White Paper

MapR Technologies

MapR — The Industry's Most Dependable Hadoop Platform

The MapR Distribution for Apache Hadoop

Easy, Dependable & Fast Hadoop

Table of Contents

Executive Summary

With the Internet now touching two billion people daily, every call, tweet, e-mail, download, or purchase generates valuable data. Companies are increasingly relying on Hadoop to unlock the hidden value of this rapidly expanding data, and to drive increased growth and profitability. Orbitz has 4.6 million unique site visitors per month. Facebook grew from 400 million users to 500 million users in less than six months. Zynga recently served up 7.5 million virtual Valentine's Day cakes. Each of these companies relies on Hadoop to process massive amounts of data and improve their business.

Hadoop uses are also not limited to analyzing clickstreams. Sensor output, videos, log files, location data, genomics, behavioral data, and seismic studies are just a few of the data sources that are driving Hadoop use across government agencies and business verticals. At the same time, experienced users understand the challenges and limitations presented by Hadoop. While there is currently a choice of six different Hadoop distributions, they all share the same configuration issues, single points of failure, data loss risks, and performance limitations.

This paper provides details on a significant new alternative — the MapR Distribution for Apache Hadoop — the easiest, most dependable, and fastest Hadoop distribution.

Strategic Hadoop

In evaluating and selecting a Hadoop distribution, organizations need to choose criteria that mean the most to their business or activity. Key questions related to the distribution include:

- **How easy is it to use?**
  How easily does data move into and out of the cluster?
  Can the cluster be easily shared across users, workloads, and geographies?
  Can the cluster easily accommodate access, protection, and security while supporting large numbers of files?

- **How dependable is the Hadoop cluster?**
  Can it be trusted for production and business-critical data?
  How does the distribution help ensure business continuity?
  Can the cluster recover data from user and application errors?
  Can data be mirrored between different clusters?

- **How does it perform?**
  Is processing limited to batch applications?
  Does the namenode create a performance bottleneck?
  Does the system use hardware efficiently?

MapR's innovations allow more businesses to harness the power of Big Data analytics. These technological advances make Hadoop easier, more dependable, and significantly faster, vastly broadening the scope of how and where Hadoop can be used.

A Complete, Advanced, Hadoop Distribution

The MapR Distribution for Apache Hadoop adds innovation to the excellent work already done by a large community of developers. With key new technology advances, MapR transforms Hadoop into a dependable and interactive system with real-time data flows.

The MapR Distribution for Apache Hadoop is 100% API compatible with Apache Hadoop including MapReduce, HDFS, and HBase. MapR fully tests and supports the complete distribution, combining MapR's intellectual property with the best of the best from the community, including the latest patches. As shown in Figure 1, MapR provides the entire Hadoop stack, including:

- Language access components (Hive and Pig)
- Database components (HBase)
- Workflow management libraries (Oozie)
- Application building libraries (Mahout)
- SQL to Hadoop database import/export (Sqoop)
- Log collection (Flume)
- The entire MapReduce layer
- Underlying storage services functionality

Overcoming the limitations of other Hadoop distributions, the MapR distribution is designed to scale efficiently from a single node to tens of thousands of nodes with petabytes of data.

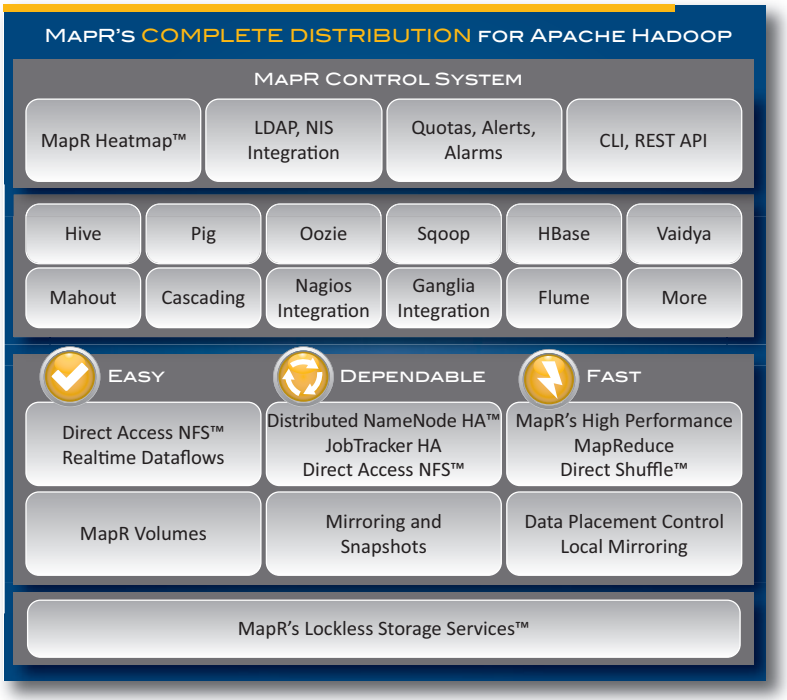


Figure 1. The MapR Distribution for Apache Hadoop is 100% compatible with Apache Hadoop, and adds multiple ease of use, dependability, and performance innovations.

Easier Hadoop

In order for Hadoop to be effective for broader groups of users and larger workloads, it must be easy to use, provision, operate, and manage at scale. MapR has invested in developing key breakthroughs that make it much easier to move data into and out of the cluster, provision cluster resources, and manage even very large Hadoop clusters with a small staff.

Moving From Batch to Real-Time Data Flows

Other Hadoop distributions manage data through a cumbersome batch process that slows data processing. The application first logs data to direct or network-attached storage. On some pre-determined time interval, data is then batch loaded into traditional Apache Hadoop's write-once file system. Finally, analytics are then run to create a result set, and those results are then batch-unloaded for further analysis.

The standard batch process causes a significant time lag between data production by the application and analysis in the Hadoop cluster. Loading data at a higher frequency minimizes this time lag, but results in a large number of small files that can easily challenge the scalability limits of traditional Hadoop. Other Hadoop distributions are also limited by the write-once Hadoop Distributed File System (HDFS). Like a conventional CD-ROM, HDFS prevents files from being modified once they have been written, and files cannot be read before they are closed.

In sharp contrast, MapR Direct Access™ NFS enables real-time read/write data flows via the industry-standard Network File System (NFS) protocol. With MapR Direct Access NFS, any remote client can simply mount the cluster. Application servers can write their log files and other data directly into the cluster, rather than writing it first to direct- or network-attached storage. Enabled by MapR Lockless Storage Services, MapR Direct Access NFS makes Hadoop radically easier and less expensive to use:

- Unlike the write-once system found in traditional Hadoop, the MapR distribution allows files to be modified, overwritten, and read as required. MapR Lockless Storage Services enable multiple concurrent reads and writes on any file.
- Graphical file browsers can be used to access and manipulate cluster data. Users can simply browse files, automatically open associated applications with a mouse click, or drag-and-drop files and directories into and out of the cluster.
- Files in the cluster can be edited directly by text editors and integrated development environments (IDEs).
- Standard command-line tools and UNIX applications and utilities (such as Grep, Sed, Tar, Sort, and Tail) can be used directly on data in the cluster. With other Hadoop distributions, the user must either develop their own tools, or copy the data out of the cluster in order to use standard tools.
- The MapR distribution greatly reduces the need for log collection tools (such as Flume) that often require agents on every application server. Application servers can either write data directly into the cluster or use standard tools like Rsync to synchronize data between local disks and the cluster.
- Application binaries, libraries, and configuration files can be stored inside the cluster and accessed directly, greatly simplifying operation.

Built-in Compression

While data compression can be done with traditional Hadoop distributions, it is difficult and inefficient. Data is typically compressed manually before copying it into the cluster and a special MapReduce job is then run to index the compressed data (assuming parallelism is desired in the application). Applications must also be modified so that they can consume the compression indices.

The MapR Distribution for Apache Hadoop provides automatic compression that offers both performance acceleration and significant storage savings. MapR's compression saves both network I/O bandwidth and storage footprint.

Multiple Cluster Support

It is often useful for organizations to operate multiple Hadoop clusters, whether to separate different data or applications, or for business continuity, or for performance. The MapR Distribution for Apache Hadoop is inherently designed to work with multiple clusters and provides direct access, remote mirroring, and multi-cluster management.

- ***Direct access.*** All MapR Hadoop clusters can be accessed directly and easily from both inside and outside of Hadoop. For example, if an organization has two clusters named "dev" and "test", then files on the dev cluster would be available under */mapr/dev* whereas files in the test cluster would be under */mapr/test*. These paths are identical whether the system is accessed through the Hadoop cluster (*hadoop fs -ls /mapr/dev/user/jdoe*) or remotely through NFS (*ls /mapr/dev/user/jdoe*). In addition, files can easily be copied between clusters with a simple copy command (*cp /mapr/dev/foo.txt /mapr/test/*). Symbolic links can easily be configured across clusters.

- ***Remote mirroring.*** The MapR distribution can be easily configured to mirror data between clusters using MapR Mirroring. This capability can be used for business continuity (by mirroring data to another cluster), or to keep a production and research clusters in sync.

- ***Multi-cluster management.*** All clusters running the MapR distribution of Apache Hadoop are visible through the MapR Control System (MCS). A user can easily view and switch between available clusters.

Provisioning, Operating, and Managing the Cluster

As data analysis needs grow, so does the need to effectively manage and utilize expensive cluster resources. Making data manageable at scale presents a significant challenge, both in terms of locating and accessing data, and in terms of applying policies to that data. Cluster infrastructure must accommodate many applications, users, and departments, and administrators need effective ways to apply policy to very large numbers of files. Cluster applications and data must also be provisioned in line with both technical needs and business-driven priorities.

Typical provisioning questions for enterprise applications might include:

- How much CPU capacity is needed (now and in the future)?
- How much storage is needed (now and in the future)?
- Will the applications require high I/O storage?
- What are the data protection requirements?
- What are the business continuity requirements?
- What security authorization and access control methods are required?

Answering these questions in a MapReduce context requires a Hadoop distribution with considerable depth and agility. Other Hadoop distributions require that policies (such as ownership, replication, etc.) be managed at the file level — a virtually impossible task given the potential for millions of files. The MapR Distribution for Apache Hadoop provides enterprise-class features and advanced data management functionality that lets organizations meet the business-level objectives outlined above simply, easily, and economically.

MapR Volumes

MapR Volumes make cluster data both easy to access and easy to manage by grouping related files and directories into a single tree structure so they can be easily organized, managed, and secured. MapR volumes provide the ability to apply policies including the following:

- ***Replication.*** The replication factor determines how many replicas of the data exist throughout the cluster.

- ***Snapshots.*** MapR Snapshots allow for online point-in-time data recovery without costly replication of data. A volume snapshot is consistent (atomic), does not copy data, and does not impact performance.

- ***Mirroring.*** MapR Mirroring allows for load balancing, cross-cluster backup, bulk data transfer, and failover for continuity. Local mirroring provides high performance for highly accessed data, while remote mirroring provides business continuity and integration between on-premise and private clouds.

- ***Quotas.*** Quotas allow organizations to accurately manage the needs of the application, user, or department by limiting the disk space for any user, group, or volume, and "charging" a volume to a specific user or group. The MapR Distribution for Apache Hadoop provides quotas for storage capacity within the cluster. Quotas can be configured on an individual volume or on a user or group. Over-quota email notifications can be sent automatically. Users and groups can come from the local system, or from standard name services such as NIS or LDAP.

- ***Data placement control.*** The MapR Distribution for Apache Hadoop allows data to be placed as desired within the cluster. For example, applications with significant I/O requirements could have their data placed on high-speed media such as SSDs, with other data placed on standard disk devices.

- ***Administration permissions.*** In some environments, a cluster administrator may want to delegate administration rights to others. Administrative permissions can be granted to allow specific users to create and remove volumes, perform mirroring and snapshots, and set quotas.

- ***Data access.*** User access to data can be established at the volume level. MapR integrates with standard directory services such as LDAP or NIS.

Easy Administration at Scale

At the scale of large Hadoop clusters, visibility and automation are essential. Administrators simply do not have the time to troubleshoot and manage servers individually. Advanced data management and self-healing capabilities make it easy for an individual administrator to manage a MapR cluster with thousands of nodes.

The MapR Distribution for Apache Hadoop includes the following to make administration easy:

- A complete, tested Hadoop stack pre-integrated with numerous components such as Hive, Pig, Oozie, etc.
- Simple installation
- Complete management through GUI, CLI, and REST APIs
- Rolling upgrades and downgrades with no downtime

The MapR Control System (MCS) provides full visibility into cluster resources and activity. As shown in Figure 2, the MCS includes the MapR Hadoop Heatmap™ that provides visual insight into node health, service status, and resource utilization, organized by cluster topology (e.g., datacenters and racks). Designed to manage large clusters with thousands of nodes, the MapR Hadoop Heatmap shows the health of the entire cluster at a glance. Filters and group actions are also provided to select specific components and perform administrative actions directly since the number of nodes, files, and volumes can be very high.

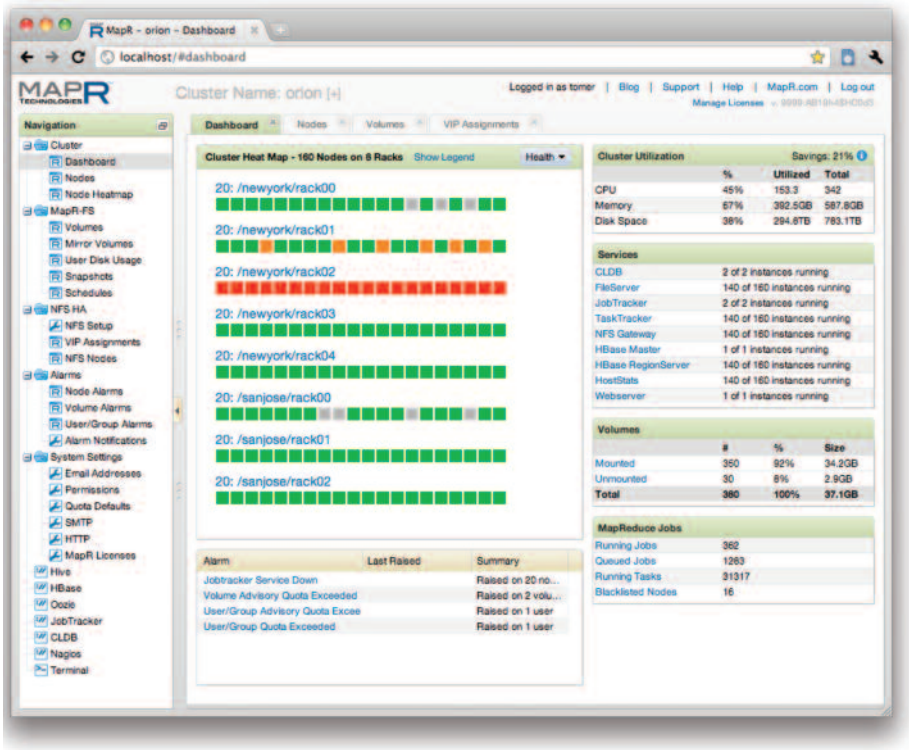


Figure 2. The MapR Hadoop Heatmap provides instant insights into the status of all nodes in each cluster.

Dependable Hadoop

Dependability is essential for business continuity, and enterprise organizations have demanding expectations in terms of reliability, availability, and storage functionality — particularly with regard to production data. Unlike other distributions, the MapR distribution features a distributed-everything architecture, functioning as an enterprise-ready cluster and trusted data store that is safe for shared environments:

- ***High availability.*** Everything in the MapR distribution is transactional, journaled and logged, and able to restart in seconds. The entire cluster is self-healing and self-tuning. The JobTracker and NameNode have been re-engineered to be distributed and replicated. NFS HA means that clients won't hang waiting for unavailable servers. Rolling upgrades make sure that the cluster is always available.

- ***Data protection.*** Unlike traditional Hadoop distributions, there are no single points of failure and a minimum of three-fold replication is provided for cluster metadata. The system guards against silent data corruption, and end-to-end checksums are performed all the way from the memory of the client through to the disk in the cluster. MapR Snapshots provide point-in-time recovery images while MapR Mirroring offers business continuity through both remote and local mirroring to protect data.

- ***Disaster recovery.*** Remote mirroring provides the capability to keep a synchronized copy of cluster data at a remote site, so that business operations can continue uninterrupted in the case of a disaster. Management of multiple on-site or geographically-dispersed clusters is simple with the MapR Control System.

- ***A safe, shared environment.*** The MapR distribution protects system resources against runaway jobs. MapR assures that no application can inadvertently take resources from the core cluster. In other distributions, a bug in a user job (such as an infinite loop) can impact critical system daemons.

- ***Monitoring.*** Filtered alarms and notifications are provided at many levels, including cluster-wide, per-service, per-volume, per user and group, and per-node with support for e-mail groups. Usage tracking and quotas help administrators effectively track and charge for resources. Integration with third-party monitoring systems is also provided.

Eliminating Lost Jobs

The Hadoop JobTracker keeps track of large numbers of Mappers and Reducers distributed across the cluster. Unfortunately, the JobTracker provided in other distributions runs only on a single node, and represents a single point of failure for the entire cluster. If the JobTracker fails, all running jobs fail and all progress is lost. In addition, administrators must first detect this situation in other Hadoop distributions, and then manually restart the JobTracker.

The MapR JobTracker HA improves recovery time objectives and provides for a self-healing cluster. Upon failure, the MapR JobTracker automatically restarts on another node in the cluster. TaskTrackers will automatically pause and then reconnect to the new JobTracker. Any currently running jobs or tasks continue without losing any progress or failing.

Distributed NameNode for HA and Scale

In Hadoop, the NameNode tracks where all the data is located in the cluster. In other Hadoop distributions, the NameNode runs on a single server, even for very large clusters which creates several problems. The MapR distribution provides a Distributed NameNode that eliminates these shortcomings.

- ***No Single point of failure.*** A single NameNode results in a single point of failure. If the name node goes down, the entire cluster becomes unavailable, requiring minutes or hours to restart the NameNode. With MapR, every node in the cluster stores and serves metadata, so that there is no loss or downtime even in the face of multiple disk or node failures.

- ***An unlimited number of files.*** Even with an exceptionally powerful server, the NameNode in other Hadoop distributions is limited to only about 70 million files. To attempt to work around this issue, many large Hadoop sites actively run Hadoop jobs to walk through the cluster and concatenate files — amounting to a significant percentage of their daily jobs and wasting both resources and money. MapR's distributed NameNode scales linearly with the number of nodes, eliminating the file limit.

- ***Performance advantage.*** In other Hadoop distributions, all metadata operations (e.g. lookups, creates) in the cluster have to go through a single NameNode, limiting performance. This shortcoming both impacts performance and restricts the workloads that can run on the cluster. With MapR, every node in the cluster stores and serves metadata, resulting in high performance that scales with the size of the cluster.

High Availability, Direct Loading for Hadoop

MapR provides several innovations that make NFS access useful and reliable. High availability (HA) has been implemented using virtual IP addresses, making Hadoop suitable for production environments. Multiple cluster nodes (up to every node in the cluster) can be designated as NFS gateways, and MapR manages failures transparently. Load balancing can be configured so that client connections are uniformly distributed among all of the NFS gateways. In addition, MapR allows NFS clients to control chunk size (e.g., 64 MB, 128 MB, 256 MB) and compression settings via a hidden pseudo-file in each directory (similar to */proc* files in Linux).

Snapshots for Easy Data Recovery

With the significant amounts of data being collected for processing today, true backups are often impractical. At the same time, organizations absolutely need the ability to revert their data to a specific point in time in the case of application corruption or user error. Replication is the only form of data protection offered by other Hadoop distributions. Unfortunately, replication only protects from disk and node failures, but does not protect from user and application errors since those errors are replicated throughout the cluster. Many Hadoop users have lost valuable data due to these errors.

MapR Snapshots let organizations address their recovery point objectives by providing a point-in-time recovery image that protects from user and application errors. MapR Snapshots can be scheduled or performed on demand, and they can be managed through the MapR Control System. MapR Snapshots operate on MapR Volumes. Recovering from a snapshot is as easy as browsing the snapshot directory and copying the directory or file to the current directory. Snapshots can be performed on individual volumes independently, and different volumes can have different schedules.

Complex schedules can be defined. For example, a snapshot schedule for "critical data" could entail:

- An hourly snapshot taken and retained for 24 hours.
- A daily snapshot taken at 12am, and retained for 7 days.
- A weekly snapshot taken on Sunday at 12am and retained for 12 weeks.

MapR Snapshots offer high performance and space efficiency and provide a number of distinct advantages.

- ***Fast performance.*** No data is copied in order to create a snapshot. As a result, a snapshot of a petabyte volume can be performed in seconds.

- ***Atomic.*** Snapshot operations are atomic and fully consistent.

- ***No impact on write performance.*** A snapshot operation does not have any impact on write performance. MapR uses redirect-on-write operations, meaning that each write in the system goes to a new block on disk. Redirect-on-write operations are more efficient than copy-on-write operations.

- ***Minimal storage usage.*** Snapshots do not consume any disk space until files are modified or deleted. Any unmodified blocks are shared between the snapshots and the current read/write image of the volume. As a result, MapR's snapshots use the minimum possible disk space, offering fast distributed snapshots with zero performance loss on writing to the original. Easy recovery is offered by simply copying a file from the snapshot.

Mirroring

Many organizations need to create physical point-in-time copies of their data. To address this requirement, MapR provides mirroring that can be used in two distinct forms. Remote mirroring (described in this section) provides mirroring between clusters for disaster-recovery, development/test, or private-public cloud integration. Local mirroring (described in the section on performance) can be used for load balancing and performance enhancements within the same cluster. Remote mirroring can facilitate numerous use cases.

- ***Disaster recovery (DR).*** With remote mirroring, organizations can deploy a secondary cluster for disaster recovery purposes — typically in a different datacenter or different geographic area. Data can then be recovered from the secondary cluster in the event of data loss. Moreover, entire applications can fail over to the secondary cluster if a disaster impacts the primary cluster.

- ***Research cluster.*** With MapR, organizations can easily deploy a research (or dev/test) cluster alongside their production cluster. The administrator simply creates mirrored volumes on the research cluster, and the system periodically mirrors the data from the production cluster to the research cluster. This capability allows users of the research cluster to operate on real, up-to-date data.

- ***Private-public cloud integration.*** While most organizations choose to run Hadoop on their own hardware infrastructure, some may find it useful to occasionally launch an additional cluster in the public cloud (e.g., Amazon EC2) for extra compute capacity. For example, an organization might decide to launch a 100-node cluster on EC2 every Friday night to address a special processing need. MapR makes it easy to synchronize data between a local cluster and a cluster in the public cloud.

- ***Efficiency.*** MapR mirrors are differential, meaning that only deltas are transferred from the source to the destination. For example, if only a single 8 KB block in a file has been modified since the previous mirroring operation, only that block will be transmitted in the next mirror. All data is compressed on the wire to expedite transfer and checksums are used to ensure integrity. When more than one mirror of a volume is desired, mirrors can be cascaded to minimize transmission bandwidth needs. Data is transferred asynchronously and in parallel by the servers providing the source volume, and does not impact local system performance.

- ***Network or sneakernet.*** If there is too much data to mirror over a network connection, the mirror can be dumped to one or more portable disks or servers at the source location, and then physically shipped and loaded at the destination (so-called "sneakernet mirroring"). Network and sneakernet mirroring can interoperate. For example, a large data mirror can be created at the source cluster, shipped to a remote location, and then bulk loaded and synced over the network with the remote cluster.

- ***Atomic operation.*** MapR Mirroring is based on MapR Snapshots, inheriting atomic operations. No change occurs on the destination cluster until all of the data has been received for a given mirroring operation. The destination cluster is then updated atomically. Mirroring relationships can be configured and monitored through the GUI, CLI or REST API. A mirroring schedule (similar to a snapshot schedule) can be set up at a volume level.

Faster Hadoop

From the beginning, the MapR Distribution for Apache Hadoop was designed to provide significant I/O and performance advantages. MapR offers enhanced performance on clusters of any size, from a single node to thousands.

Architected for Performance

From the outset, the MapR distribution has been re-designed for performance, with architectural advancements at all levels, including:

- ***MapR Lockless Storage Services.*** MapR Lockless Storage Services accelerate MapReduce performance and provide multi-dimensional scalability. Data routing tables are used rather than mutexes or spinlocks in the implementation, eliminating lock contention. State machines are used rather than threads, reserving thread resources for executing user applications that share the nodes. MapR Lockless Storage Services write directly to block devices rather than through HDFS or Linux file system layers.

- ***Optimized shuffle.*** MapR's shuffle takes advantage of MapR Lockless Storage Services for shuffling data between Mappers and Reducers. As a result, when the Reducers read the Mapper outputs, the data is read from disk sequentially (potentially across multiple disks), resulting in very high performance. The optimized MapR shuffle does not utilize the Linux page cache, and thus avoids competing with user applications for precious memory resources. The result is a shuffle that is typically three times faster than other distributions.

- ***Distributed NameNode.*** Other Hadoop distributions rely on a single NameNode for all metadata operations for the entire cluster. In contrast, the MapR Distributed NameNode distributes metadata operations across all cluster nodes, yielding orders of magnitude greater scalability.

- ***Built-in compression.*** Data is transparently compressed with the MapR distribution, saving disk and network I/O.

- ***Implementation language.*** The lower layers of the MapR distribution are written in the C/C++ language. Beyond higher efficiency and performance, this implementation choice also overcomes the garbage collection issues that impact other distributions.

- ***Multi-NIC support.*** Most servers today are equipped with at least two network interface controllers (NICs). The MapR distribution can utilize multiple network interface controllers (NICs) on each node via NIC bonding, without requiring port trunking at the switch level. From the network perspective, only one socket is opened between any two peers in the MapR Hadoop cluster.

- ***Minimal CPU/memory footprint.*** Since MapReduce applications run on the same nodes as file services in Hadoop clusters, MapR set out to minimize the CPU and memory footprint of the infrastructure itself. A minimal footprint preserves CPU, memory, and other resources for application processing. In addition, all MapR services are implemented as user-space processes — yielding performance benefits without compromising stability.

Performance-Related Features

Beyond the architectural advantages of MapR's implementation, data placement control and local mirroring are provided to customize Hadoop implementations and increase performance.

- ***Data placement control.*** The MapR distribution provides data placement control, unlike other Hadoop distributions that offer no control over where data is physically stored in the cluster. Data placement control can be used to set the policy on a volume to restrict that volume to a subset of the available nodes. A volume could be restricted to nodes in a specific rack or datacenter, or to nodes with a specific hardware configuration. For example, an application with a randomly and frequently-accessed session lookup table could have the MapR Volume containing the lookup table restricted to the nodes in the cluster equipped with SSD drives, thus resulting in much higher performance.

- ***Local mirroring and Mirror Volumes.*** MapReduce spawns large numbers of processes that often access the same data. With traditional Hadoop technology, these simultaneous accesses to a single file system element can quickly overwhelm file servers and slow performance, particularly at job startup. With local mirroring (also known as Mirror Volumes) multiple copies of a volume can be created. The copies are updated asynchronously, and all copies are accessed through the same path. The system automatically load balances the read requests between these copies.

Performance Testing

MapR has done performance testing to compare and evaluate the performance of the MapR Distribution for Apache Hadoop with other distributions. Results in the categories of streaming I/O, random I/O, and MapReduce performance are provided in the sections that follow.

Streaming I/O Performance

As one of the first standard benchmarks for I/O performance in a Hadoop context, the DFSIO benchmark provides a useful metric to gauge streaming I/O performance. The benchmark is a MapReduce job with multiple mappers and a single reducer. The key measurement is the transfer rate (in MB/s) for an average mapper. For this test, MapR engineers evaluated DFSIO on a 10-node cluster (Figure 3).

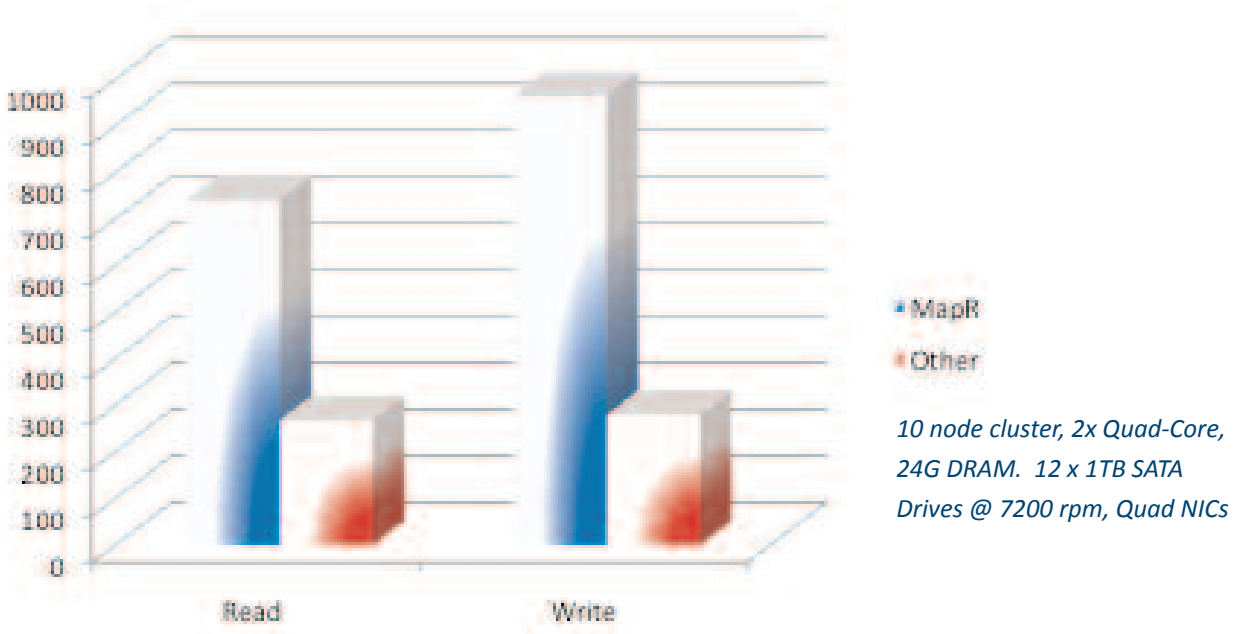


Figure 3. The DFSIO benchmark (larger is better) performs three times faster on the MapR Distribution for Apache Hadoop than on traditional Apache Hadoop.

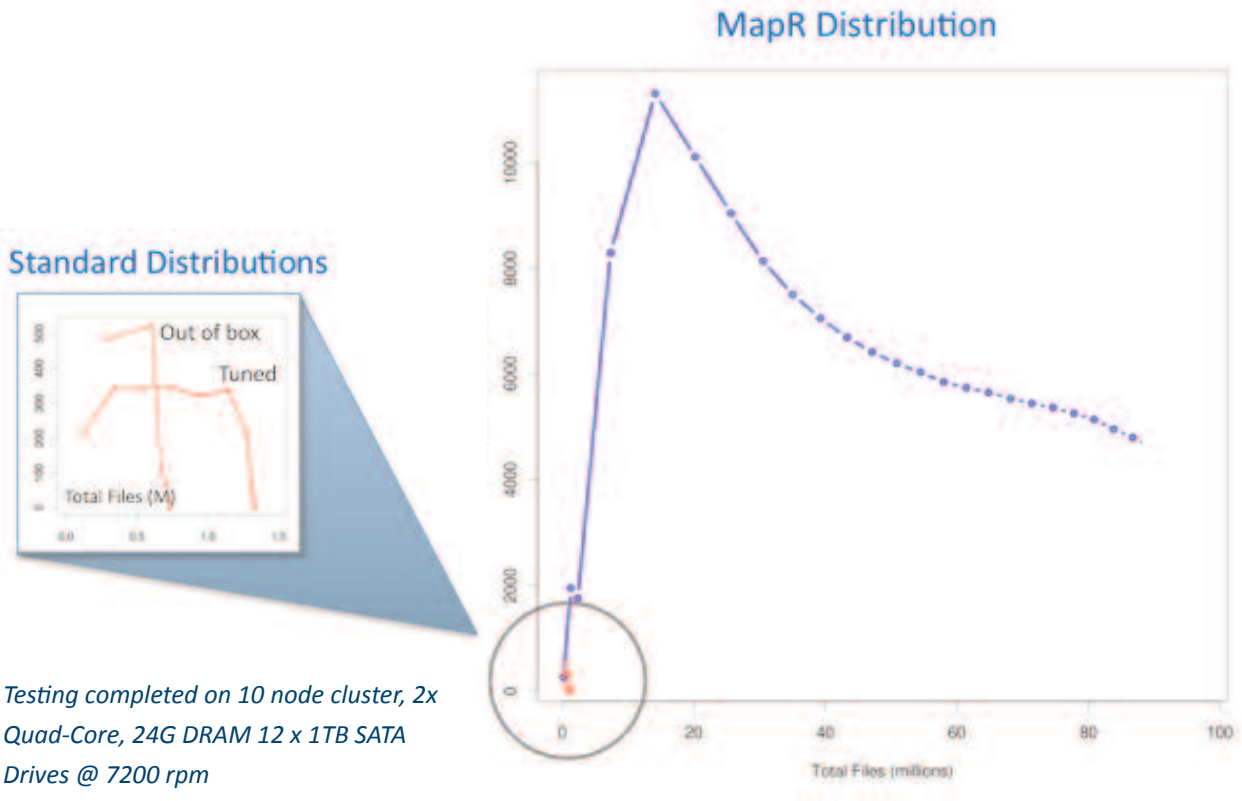
The systems used in the 10-node cluster for this test featured two quad-core processors, 24 GB of RAM, and twelve 1 TB 7200 RPM SATA disk drives. As the chart illustrates, I/O was basically running at the physical limitation of the hardware for the MapR distribution. The CPU was basically idle during these tests, attesting to efficient data paths. Write speed in the test is slightly slower due to checksum computations.

Random I/O Performance

Some applications require the ability to create and address large numbers of files. To evaluate performance in this area, MapR engineers tested the MapR Distribution for Apache Hadoop against traditional Hadoop with a variation on the standard NNBench test that performs the following steps repeatedly:

- Create a file
- Write 100 bytes to the file
- Close the file

The application was run on the same 10-node cluster configured with both traditional Apache Hadoop and the MapR distribution. Block reports were disabled on traditional Apache Hadoop to allow completion of the test. The results of the test are shown in Figure 4.



Testing completed on 10 node cluster, 2x Quad-Core, 24G DRAM 12 x 1TB SATA Drives @ 7200 rpm

Figure 4. The MapR Distribution for Apache Hadoop outperforms traditional Apache Hadoop for random I/O.

The MapR Distribution for Apache Hadoop demonstrated dramatically better results in both rate (vertical axis) and volume (horizontal axis). In fact, the disparity is so large that the traditional Apache Hadoop results have to be magnified to even be visible. Even with block reports turned off, traditional Apache Hadoop was able to store less than 1.5 million files before the rate dropped precipitously. In contrast, the MapR distribution was able to write 90 million files total at rates of 12,000 declining to 4,000 files per second. These results demonstrate a sixty-fold scalability factor.

MapReduce Performance

Beyond I/O, engineers wanted to evaluate the data analysis performance of the MapR distribution. The Terasort benchmark was run on a 10-node cluster, with each node comprised of two quad-core processors, 24 GB of RAM, and twelve 1 TB SATA disk drives. The results of the benchmark test are shown in Figure 5.

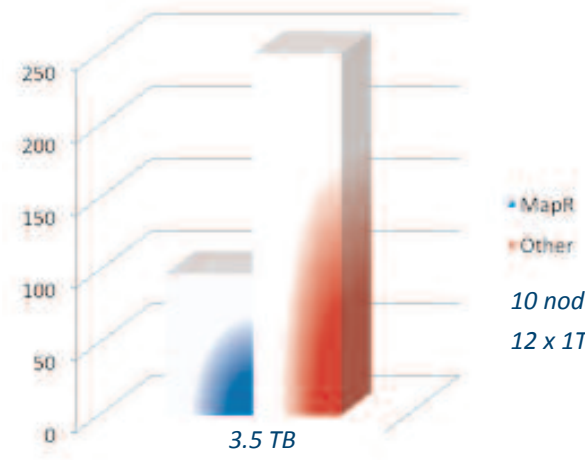


Figure 5. The MapR Distribution for Apache Hadoop provides almost three times the performance of traditional Hadoop in the Terasort benchmark (smaller is better).

ATTRIBUTE	APACHE HADOOP	MAPR DISTRIBUTION FOR APACHE HADOOP
Cluster Size	1-3,000	1-10,000+
Maximum Data in Cluster	20 PB	1,000 PB
Maximum Files	70 Million	1 Trillion
Volumes	—	200,000
Performance	1x	3x to 20x
Infrastructure Footprint	1x	50% Smaller

Table 1 compares scalability between the MapR distribution and other Apache Hadoop distributions.

Conclusion

MapR believes that the increasing criticality of data processing demands a strategic focus on the choice of Hadoop platform. While other distributions are already available, only the MapR Distribution for Apache Hadoop provides a demonstrable difference and addresses other Hadoop shortcomings and limitations (Table 2). The MapR Distribution for Apache Hadoop provides unique features and functionality that simply are not available with any other Hadoop distribution, including:

- Simple installation, provisioning and manageability for cluster visibility and ease of use
- Enterprise-class dependability, storage access, and storage management
- Breakthrough performance leading to cost containment through drastically lower hardware requirements

MapR's investments and innovations make Hadoop easy, dependable, and fast for today's most demanding applications — while providing Hadoop infrastructure that is ready for the challenges to come.

CRITERIA	TRADITIONAL APACHE HADOOP	MAPR DISTRIBUTION FOR APACHE HADOOP
Completeness	No	Yes. Includes popular tools. Does not require expensive external backup storage
Performance	1x	3x to 20x performance improvement, data placement control, and optimized shuffle
Cost of Hardware	1x	1/2x and first-year ROI
Scalability (files)	70 Million	Unlimited number of files. MapR Distributed Name Node for scalability
Ease of Operation	No	MapR Heatmap, GUI, Alarms and filters
Continuous data load and access to results	No	MapR Direct Access NFS for MapR Realtime Hadoop
Dependability and Business Continuity	No	MapR JobTracker HA, MapR Distributed NameNode, MapR Snapshots for point-in-time recovery, MapR Mirroring for disaster recovery
Reliable sharing of the cluster	No	Alerts, alarms, quotas, users, groups, and system resource isolation
Provisioning & usage tracking	No	MapR Volumes, quotas, and users
Multi-datacenter	No	MapR Mirroring across clusters and datacenters. Manage multiple clusters from a single GUI
Support	Available	Yes

Table 2 The MapR Distribution Advantages

MAPR TECHNOLOGIES

MapR Technologies is the creator of the industry's fastest, most dependable and easiest to use distribution for Apache Hadoop. MapR Technologies is dedicated to advancing the Hadoop platform and ecosystem to enable more businesses to harness the power of big data analytics for competitive advantage. For more information, please visit www.mapr.com.