best Practices  
<https://www.netapp.com/us/media/tr-4382.pdf>
- MixApart: Decoupled Analytics for Shared Storage Systems  
<https://www.usenix.org/system/files/conference/fast13/fast13-final58.pdf>
- Apache Hadoop YARN—Concepts and Applications  
<https://blog.cloudera.com/apache-hadoop-yarn-concepts-and-applications/>
- Apache HBase  
[https://en.wikipedia.org/wiki/Apache\\_HBase](https://en.wikipedia.org/wiki/Apache_HBase)

Refer to the [Interoperability Matrix Tool \(IMT\)](#) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

### **Copyright Information**

Copyright © 2019 NetApp, Inc. All rights reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

### **Trademark Information**

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.

TR-4715-0918