WHITEPAPER

A JOINT PUBLICATION BY INTEL AND TWITTER

Data Center
Hadoop Performance

Boosting Hadoop* Performance and Cost Efficiency with Caching, Fast SSDs, and More Compute

Through experimentation and collaboration, Twitter discovers that increasing the core density of its Hadoop* clusters by 6X would result in 30 percent lower TCO and up to 50 percent faster runtimes[1]

Table of Contents

Executive Overview

Storage I/O can be a significant performance bottleneck for Hadoop* clusters, especially in hyperscale deployments like those at Twitter, where a single cluster can have up to 10,000 nodes and nearly 100 PB of logical storage. The typical Hadoop cluster at Twitter contains over 100,000 hard disk drives (HDDs)—but this configuration was reaching an I/O performance limit because while HDD capacity has increased over time, HDD performance has not significantly changed.[2] Therefore, simply adding more, bigger HDDs wasn't going to solve Twitter's scaling challenges—in fact, it would make things worse as the I/O per GB decreases. Adding more spindles per node was not feasible due to space and power limitations.

Working in collaboration with an Intel engineering team, Twitter engineers conducted a series of experiments that revealed that storing temporary files managed by YARN* (Yet Another Resource Negotiator*) on a fast SSD enabled significant performance improvements on existing hardware (up to a 50 percent reduction in runtime).[3] The team also discovered that removing a storage I/O bottleneck enabled them to use larger hard drives while simultaneously increasing processor utilization, which in turn resulted in the ability to use higher-core-count processors. This positively affected storage performance, and contributed to higher data center density by reducing the number of required HDDs.

Higher density leads to total cost of ownership (TCO) savings through energy efficiency, fewer racks, and a smaller data center footprint. Overall, Twitter expects that caching temporary data and increasing core counts will result in approximately 30 percent lower TCO and over 50 percent faster runtimes, compared to their legacy production cluster configuration.[1]

Business Challenge

Twitter uses Hadoop* for storing data and performing advanced analytics to generate important business insights. As one of the largest Hadoop users in the world, Twitter's Hadoop clusters comprise half a million compute threads and more than 300 PB of logical storage total (30 PB logical storage or more per cluster), which results in an exabyte of physical storage due to replication. Peak cluster size can exceed 10,000 nodes, and Twitter processes over 1 trillion events per day.

Authors

Dave Beckett
Staff Site Reliability Engineer,
Twitter, Inc.

Matt Singer
Sr. Staff Hardware Engineer, Twitter, Inc.

Milind Damle,
Sr. Engineering Director,
Big Data Solutions and Performance Engineering, Intel Corp.

Rakesh Radhakrishnan
Sr. Software Engineer, Hadoop HDFS Expert, Hadoop PMC Member,
Intel Corp.

Barrie Wheeler
Sr. Application Engineer,
Intel® Cache Acceleration Software Expert, Intel Corp.

Contributors

Varun Sampat
Sr. Hardware Engineer, Twitter, Inc.

Mark Schonbach
Sr. Site Reliability Engineer, Twitter, Inc.

Ali Alavi
Industry Technical Specialist for Twitter, Cloud Service Providers, Intel Corp.

Mauricio Cuervo
Sr. Account Executive for Twitter, Project Lead, Cloud Service Providers, Intel Corp.

Juan Fernandez
NSG Technical Solutions Specialist, Intel Corp.

Uma Gangumalla
Sr. Software Engineer, Hadoop HDFS Expert, HDFS PCM Member, Intel Corp.

Devaraj Kavali
Sr. Software Engineer, Hadoop YARN Expert, Intel Corp.

David Leone
Application Engineering Manager, Attached Platform Storage Software Division Lead, Intel Corp.

Brien Porter
Sr. Program Manager, Open Source Tech Lead, Intel Corp.

Michal Wysoczanski
Sr. Software Architect, Intel® Cache Acceleration Software Lead, Intel Corp.

Figure 1 shows the data flow in a typical Hadoop cluster at Twitter. The Hadoop Distributed File System* (HDFS*) produces approximately one data flow per HDD, while Map-Reduce processing (managed by YARN) results in multiple data flows for the purpose of storing temporary data. Each of these temporary data flows is targeted to a different HDD, overlapping with the HDFS data flows.

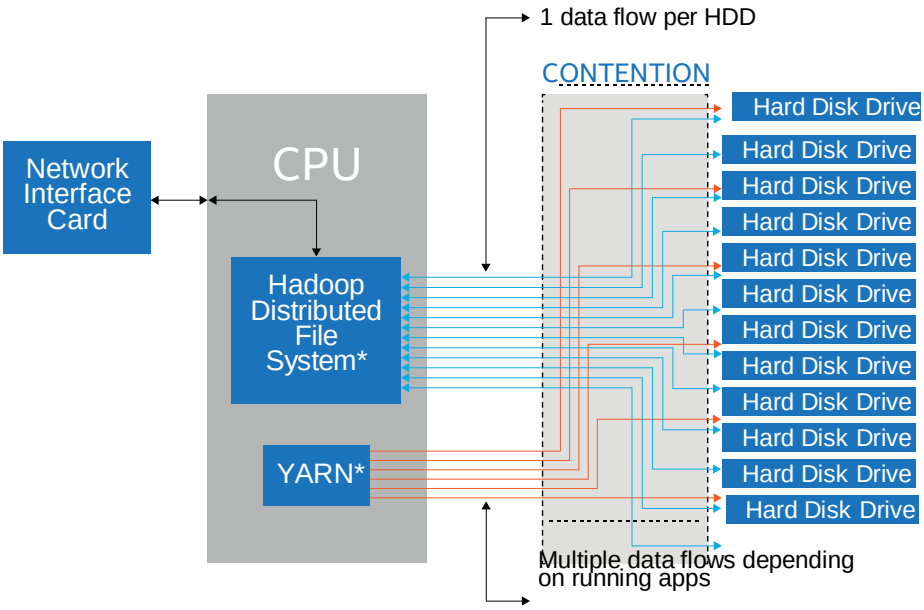


Figure 1 . Typical data flow in a Hadoop* cluster results in HDFS* data and temporary data managed by YARN* contending for the HDDs.

Because of their affordable cost per GB, 7200RPM HDDs are the workhorses of Twitter's Hadoop clusters. And, up until recently, simply adding more spindles as storage needs expanded seemed the best solution. But eventually, the number of HDDs reached critical mass—HDD capacities have increased over time, but their throughput and I/O per second (IOPS) have remained relatively stagnant. As a result, the number of IOPS per GB of storage was limited and constrained potential architectural and hardware choices; having to add more servers to the cluster was driving up costs. The question was, what could be done to boost I/O performance without increasing cost significantly? The Twitter engineers set out to investigate. What they learned dispelled some long-held assumptions.

When Guessing Wrong Leads to Amazing Insights

Firm believers in "you can't improve something if you can't measure it," the Twitter team decided to measure I/O and CPU usage in a test cluster, using a combination of the following:

- A synthetic benchmark (Terasort*)
- A replay of highly representative production workloads (using Gridmix*)
- A system profiler (Intel® VTune™ Amplifier - Platform Profiler)

Twitter's test cluster used dual-socket Intel® Xeon® E5-2630 processors v4, which provide 10 cores/20 threads per socket.[4] The cluster consisted of 102 nodes spread across six racks with 25 GbE connectivity. In parallel, Intel set up a smaller lab (only nine nodes). The investigative journey wasn't a straight line to success; yet the collaboration and experimentation between the two teams revealed some rather surprising insights into the inner workings of Hadoop I/O (see Figure 2).

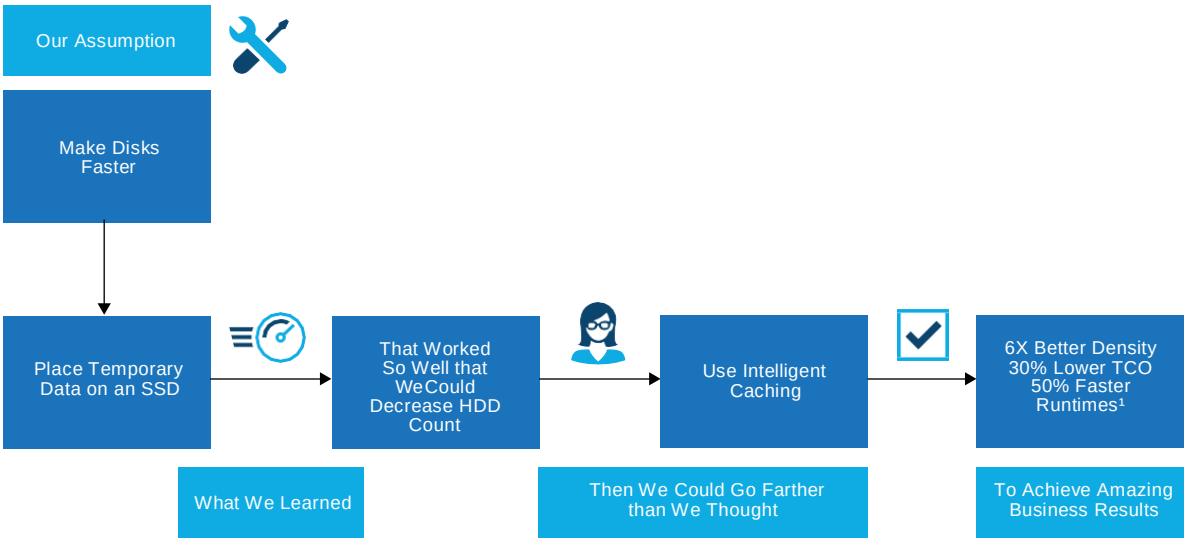


Figure 2 . The Twitter team's investigative journey began with one goal, but through a process of experimentation led to unexpected business benefits.

Caching Everything Doesn't Really Help

At the outset, Twitter engineers assumed that it was the sheer mass of data that was causing the Hadoop slowdown. After discussion with Intel engineers, the Twitter team decided to explore caching the entire disk subsystem, using an Intel® Optane™ Solid State Drive (SSD) and Intel® Cache Acceleration Software (Intel® CAS).

HDFS is designed for large bulk reads and writes of large data files. Twitter typically uses 512 MB block sizes for HDFS, so in most applications it reads or writes 0.5 GB of data at once. The worker nodes where HDFS runs have 12 drives and each drive reads or writes bulk data for HDFS use. Twitter engineers rely on the aggregate IOPS of all the drives to ensure the aggregate performance remains high. But as disks became large, the IOPS per gigabyte decreased. The first hypothesis was that fast caching might help alleviate this by smoothing out the I/O, even though the read and write sizes were large compared to typical cache drive sizes.

After putting this hypothesis to the test, the Twitter team found that caching didn't help. Why? The performance benefits of caching generally apply to data that needs to be accessed multiple times. But for Twitter's workloads, the HDFS data is written to disk once and generally not consumed again for quite some time. As a result, not only was there no performance improvement, but tests showed a small degradation of I/O performance (sometimes this effect is called "polluting the cache"). What followed was more discussion and a second hypothesis: what if the temporary data (see the sidebar, "An Introduction to Apache Hadoop* and YARN*") was stored on an SSD?

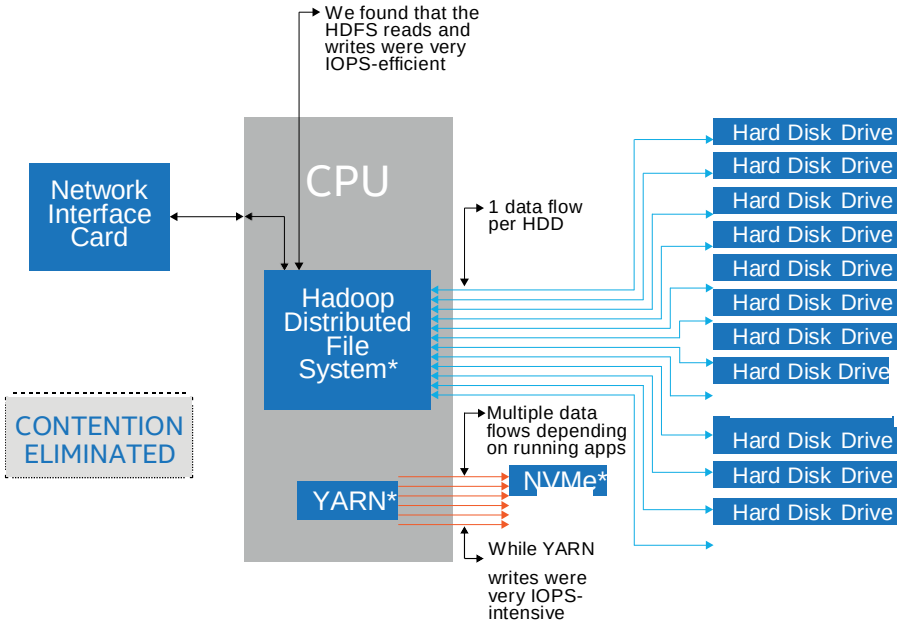


Figure 3 . When using an NVMe*-based SSD to store temporary data managed by YARN*, the contention for the HDDs is eliminated.

Placing Temporary Data on an SSD

After discussing the second hypothesis with Intel engineers, the Twitter team decided to explore selectively placing the temporary data contained in the YARN Temp directory (see Figure 3).

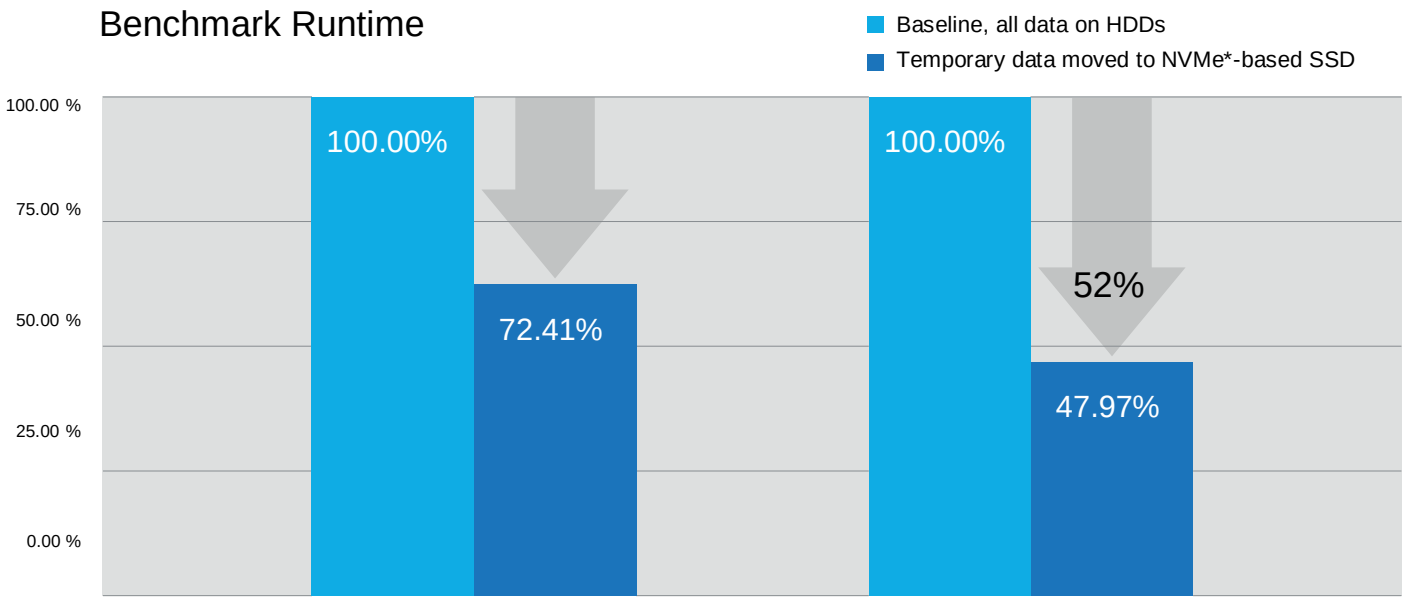


Figure 4 . Storing temporary data on an NVMe*-based SSD resulted in significant reduction in benchmark runtimes.[5]

The results were relatively astounding. It turns out the HDFS data reads and writes were very IOPS-efficient, while the temporary data was much more I/O intensive than originally thought. As shown in Figure 4, simply adding one Intel® Optane™ SSD DC P4800X to each host in the test cluster to store the temporary data resulted in a 27.5 percent runtime reduction for Gridmix, and a 52 percent runtime reduction for Terasort.[5]

The runtime reductions were possible because MapReduce* temporary files and the Hadoop Distributed File System* (HDFS*) were not contending for the same disk. Therefore, HDD utilization dropped, and Hadoop could serve up data faster. The application profiler revealed that without the SSD, the HDD was moving an average of 37 MB/sec; with the SSD, the average HDD traffic dropped to about 6 MB/second. That is a small fraction of the approximately 200 MB/second rated capability of the HDDs in use at Twitter. During the Gridmix test, the profiler also indicated that CPU utilization increased from an average of 40 percent to an average of 57 percent. That is, the CPU was doing work at 1.4X the original rate, which correlates with the reduced runtimes.

An interesting development was that the Intel lab tests showed a 51.7 percent runtime reduction for Gridmix—nearly twice the results obtained in the Twitter lab.[6] When comparing notes, it became clear that Intel's lab configuration of 112 threads using Intel® Xeon® Platinum 8180 processors with eight HDDs was very different than Twitter's 40 threads and 12 HDDs. Intel's lab system had many more threads per HDD (14) compared to Twitter's lab (3.33). By removing the storage bottleneck using an NVMe-based SSD, the I/O tasks became more compute-bound; therefore, Intel's higher-core-count test cluster could scale more effectively. (See "Doing More with Less" for a detailed discussion of the relationship between core count and I/O performance.)

An important takeaway from Twitter's testing is that it isn't enough just to focus on low-level benchmarks, but rather on application benchmarks. Low-level read and write performance numbers are not a great indicator of performance limits. Real-life workloads are a mixture of read, write and compute, and there are different types of reads and writes. Twitter's Hadoop workload was a mix of reads and writes of 512 MB files and relatively small files. This mixture would be difficult to optimize without splitting the workload.

Doing More with Less

The significant drop in HDD utilization (almost 84 percent) when the SSD was added led the Twitter team to explore a counter-intuitive idea: Could they reduce the number of HDDs in a node? To answer that question, they retested the cluster using nodes with the baseline twelve (12) HDDs, then with six (6) HDDs, and then with only three (3) HDDs per node. Figure 5 shows the astonishing results: it was possible to reduce the number of HDDs by 75 percent without increasing the Gridmix runtime.[7] Without the SSD for storing temporary data, the Gridmix runtime increased significantly as HDDs were removed from the nodes—with only three HDDs and no caching, the benchmark took 231 percent longer to run. But with the SSD, the runtime remained virtually unchanged as HDDs were removed.

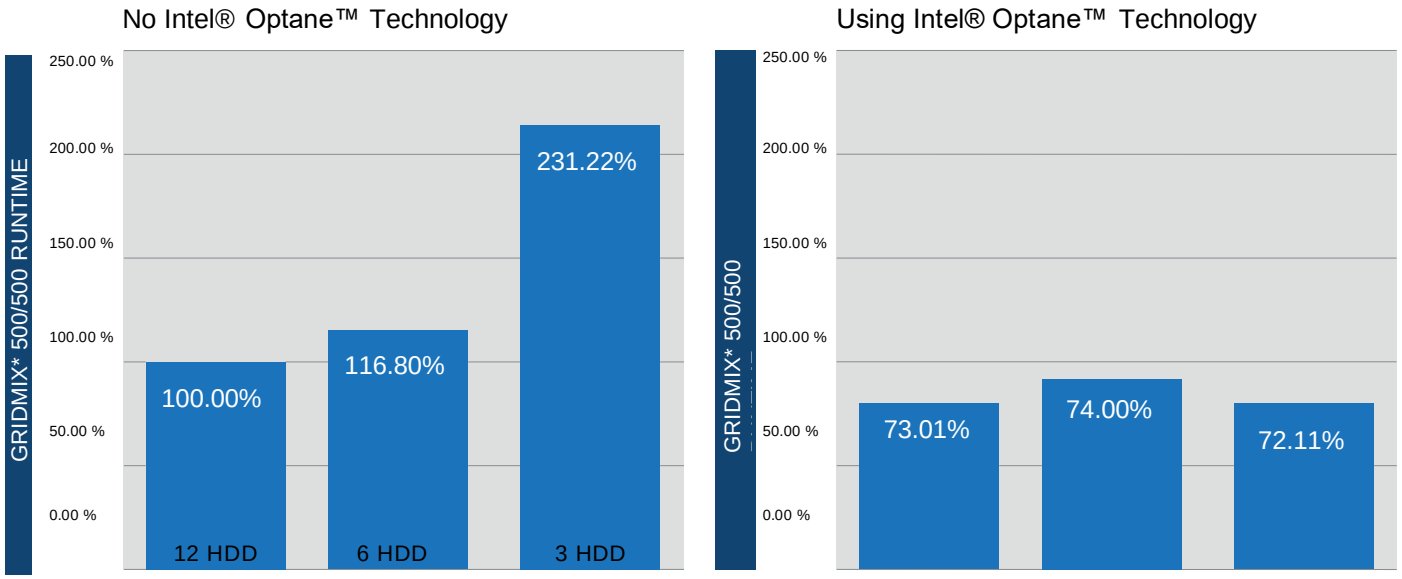


Figure 5 . Temporary data stored with Intel® Optane™ technology enabled use of fewer HDDs without affecting benchmark runtime.[7]

To further explore the relationship between core count and I/O performance, the Twitter team experimented with various configurations. They used the EMON tool, part of the Intel VTune Amplifier - Platform Profiler, to simulate CPU scaling from a baseline 10-core/20-thread system to a 20-core/40-thread system.[8] Some of the results of those tests are shown in Figure 6. With more compute power combined with SSD storing of temporary data, it was possible to reduce the number of HDDs by 75 percent and reduce runtime by about 40 percent compared to the baseline of 12 HDDs with a lower thread count. In other words, after optimizing the storage subsystem Twitter's storage clusters would actually be CPU-bound, not IOPS-bound. Once the SSD was added for temporary data, huge CPU-driven scaling possibilities were revealed.

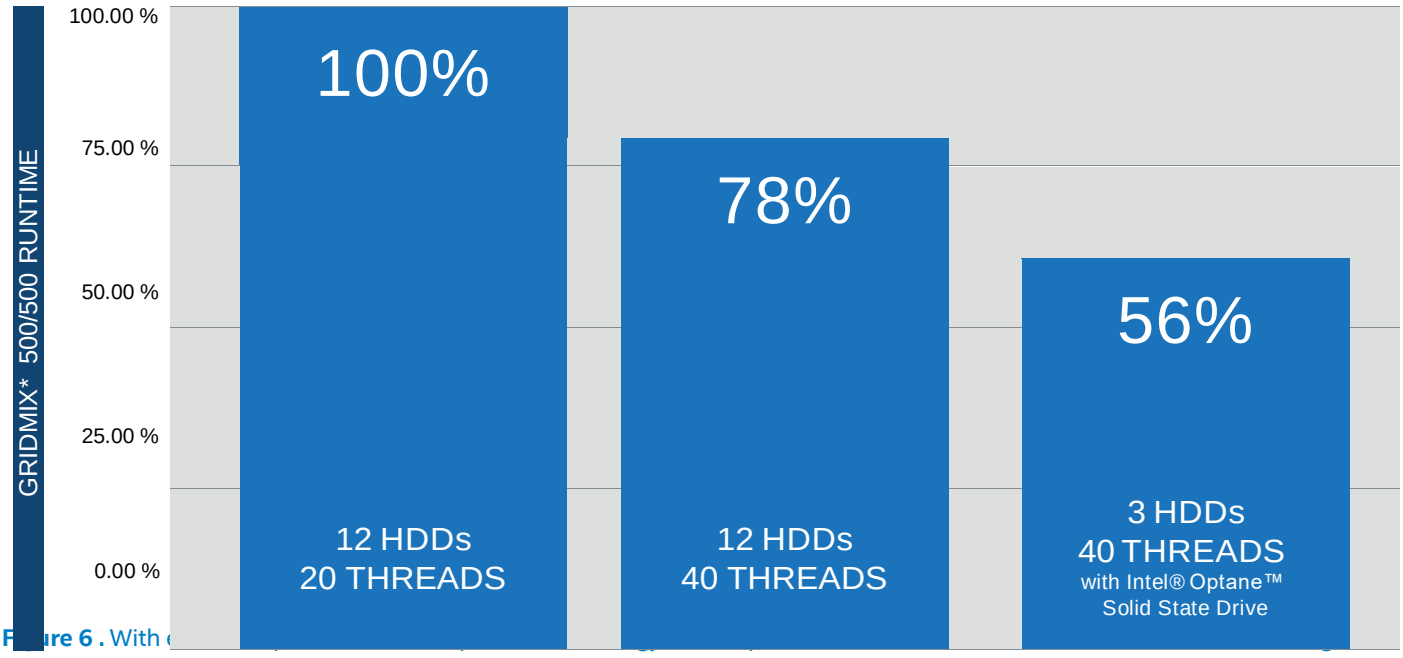


Figure 6 . With [illegible] benchmark runtime by about 40 percent.[8]

The team thoroughly analyzed the test results data and consulted with Intel Software and Services (SSG) engineers. By extrapolation from all the data, the team determined that a 24-core system, based on the Intel® Xeon® Gold 6262 processor, would be optimal for Twitter's clusters.

It is important to note that the low power consumption of this 2nd Generation Intel® Xeon® Scalable processor was critical to the deployment decision, because it provides a high core count at a great thermal design power (TDP).

Density Is the Driver of Savings

The business benefits of intelligent fast SSD caching of temporary data, combined with increased CPU power, were even more than the Twitter team imagined at first. Looking at early results, they thought maybe to double or maybe even triple the cluster density. But with the planned configuration (see Table 1), they actually expect to achieve 6X more compute density compared to the legacy cluster configuration.

Densification leads to savings through fewer HDDs and fewer servers (CapEx avoidance) as well as through reduced drive maintenance and power and cooling costs (OpEx savings) and a smaller data center footprint. For example, with the planned configuration, Twitter's clusters will go from over 100,000 HDDs to only 20,000 HDDs. That means far fewer fans, power supplies, and other moving parts that can—and do—fail. Additionally, Twitter expects that the reduction in HDD count will translate directly to lower operation burden due to HDD failures.

After the evaluation, Twitter looked into other aspects of the configuration and future needs, to arrive at a planned configuration decision. After determining the appropriate CPU scaling, and anticipating more compute-heavy workloads with consequently more YARN temporary space requirements, Twitter determined the following using telemetry data:

- The thread density needed to be increased.
- Consequently, the space for temporary data needed to be increased.
- The caching SSD needed to be at least 6.4 TB per node (95th percentile of maximum amount of data seen).[9]
- The number of HDDs should not be reduced by the maximum indicated by test results, to allow for IOPS and bandwidth headroom.

The Twitter team is planning to deploy five racks equipped with Intel Xeon Gold processors 6262V to fully validate the new hardware on a production load. Overall, Twitter expects to achieve a 30 percent TCO reduction with the new planned configuration.

Table 1 . Planned Configuration for Densification and Performance Improvement

	Legacy Configuration	Planned Configuration
Processor	Intel® Xeon® E3-1230 processor v6 (single socket, 4 cores)	Intel® Xeon® Gold 6262V processor (single socket, 24 cores)
Memory	32 to 64 GB	192 GB
Hard Disk Drive (HDD)	12x 1 or 2 TB HDDs	8x 6 TB HDDs
Boot Disk	Intel® S4500 240 GB	Intel® S4510 240 GB
Caching SSD (YARN* storage, temporary data)	N/A	1x Intel® SSD DC P4610 6.4 TB (High-Performance NVMe*-based SSD)
Compute	1X	6X
Storage	1X	3X to 6X per node
Rack Reduction Factor	1X	4X
Compute Scaling	1X	6X
Caching Software	N/A	Intel® Cache Acceleration Software (Intel® CAS)
Network	1 GB to 10 GB	25 GB

Collaborating to Achieve Intelligent Caching

In Twitter's initial tests with the NVMe-based SSD, the temporary data was sent directly to the SSD (with no caching), because the original analysis of their workload indicated that all of the temporary data would fit on a 6.4 TB SSD. But after more research, the Twitter engineers discovered that with the continuous growth of data flowing through Twitter's clusters, analytics workloads are expected to grow as well. As a result, the temporary data would grow to far exceed 6.4 TB (up to 12 TB or more). So, the Twitter team collaborated with Intel CAS engineers to explore an existing capability within Intel CAS to smoothly flush data to another drive when a cache device becomes full. This prevents any cases where the dedicated YARN NVMe-based SSD runs out of space, which would cause job failures.

The temporary data managed by YARN is typically processed into Hadoop-configured directories, so the Twitter team requested that Intel CAS support directory-specific caching. The Intel team modified Intel CAS and performed intensive performance testing to successfully meet the request. This feature ensures that all temporary data can be isolated by directory and moved to the cache device. In this way, the temporary data receives the full benefit of caching, plus takes advantage of the capability of Intel CAS to protect data when the cache device becomes full, at which time the temporary data spills to an HDD. Most jobs produce a small amount of temporary data and can rely solely on the capacity of the SSD, but large jobs that spill over still get the maximum benefit of caching. The maximum size of the temporary data is thus not limited by the size of the cache SSD. However, to maximize performance, the drive needs to be larger than the temporary data most of the time.

An Introduction to Apache Hadoop* and YARN*

The Apache Hadoop* software system is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. The system is designed to detect and handle failures at the application layer, to deliver a highly-available service on top of a cluster of computers, each of which may be prone to failures. YARN* is the resource manager and job scheduler for Apache Hadoop*. Essentially, you can think of YARN as an abstraction of resource management in Hadoop V1, allowing the use of different compute frameworks beyond MapReduce*, such as Spark*.

In a cluster architecture, YARN is a software layer that resides between the Hadoop Distributed File System* (HDFS*) and the processing engines that run applications. Applications using frameworks on top of YARN (such as MapReduce) create temporary files, such as map outputs, when running jobs. The applications write this temporary data to disks as jobs run, and then clear the temporary data as jobs complete. Generally speaking, each temporary data file is fairly small. The combination of small file sizes and repeated access makes this temporary data a natural fit for caching to a separate drive (rather than writing it to the main hard disk drives (HDDs) being used by HDFS). In addition, caching reduces contention for the HDDs.

Best Practices for Transforming Your Hadoop Clusters

A few things the Twitter team learned along the way:

- Be prepared to challenge long-held assumptions; this frees you up to make unexpected discoveries. For example, the Twitter team never anticipated that there was a way to remove 75 percent of the HDDs from the system without harming performance.
- Use a well-defined process for efficient experimentation: measure, experiment, learn, and repeat.
- For measuring, an advanced profiler with good visualization tools makes it easy to see what is really happening in both test and production clusters.
- Be sure to measure more than just low-level read and write performance. It is important to understand your particular workload—which may be quite different than Twitter's. In particular, develop an understanding of what types of reads and writes are occurring. That will guide your optimization efforts.
- Collaborate with other experts who can give you new ideas. For example, the Intel team shared their test results, helped explain why certain things happened and helped Twitter meet or exceed their scaling and energy-efficiency goals when planning the new cluster configuration.

Next Steps

Experimentation is never really done; you can always learn more and improve. Going forward, the Twitter team intends to conduct additional experiments to explore the following:

- Optimal cache capacity on NVMe-based SSDs
- SSD endurance needs
- Optimal balance of HDDs, threads and NVMe*-based SSDs

Conclusion

Figure 7 consolidates the Twitter team's key learnings. After moving the temporary data from MapReduce processes to a fast SSD, it became clear that fewer HDDs were required. And, more compute threads per disk can further enhance performance. The entire discovery process was the result of fruitful collaboration between Twitter and Intel engineers, seeking a solution that solved Twitter's challenges. For example, the Intel CAS directory-specific caching capability was a direct result of this collaboration. Twitter and Intel will continue to work together, sharing learnings and further optimizing Twitter's Hadoop clusters.

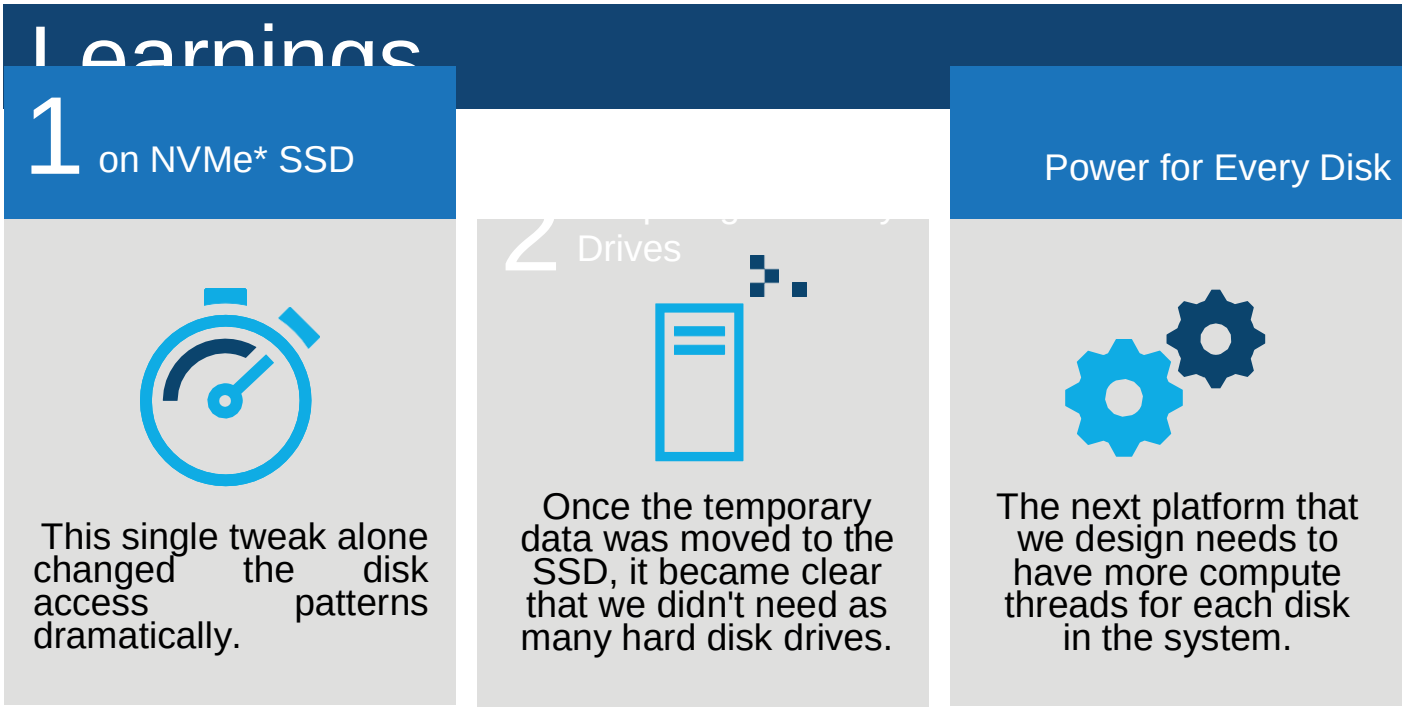


Figure 7 . Key learnings from Twitter's Hadoop* cluster optimization efforts.

Learn More

You may find the following resources useful:

- Intel® Resources for Cloud Service Providers
- Intel® SSD DC D7 Series
- Intel® Optane™ SSD DC P4800X Series
- Intel® Cache Acceleration Software (Intel® CAS)

[1] Baseline: Single-socket Intel® Xeon® processor E3-1230 v6 (4 cores); 32 to 64 GB RAM; 1x 1 TB or 2 TB HDDs; Intel® S4500 240 GB boot disk; 1 GbE to 10 GbE Ethernet; no caching.
Test: Single-socket Intel Xeon Gold 6262 processor (24 cores); 192 GB RAM; Intel S4500 240 GB boot disk; 8x 6 TB HDDs; 1x Intel® SSD DC P4610 6.4TB; 25 GbE Ethernet; caching using Intel® Cache Acceleration Software.

OS: Twitter CentOS* 6 Derivative, Kernel Version 2.6.74-t1.el6.x86_64 (based on upstream 4.14.12 Kernel) , BIOS Version: D3WWM11, Microcode Version: 0xb000021

[2] Backblaze, September 2018, "Hard Disk Drive (HDD) vs Solid State Drive (SSD): What's the Diff?" https://www.backblaze.com/blog/hdd-versus-ssd-whats-the-diff/

[3] Baseline: Dual-socket Intel® Xeon® E5-2630 v4 @ 2.2 GHz (10 cores/20 threads per socket); 128 GB RAM; 12x 6 TB 7200 RPM SATA HDD; 1x SATA SSD boot disk; 25 GbE Ethernet; 102 nodes spread across 6 racks. Workload: Gridmix* and Terasort*. Gridmix Score: 3309 seconds; Terasort Score: 5504 seconds
Test: Dual-socket Intel® Xeon® E5-2630 v4 @ 2.2 GHz (10 cores/20 threads per socket); 128 GB RAM; 12x 6 TB 7200 RPM SATA HDD; 1x SATA SSD boot disk; 1x 750 GB Intel® Optane™ DC P4800X NVMe*-based SSD; 25 GbE Ethernet; 102 nodes spread across 6 racks. Workload: Gridmix and Terasort. Gridmix Score: 2396 seconds; Terasort Score: 2640 seconds

OS: Twitter CentOS* 6 Derivative, Kernel Version 2.6.74-t1.el6.x86_64 (based on upstream 4.14.12 Kernel) , BIOS Version: D3WWM11, Microcode Version: 0xb000021

[4] Note that the test cluster used a higher core count than Twitter's production Hadoop* clusters, which provided only 4 cores/8 threads per HDD.

[5] Baseline: Dual-socket Intel® Xeon® E5-2630 v4 @ 2.2 GHz (10 cores/20 threads per socket); 128 GB RAM; 12x 6 TB 7200 RPM SATA HDD; 1x SATA SSD boot disk; 25 GbE Ethernet; 102 nodes spread across 6 racks. Workload: Gridmix* and Terasort*. Gridmix Score: 3309 seconds; Terasort Score: 5504 seconds
Test: Dual-socket Intel Xeon E5-2630 v4 @ 2.2 GHz (10 cores/20 threads per socket); 128 GB RAM; 12x 6 TB 7200 RPM SATA HDD; 1x SATA SSD boot disk; 1x 750 GB Intel® Optane™ DC P4800X NVMe*-based SSD; 25 GbE Ethernet; 102 nodes spread across 6 racks. Workload: Gridmix and Terasort. Gridmix Score: 2396 seconds; Terasort Score: 2640 seconds

OS: Twitter CentOS* 6 Derivative, Kernel Version 2.6.74-t1.el6.x86_64 (based on upstream 4.14.12 Kernel) , BIOS Version: D3WWM11, Microcode Version: 0xb000021

[6] Testing by Intel.
Baseline: 1x Name Node (2x Intel® Xeon® E5-2699 v4 @2.20 GHz, 128GB DDR4-2666 ECC, Intel® SSD DC S4600 for boot drive 240 GB, 2x Intel® Ethernet Controller 10-Gigabit X540-AT2 rev 01); 9x Data Node (2x Intel Xeon Platinum 8180 Processor @ 2.5 GHz, 128 GB DDR4-2666 ECC, Intel SSD DC S4600 for boot drive 240 GB, 4x Intel® Ethernet Controller X710/X557-AT 10GBASE-T rev 02, 8x HDD Seagate 7200RPM SATA ST4000NM0085. Gridmix* Score: 5592 seconds
Test: 1x Name Node (2x Intel® Xeon® E5-2699 v4 @2.20 GHz, 128GB DDR4-2666 ECC, Intel® SSD DC S4600 for boot drive 240 GB, 2x Intel® Ethernet Controller 10-Gigabit X540-AT2 rev 01); 9x Data Node (2x Intel Xeon Platinum 8180 Processor @ 2.5 GHz, 128 GB DDR4-2666 ECC, Intel SSD DC S4600 for boot drive 240 GB, 4x Intel® Ethernet Controller X710/X557-AT 10GBASE-T rev 02, 8x HDD Seagate 7200RPM SATA ST4000NM0085, 1x NVMe*-based Intel® P4600 1.6 TB SSD and 1x NVMe-based Intel(r) Optane(tm) P4800X 750 GB SSD for temporary data). Gridmix Score: 2702 seconds
Software: OS: Twitter CentOS* 6 Derivative, Kernel Version 2.6.74-t1.el6.x86_64 (based on 4.14.12 Kernel), Application: Apache Hadoop* 2.9 Replication Factor 3, Network Interface Bonding: 2x10 Gbps interfaces bonded 20 Gbps Mode 4 LACP, Intel® Cache Acceleration Software v3.9 (YARN* directories and metadata cached), Supermicro* X11DPU BIOS Rev:2.0a, Microcode version: 0x200003a

[7] Baseline: Dual-socket Intel® Xeon® E5-2630 v4 @ 2.2 GHz (10 cores/20 threads per socket); 128 GB RAM; 12x, 6x and 3x 6 TB 7200 RPM SATA HDD; 1x SATA SSD boot disk; 25 GbE Ethernet; 102 nodes spread across 6 racks. Workload: Gridmix*. Gridmix Score (12 HDDs): 3309 seconds, Gridmix Score (6 HDDs): 3865 seconds, Gridmix Score (3 HDDs): 7651 seconds
Test: Dual-socket Intel Xeon E5-2630 v4 @ 2.2 GHz (10 cores/20 threads per socket); 128 GB RAM; 12x, 6x and 3x 6 TB 7200 RPM SATA HDD; 1x SATA SSD boot disk; 1x 750 GB Intel® Optane™ DC P4800X NVMe*-based SSD; 25 GbE Ethernet; 102 nodes spread across 6 racks. Workload: Gridmix*. Gridmix Score (12 HDDs): 2416 seconds, Gridmix Score (6 HDDs): 2448.5 seconds, Gridmix Score (3 HDDs): 2386 seconds

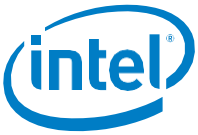
OS: Twitter CentOS* 6 Derivative, Kernel Version 2.6.74-t1.el6.x86_64 (based on upstream 4.14.12 Kernel) , BIOS Version: D3WWM11, Microcode Version: 0xb000021

[8] Baseline (12 HDDs, 20 threads): Dual-socket Intel® Xeon® E5-2630 v4 @ 2.2 GHz (10 cores/20 threads per socket, but with half the cores turned off), 128 GB RAM, 12x 6 TB 7200 RPM SATA HDD, 1x SATA SSD boot disk, 25 GbE Ethernet; 102 nodes spread across 6 racks. Workload: Gridmix*. Gridmix Score: 4227 seconds
Test (12 HDDs, 40 threads): Dual-socket Intel Xeon E5-2630 v4 @ 2.2 GHz (10 cores/20 threads per socket, all cores active), 128 GB RAM, 12x 6 TB 7200 RPM SATA HDD, 1x SATA SSD boot disk, 25 GbE Ethernet; 102 nodes spread across 6 racks. Workload: Gridmix*. Gridmix Score: 3309 seconds
Test (3 HDDs, 40 threads with NVMe*-based caching): Dual-socket Intel Xeon E5-2630 v4 @ 2.2 GHz (10 cores/20 threads per socket, all cores active), 128 GB RAM, 3x 6 TB 7200 RPM SATA HDD, 1x SATA SSD boot disk, 1x 750 GB Intel® Optane™ DC P4800X NVMe*-based SSD, 25 GbE Ethernet; 102 nodes spread across 6 racks. Workload: Gridmix*. Gridmix Score: 2386 seconds

OS: Twitter CentOS* 6 Derivative, Kernel Version 2.6.74-t1.el6.x86_64 (based on upstream 4.14.12 Kernel) , BIOS Version: D3WWM11, Microcode Version: 0xb000021

[9] 6.4 TB is far larger than the biggest available Intel® Optane™ DC SSD (1.5 TB). Therefore, although the tests used an Intel Optane DC SSD for caching the temporary data, for its planned production configuration Twitter chose the Intel® SSD DC P4610, as it provided the right balance of NVMe*-based high performance and the high capacity that Twitter required.

Solution Provided By:



Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors.

Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit www.intel.com/benchmarks.

Testing by Twitter. See configuration disclosures for details.

Performance results are based on testing as of September 26, 2018 and may not reflect all publicly available security updates. See configuration disclosure for details. No component or product can be absolutely secure.

Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Intel does not control or audit third-party data. You should review this content, consult other sources, and confirm whether referenced data are accurate.

Cost reduction scenarios described are intended as examples of how a given Intel- based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

0319/CAT/RW/PDF 338949-001EN