

Advancing Hadoop—MapR's M7 Edition

By John Webster

December 12, 2012



Evaluator Group

Enabling you to make the best technology decisions



Table of Contents

The HBase Value Propositions	3
HBase and the Enterprise	4
Compactions.....	4
Region Server Outages	5
No Point in Time Recovery Capability	5
Competition for Disk Space and Disk Utilization.....	5
Limitations Inherited from HDFS.....	5
MapR.....	5
Hadoop Cluster Availability	6
Data Protection and Integrity.....	6
MapR M7 Edition.....	6
Volume Management for HBase	7
Cluster Recoverability and Business Continuance	8
Data Corruption.....	8
Site Failure	8
Planned and Unplanned Downtime	8
Performance.....	8

The number of enterprise-level deployments of Hadoop MapReduce is rising quickly, driven by a need to understand and potentially adopt this new business analytics platform for business applications. We note that pilot Hadoop projects are underway within many of the Fortune 1000 group of companies. Responding to this demand, the Hadoop ecosystem is now offering “enterprise” versions of Hadoop.

As we noted in our previous Evaluator Group Technology Insight Paper entitled “Evaluating Hadoop in the Enterprise,” we noted that Version 1.0 of Apache Hadoop could be made more enterprise ready by:

- Eliminating well-known single points of failure (Name Node and Job Tracker)
- Addressing the potential for data loss resulting from data corruption that can be propagated across data copies created by HDFS
- Providing disaster recovery capabilities through the implementation of remote data mirroring
- Advancing the manageability of Hadoop clusters by IT administrators who have little expertise with Hadoop going into enterprise data center-level implementations

But these are not the only ways that Hadoop could be advanced that would appeal to enterprise IT administrators who ultimately have the responsibility for delivering Hadoop-based applications to users, as well as assuring their usability and availability. Here we look at the capabilities of HBase and illustrate how an HBase implementation could be optimized with MapR's M7 Edition.

The HBase Value Propositions

HBase is a “NoSQL” database. As such, it is not an RDBMS that uses SQL queries to access data. It is however a distributed database that is capable of large scale implementation with “strong consistency” – meaning that consistency across nodes in the distributed cluster is given high priority. This characteristic makes it highly suitable for large scale implementations where data integrity is important such as consumer-oriented applications that are Web-facing (i.e. shopping carts for large online retailers). Many Hadoop MapReduce users also run HBase.

The other very significant aspect of this distributed database for Hadoop users is that HBase uses the Hadoop Distributed File System (HDFS) as its data store meaning that the data sets used by Hadoop MapReduce and HBase are one and the same. There are at least two advantages to HBase/MapReduce data sharing:

1. Hadoop MapReduce is widely considered to be a batch process. However, there is growing desire among Hadoop users to make the platform responsive to applications needing real time data access and information delivery. Converging MapReduce and HBase to the same platform means

that data stored in HDFS as a result of applications running on HBase are available to the MapReduce side of Hadoop.

2. The ability of enterprise IT to move data—and large volumes of it—will become increasingly critical to the success of Hadoop. However, data movement in this context has multi-dimensional challenges:
 - Data has to get from the many sources to the Data Nodes
 - Data has to flow between Name Nodes and Data Nodes during the analytics process
 - Results in the form of data have to be delivered to information users
 - Source data and the results of Hadoop-based analytics processes are likely to be shared with other systems, again requiring data movement

One way to reduce the volume of data burdening networks that are dedicated to Hadoop, as well as existing enterprise networks, is to not move data to and from Hadoop in the first place. This can be done with structured database-generated data by converging database processes that are often a data source with Hadoop's MapReduce processes. In essence, one supports both database and analytics applications from the same processing cluster allowing Hadoop analytics to be run against structured and unstructured data, both of which are already cluster-resident.

HBase and the Enterprise

For the reasons noted above, the convergence of HBase with Hadoop MapReduce offers advantages, not the least of which is the ability to run large volume analytics and database applications on the same distributed cluster. However, because HDFS and HBase are still evolving within the Apache open source community, there are a number of manageability and availability issues with Hadoop that have yet to be addressed and that will give potential enterprise users pause.

Compactions

Under normal operating conditions, HBase can create a large number of HFiles. In order to maintain storage efficiency, IT administrators run a compaction process within HBase that reclaims wasted disk capacity that builds up over time and reduces the number of files HBase has to track. HBase includes a compaction process that periodically rewrites multiple HFiles into a single HFile per region.

There are two types of compaction processes available to IT administrators: “major” and “minor.” Minor compactions look for small files and merge them into larger files. “Major” compactions look at all files

within a region to perform mergers and also deletes extraneous data. However, as useful as these processes are, compactions disrupt HBase operations. Running a compaction process drives significant I/O overhead and for this reason, Hadoop administrators run them manually during off-peak workloads to avoid degradation in cluster performance.

Region Server Outages

HBase uses RegionServers to manage HBase processing functions. A RegionServer outage can cause data to be unavailable for up to thirty minutes while the write-ahead logs (WALs) are replayed for the impacted regions. Reconfiguration and other administrative tasks related to RegionServers will require that the entire cluster be taken down.

No Point in Time Recovery Capability

Exposure to data loss in Apache Hadoop is still a major concern. The ability to use point-in-time snapshots to recover from an occurrence of data corruption has long been available to enterprise storage administrators, but is still not available to Hadoop administrators.

Competition for Disk Space and Disk Utilization

HBase competes with HDFS for disk space on cluster nodes. A cluster node with 36TB disk space for example can only handle about 2TB of HBase data. Additionally, HBase does not take into account block locations when assigning data to regions causing RegionServers to read data over the cluster network rather than the local drives and adding processing latency.

Limitations Inherited from HDFS

The Name Node in version 1.0 of Apache Hadoop which, among other things, stores metadata for the Hadoop cluster has become a well-known single point of failure. If the Name Node fails, data could be lost. Similarly if the JobTracker fails, the job has to be reloaded and rerun from the beginning. More detail on this issue is covered in Evaluator Group's Technical Insight Paper entitled "Evaluating Hadoop in the Enterprise Data Center."

MapR

Hadoop is now being adopted as an alternative to traditional data warehouse-based analytics systems by enterprise IT. In an Evaluator Group Technology Insight paper entitled "Evaluating Hadoop in the Data Center," we noted that IT administrators should evaluate Hadoop on the basis of established management requirements and suitability to enterprise production data centers. We also noted that, while the contributors to Apache Hadoop are at least aware of these requirements, progress toward

addressing them has been slow causing some enterprises to push production deployments to the back burner.

MapR, founded in 2011, develops, distributes, and supports a distribution of Apache Hadoop that addresses many of the enterprise quality issues currently limiting its implementation in production data centers. MapR replaces the Hadoop Distributed File System (HDFS) with one that eliminates weak points in HDFS that enterprise IT administrators are particularly sensitive to, but is fully compatible with MapReduce, HDFS and HBase. The imitations addressed include but are not limited to:

Hadoop Cluster Availability

MapR has eliminated the Name Node/Data Node distinction by distributing metadata processing across cluster nodes. Therefore, there is no exposure to Name Node failure. In addition, planned upgrades to Apache Hadoop currently require outages. MapR supports rolling upgrades with no downtime.

Data Protection and Integrity

MapR supports both local and remote data mirroring functions so that Hadoop can be included under the enterprises' disaster recovery/business continuance plan. MapR also supports snapshots to recover from human and machine-induced occurrences of data corruption.

MapR M7 Edition

To address some of the operational and performance issues of the standard Apache distribution of HBase MapR has introduced its M7 Edition. M7 unifies the storage and processing of files (HDFS) as well as tables (HBase) to a single platform. As a result, MapR's existing management, access and data protection capabilities described above now apply to applications supported by HBase.

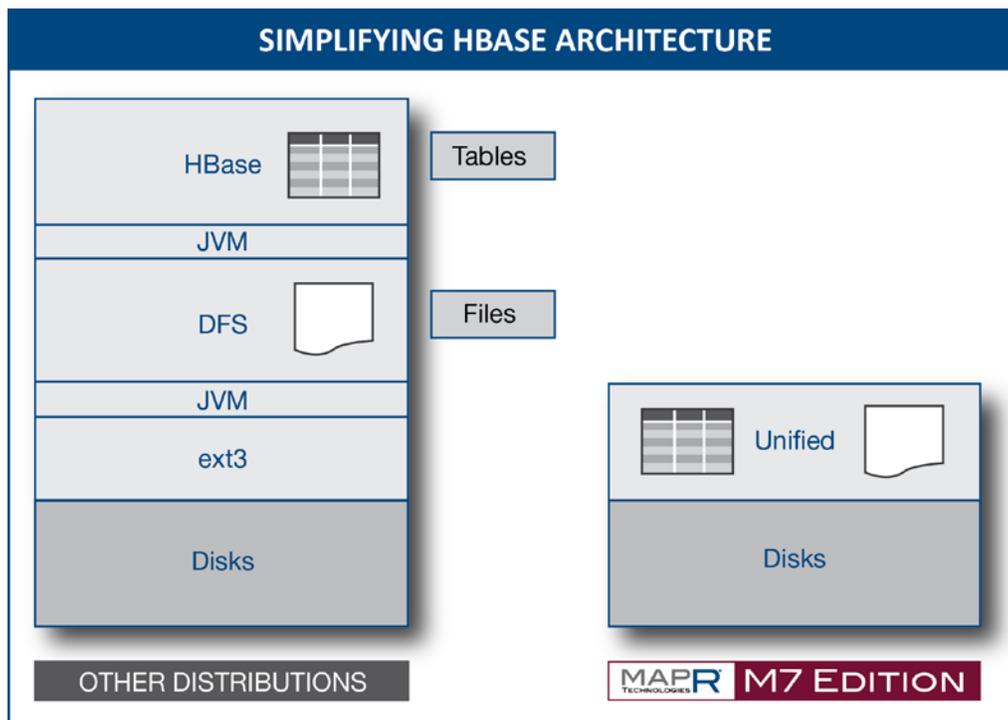
MapR Editions

MapR currently offers three Hadoop editions:

M3 – a free distribution that includes HBase, Pig, Hive, and other popular Hadoop components.

M5 Enterprise – same as M3 but includes MapR's advanced high availability and data protection features and is available on a fully supported subscription basis.

M7 – includes the enterprise grade features for HBase described below plus everything in M5.



Source: MapR Technologies, Inc.

Figure 1: HBase tables are layered on top of distributed files in the Apache and other distributions (left), but co-exist in M7 and share a common namespace (right)

Volume Management for HBase

A key differentiator between Apache-based distributions of HBase and M7 is the use of storage volumes. With M7, the integration of files and tables into a single data store allows administrators to manage both tables and files using volumes and directories via a single management interface. Typical HBase implementations require administrators to perform RegionServer maintenance, manual table splits, merges and compactions. With a unified namespace for tables and files stored within volumes, administrators can set policies, quotas and access privileges uniformly for individual users or groups of users at a project level on both files and tables. Additionally, the automated management capabilities found in MapR, such as data placement and multi-tenancy, are extended to HBase workloads. Isolated work environments can be created for different database users or applications and tables can be placed on specific hardware for better performance or load isolation.

M7's unified, volume-based architecture allows developers to create their own tables without administrative assistance. Developers can create as many temporary tables as needed to optimize

application workflow, store large objects such as images or videos in a table, and expand the use of HBase to more applications.

Cluster Recoverability and Business Continuance

M7 supports recovery from a number of failure modes that traditionally have been the subject of great concern for enterprise administrators:

Data Corruption

The implementation and use of volumes allows enterprise IT administrators to apply their knowledge of volume management to cluster reliability and recoverability. For example, MapR's snapshot capability is extended to HBase tables such that administrators can integrate the scheduling of MapR's snapshots for table and files and associate these snapshots with specific users or specific applications. In the event of data corruption, point-in-time recovery of tabular data can be performed by directly accessing the snapshot version of the tables and or files affected. Recovery can be as granular as desired, from the cell level up to multiple tables in the case of tabular data, and at the individual file level for files data.

Site Failure

Tables can also be mirrored to geographically dispersed backup clusters for business continuity purposes or for providing geographically distributed access. In the case of a primary site failure, M7 is immediately available at the recovery site.

Planned and Unplanned Downtime

Rolling upgrades can be used to avoid planned downtime for applications using tabular data. MapR clusters can be upgraded a few nodes at a time and without disruption to application availability. Unplanned downtime is avoided via M7's ability to recover automatically from node failures.

Performance

Performance of the cluster supporting HBase depends on several underlying storage subsystem factors:

- Latency introduced by I/O to disk
- Data locality
- Read amplification (ratio of read operations from disk to application reads)

- Write amplification (ratio of write operations to disk to application writes)
- I/O storms
- Disk space overhead
- Skewed data handling (rewriting values for similar keys) come into play

M7's design has addressed all of these issues with a purpose-built architecture to process files and tables within a unified data layer. Both tables and files share a common namespace and can be managed using volumes and directories. With M7, the layered architecture of HBase is eliminated. HBase applications can access data directly without the redundancy of extra layers of communication yielding a single, scalable and more reliable data store that offers higher performance and is easier to develop to and administer.

Evaluator Group Assessment

The convergence of a NoSQL database engine (HBase) with MapReduce analytics under the Hadoop processing umbrella can offer significant advantages to enterprises looking at Hadoop alternatives to the more traditional styles of database operations and data warehousing that have been in use for decades. These include greater Hadoop-related hardware and network efficiency, simplified management, and progression toward real time analytics capabilities. However, HBase is currently limited in its ability to be considered ready for to support enterprise quality production applications.

MapR is the first to address the many of HBase's limitations with the release of M7—a Hadoop MapReduce implementation that offers an enterprise-quality database and file system in a single system. As such, it consolidates structured and unstructured data types and makes them immediately available to MapReduce-base analytics application.

About Evaluator Group

Evaluator Group Inc. is dedicated to helping IT professionals and vendors create and implement strategies that make the most of the value of their storage and digital information. Evaluator Group services deliver in-depth, unbiased analysis on storage architectures, infrastructures and management for IT professionals. Since 1997 Evaluator Group has provided services for thousands of end users and vendor professionals through product and market evaluations, competitive analysis and education. www.evaluatorgroup.com Follow us on Twitter @evaluator_group

Copyright 2012 Evaluator Group, Inc. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, or stored in a database or retrieval system for any purpose without the express written consent of Evaluator Group Inc. The information contained in this document is subject to change without notice. Evaluator Group assumes no responsibility for errors or omissions. Evaluator Group makes no expressed or implied warranties in this document relating to the use or operation of the products described herein. In no event shall Evaluator Group be liable for any indirect, special, consequential or incidental damages arising out of or associated with any aspect of this publication, even if advised of the possibility of such damages. The Evaluator Series is a trademark of Evaluator Group, Inc. All other trademarks are the property of their respective companies.