

# Hadoop 2.7.3 Drives Faster Results with Flash



## 9100PRO<sup>1</sup> HDFS Dynamic Tiering Compared to All-HDD and “Hot” Policy:

- 15% improvement with one-SSD policy
- 25% improvement with all-SSD policy

## Overview

The introduction of Apache Hadoop® 2.7.3 introduces several new storage policies. Designed to help optimize resources (separating hot and cold data), two of these new policies are particularly relevant when integrating SSDs<sup>2</sup>:

- One\_SSD – One replica is stored in SSD, remaining replicas are stored in DISK
- All\_SSD – All replicas are stored in SSD

These SSD-focused policies help manage where HDFS stores replicas of data currently in use.

The default storage policy keeps all active data on disk. This policy is called “Hot.”

Hot: The data that is popular and still being used for processing will stay in this policy. When a block is hot, all replicas are stored in DISK (HDDs). This is an all-HDD (baseline) policy

This technical brief shows how adding one 9100 PRO NVMe SSD to an existing HDD-based cluster and combining it with either of the SSD-related policies can reduce standard Hadoop benchmark run time.

We tested an all HDD and HDD with 9100 PRO configurations over the course of 10 benchmark runs. We compared the average completion time of each SSD-enabled configuration to the average completion time of the HDD baseline:

Policy	9100 Quantity	Improvement
Hot (HDD)	0	Baseline
One_SSD	1	15%
All_SSD	1	25%

*Table 1: Storage Policy and 9100 Benchmark Completion Time vs. Hot Policy and HDD*

<sup>1</sup> 9100 PRO U.2, 3.2TB tested. Other capacities and form factors also available.

<sup>2</sup> See <https://hadoop.apache.org/docs/r2.7.3/hadoop-project-dist/hadoop-hdfs/ArchivalStorage.html> for additional details.

## More Options

We can use the new storage policies in Apache Hadoop 2.7.3 to better control where data is stored. Designed to enable cluster capacity expansion without affecting computing capability, these policies enable files to be stored on different physical media.

Storage policies are assigned on a per-folder basis, Administrators can use Cron jobs to move data and replicas between directories and storage types (once the access times on the data has passed a certain point).

We can use these policies (in particular one\_SSD and all\_SSD) to enable dynamic data tiering for a more efficient separation of compute and storage resources. These policies (and the flexibility they provide) are of particular value when we need faster results, but have a limited budget (for example, a budget that only allows us to add one 9100 PRO per node).

## Faster Results

To gauge the performance benefits of using these two new storage policies with our 9100 PRO, we built and tested two different node configurations – one using only 5x 15K RPM SAS HDDs and a second that used the same HDD storage and added a single 9100 PRO U.2 3.2TB NVMe SSD. We used the HDD + 9100 PRO configuration to test standard benchmark completion time when we implemented the one\_SSD and all\_SSD policies. The table below shows the physical configurations tested and the storage policy we used with each:

Node Type	Server	HDDs	RAID Level	9100 PRO	Storage Policy
HDD Only	2U, 2 Socket Xeon E5-2690	5	RAID 0	0	hot
HDD with one 9100 PRO 3.2TB		5	RAID 0	1	one_SSD

Table 2: Node Configurations

We ran standard, built-in Hadoop benchmarks (Wordcount, Word Aggregate Histogram, Sort, Test-DFSIO-Write, Test-DFSIO-Read, Teragen, Terasort) 10 times and compared the average completion time across all 10 runs. We used the Hot policy on the all HDD configuration and the one\_SSD and all\_SSD policies on the configuration with the same HDD and one 9100 PRO. Figure 1 shows the results and the results are summarized in Table 3.

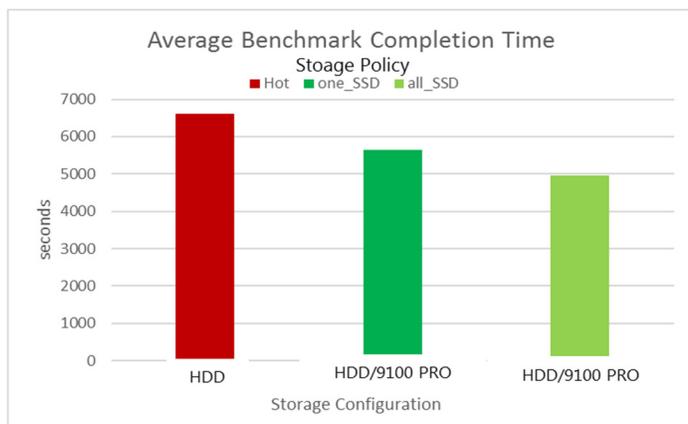


Figure 1: Average Benchmark Completion Time

Policy	Storage	Ave. Completion Time (seconds)	Improvement (vs baseline)
Hot	5x HDD	6616	Baseline
one_SSD	5x HDD / 1x	5642	15%
all_SSD	9100 PRO	4969	25%

Table 3: Summary of Average Benchmark Completion Time

Note that the master servers were virtualized in all tests. Virtualizing the master servers did not introduce any related performance limitations.

## Reduce Cluster Sprawl

For large, busy clusters reducing run time has real implications. If our cluster is busy and we can process a data set faster, we free up cluster resources sooner (because we complete processing sooner). This enables the cluster to begin working on the next data set.

For large, busy clusters reducing run time has real implications. If our cluster is busy and we can process a data set faster, we free up cluster resources sooner (because we complete processing sooner). This enables the cluster to begin working on the next data set.

We can relate faster processing time to reduced cluster size – if we have a fixed data set that we need processed and we can do it faster, we can also do it with fewer nodes.

For example, suppose we have a 100-node cluster that can process a given data set in 24 hours. If we improve that cluster's processing capability by 25% (runtime reduction), we would expect to process the same data set in 24 hours – with 25% fewer nodes. Although an approximation, we see how what may seem to be small reductions in run time may help us reduce the size of our cluster as well as related expenses.

## Summary

Apache Hadoop 2.7.3 offers us several new HDFS storage policies. Designed to help with optimization (by enabling us to store active data replicas in different locations), we found two of these policies very useful when combined with high performance SSDs like our 9100 PRO.

We tested two configurations (one with just HDDs and one with the same HDDs and a single 9100 PRO NVMe SSD). We measured standard Hadoop benchmark completion time (averaged over 10 runs) for the HDD configuration using the "hot" policy and compared that to the HDD + 9100 PRO using the "one\_SSD" and "all\_SSD" policies.

We saw a 15% to 25% improvement in benchmark completion time with the HDD + 9100 PRO configurations using the one\_SSD and all\_SSD policies respectively.

These new storage policies combined with the 9100 PRO give us more options. We can improve the capabilities of existing deployments by extending their value by a simple upgrade. We can also help reduce cluster sprawl for active clusters (by processing the same data set faster or reducing the size of the cluster).

## How We Measured These Results

We used the standard benchmarks and built it to most Hadoop distributions, running the set 10 times and recording the mean completion time. Table 4 shows the benchmarks we used and their parameters.

Benchmark	Parameters
Wordcount	mapreduce.randomtextwriter.totalbytes=2147483648000 mapreduce.job.maps=100 mapreduce.job.reduces=100 mapreduce.output.fileoutputformat.compress=false
Word Aggregate Histogram	Default (no parameters set)
Sort	-r 100
Test DFSIO Write	-write -nrFiles 1000 -size 1024MB
Test DFSIO-Read	-read -nrFiles 1000 -size 1024MB
Teragen	mapreduce.job.maps=100 mapreduce.job.reduces=100 21474836480 (bytes)
Terasoft	mapreduce.job.maps=100 mapreduce.job.reduces=100

*Table 4: Benchmark Parameters*

## micron.com

This technical marketing brief is published by Micron and has not been authorized, sponsored, or otherwise approved by <<company name>>. Products are warranted only to meet Micron's production data sheet specifications. Products and specifications are subject to change without notice.

©2016 Micron Technology, Inc. Micron and the Micron logo are trademarks of Micron Technology, Inc. <include third-party trademark statements here.>> All other trademarks are the property of their respective owners. All rights reserved.

Rev. A 3/17 CCMMD-676576390-10662