"Big Data" Management and Security Application to Telemetry Data Products

Jeff Kalibjian

Hewlett Packard Corporation

ABSTRACT

"Big Data" [1] and the security challenge of managing "Big Data" is a hot topic in the IT world. The term "Big Data" is used to describe very large data sets that cannot be processed by traditional database applications in "tractable" periods of time. Securing data in a conventional database is challenge enough; securing data whose size may exceed hundreds of terabytes or even petabytes is even more daunting! As the size of telemetry product and telemetry post-processed product continues to grow, "Big Data" management techniques and the securing of that data may have ever increasing application in the telemetry realm. After reviewing "Big Data", "Big Data" security and management basics, potential application to telemetry post-processed product will be explored.

KEYWORDS

Big Data, Hadoop, Big Data Security, MapReduce.

HOW BIG IS "BIG?"

As mentioned above the term "Big Data" implies data sets so large that they cannot be stored or analyzed utilizing conventional clustered Relational Database Management Systems (RDBMS) servers with attached storage. The notion is very easy to understand at a "gut" level: that is, the amount of data coming into the organization is ever increasing; yet, the ability to analyze the data is ever decreasing! While these concepts are somewhat relative to IT organization size----in the end every company reaches the same dilemma: they just can't keep up with the volume of data. Table 1 illustrates some current data challenges in industry and the public sector. Observe that in some extreme situations, it may be simply beyond the scope of current technology to capture and process all the data.

YOU'LL KNOW IT WHEN YOU "V' IT

Generally "Big Data" can be characterized by three characteristics: volume, velocity, and variety. While the sheer volume of data produces the "Big," it is the data types (or variety) of

Table 1 Some Relative Data Sizes

Hubbell Space Telescope	0.01 Tb/day (11.8 Gb/day)
Boeing 787	One round trip typically generates 1 TB of data
Twitter	7 Tb of data per day (2012 statistic)
Facebook	10 Tb of data per day (2012 statistic)
Large Hadron Collider (LHC)	42 Tb/day

the data that dictates the “non standard” storage paradigm. Data warehousing applications typically do well with structured data, while unstructured data is better suited for a specialty approaches, like Hadoop [2]. Finally, while velocity is pretty self-explanatory, it should be

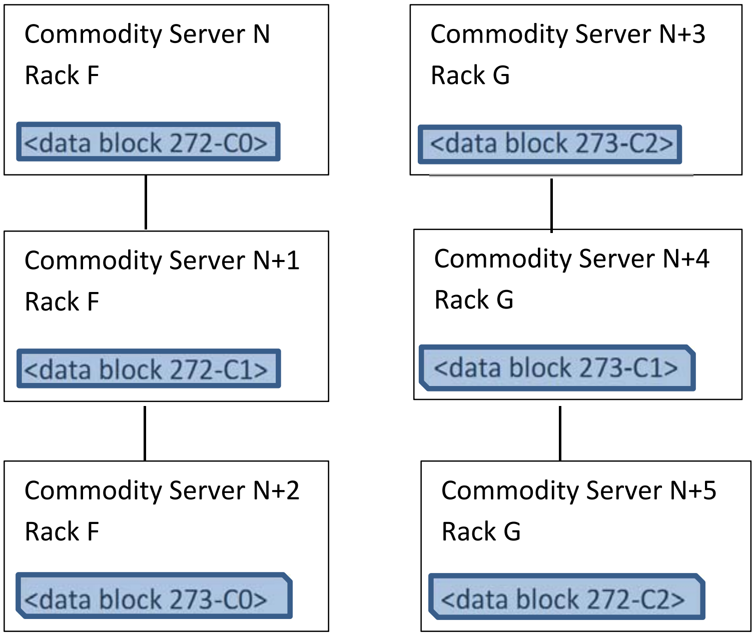

Figure 1. HDFS data block “striping.” Data blocks are redundantly written to two other servers in the HDFS cluster, including to at least one server in a different rack. Note” –C0” read as “Copy 0,” “–C1” read as “Copy 1,” etc.

noted that it not only conveys how fast the information may be flowing into an organization; but how quickly it must be processed to reach a decision point in order to perpetuate a feedback loop. Sometimes data may be at such a high velocity, that it cannot be saved in its entirety. In such cases "stream data processing" must be done in order to reduce the data to merely an immense "tractable" size!

HADOOP THE BIG DATA WORKHORSE

A discussion of "Big (unstructured) Data" would not be complete without elaborating on the most popular tools for storing and analyzing it; namely, Hadoop. Hadoop is actually a programming environment that is comprised of The Hadoop Distributed File System (HDFS) and The Hadoop programming environment (utilizing MapReduce). Today Hadoop is open source. It is an Apache project in the Apache Software Foundation. It started out as a Google project in which a distributed and redundant file system was created (Google File System, GFS) to store extremely large amounts of data that could be analyzed by mapper and reduce tasks (hence the term MapReduce used above) to manipulate and analyze data. The HDFS is architected to be deployed on many (i.e. thousands and more) networked "commodity servers" with just a few (not necessarily RAIDed; Redundant Array of Independent Disks) "cheap" disk drives. Such servers typically have an unimpressive reliability record (i.e. Mean Time to Failure, MTF, high). However, that is not a problem because in HDFS data is broken up into smaller blocks (typically 64 Mb) and are not only written to one node in the HDFS cluster, but two others as well---with the requirement that at least one of those two nodes the data is redundantly written to reside in a different server rack (see Figure 1). This replication feature is thus able to accommodate the high MTF rates of the commodity servers likely to be seen in Hadoop clusters. The NameNode server in the HDFS keeps track of all data placement; hence it is a very important node and is often implemented with redundancy as well.

MAPREDUCE: THE HADOOP DATA PROCESSING PARADIGM

Hadoop makes available two distinct processes that facilitate "big data crunching;" namely map and reduce; and hence the term "MapReduce." Map can be thought of as data transformation---that is, sets of data are taken in as input, and other sets of data produced in output----the key here being "manageable" sets of data. Manageable in the sense of can be comfortably crunched on one of the commodity servers mentioned in the prior section. So basically in the map phase, the problem is broken up into pieces that can be distributed on the commodity servers in the cluster to work on in parallel and hopefully not re-distributed; that is, when at all possible process the data where it resides! Reduce actually can combine the transformed data sets into smaller sets of data---essentially recombining the pieces and doing additional processing on those recombined results. The concepts lend themselves to "excessive" parallelism----implying the commodity servers mentioned above are maximally leveraged to crunch data in "manageable" sets that taken in their entirety would be entirely “unmanageable.”   So basically it’s divide, distribute, crunch, re-combine, crunch, reduce, and conquer paradigm.

SECURING BIG DATA

A fundamental question arises as to whether a data set is large or small; does it truly effect the requirements for securing the data? The answer is a resounding no! The criteria for securing data should be primarily related to its value (and the economic loss associated with its compromise). However, implementing a robust security paradigm around a distributed processing/storage framework like Hadoop will present its challenges. As mentioned in the prior sentence, the security goals will remain the same as with any sensitive data; namely,

- Protect the data at rest
- Protect the data when moved
- Protect the data when processed
- Protect intermediate data product
- Protect who has access to the data
- Ensure data integrity

Assuming security of the data is a requirement for a data set in Hadoop, how might the goals in the prior section be implemented? The first element would start with cryptographic key management. A robust key management system (preferably FIPS, Federal Information Processing Standard, 140-2 [3] evaluated) would need to be implemented to store and manage the cryptographic keys used to secure the data blocks at rest in the Hadoop cluster. Depending on the types of the data in the data blocks one or more keys may be required per commodity server. When data is in transit it is recommended TLS, Transport Layer Security (formally known as Secure Sockets Layer, SSL) be utilized with both client and server authentication enabled. This is important because when an element in the Hadoop cluster is contacted, it is vital that it is validated as a legitimate member of the cluster. The commodity servers themselves should be using at minimum a locked down Linux operating system, or better yet an operating system that has a Common Criteria [4] evaluation.   A secure OS will help with the satisfying of the protecting of the data when being processed; robustly implementing that goal might imply extreme measures like deploying FIPS 140-2 evaluated hardware in the Hadoop cluster. So much for the “commodity server” concept! Lastly, protecting the integrity of the original data might require the data blocks to be originally digitally signed so it can always be verified that the original data in the cluster is intact. Maintaining the integrity of intermediate “MapReduce” product could also be achieved with digital signatures (if needed). These concepts are illustrated in Figure 2.

BIG DATA AND TELEMETRY

The trend in telemetry has been, is presently, and will continue to be “send more” and the result is data thresholds that can easily exceed the “Twitter” or “Facebook” daily rates (see Table 1). This implies the opportunity to use Hadoop technologies for some aspect of telemetry data management/analysis. Consider Table 1 again and notice that a round trip of a Boeing 787 produces approximately 1Tb of telemetry data. While on its own, 1 TB of data is very manageable by conventional DB techniques; consider an airline company’s dilemma if they have

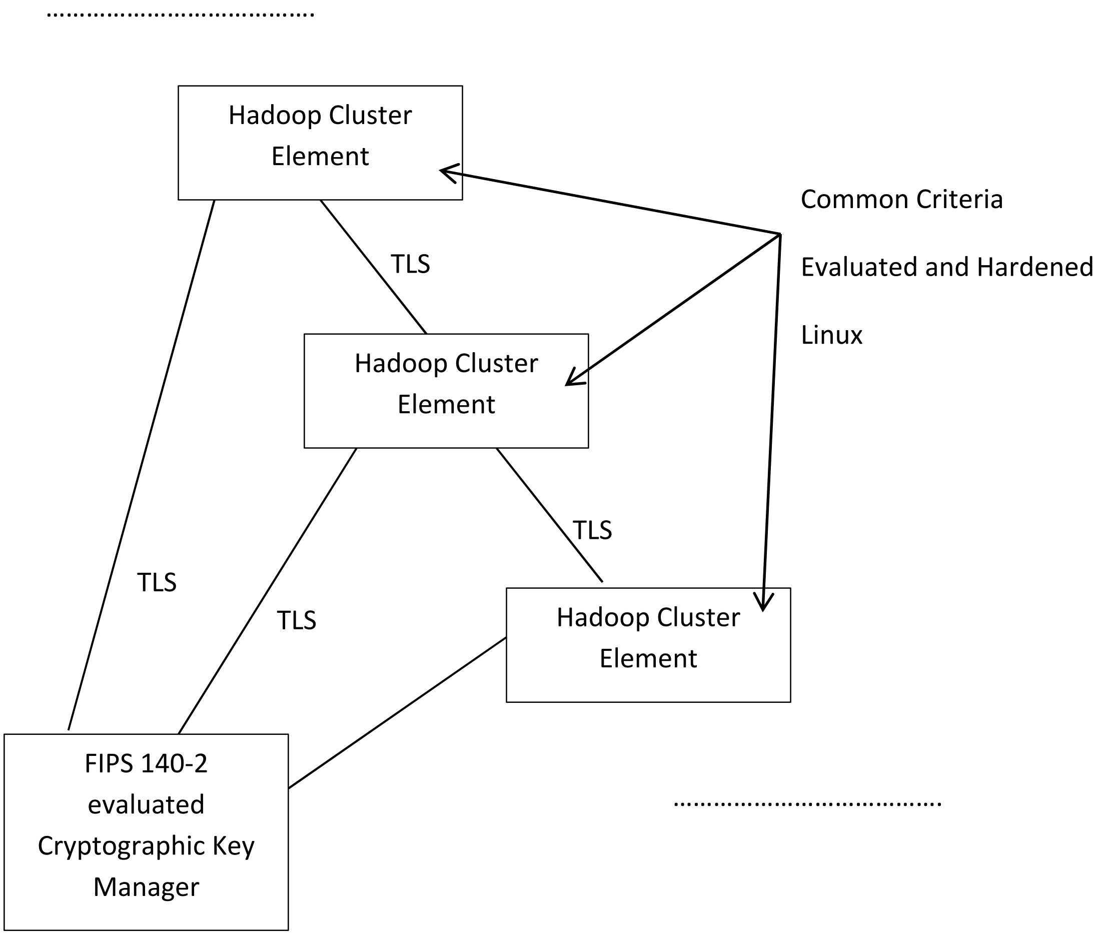

Figure 2. A fragment of a Hadoop cluster implementing robust security concepts.

even 25 787 flights a day!!! Suddenly that's 25 TB of data each day that must be managed, which puts that airline operator in the "Big Data" league!

So the question is, what will a company gain by spending the dollars to implement a "Big Data" approach to telemetry data management? It is unclear. Planes, rockets and spacecraft don't fall out of the sky every day; thankfully, so one could make the argument that the current approach of aggregate, analyze, sift, re-analyze, dump, works fine. However, it is inevitable that a hybrid approach will emerge (see Figure 3) in which a data "dump" to clear out the primary analysis database, will move that data over to a Hadoop cluster. Once there, creative analysts will have an unprecedented opportunity to leverage MapReduce techniques on data sets unprecedently large even for the telemetry industry.

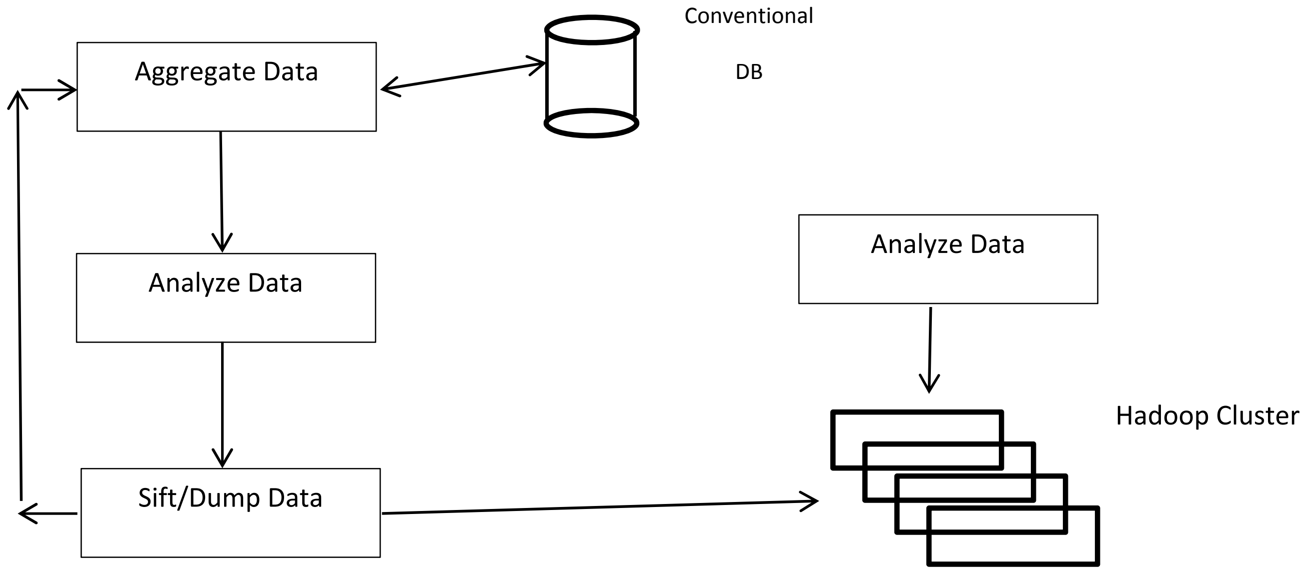

Figure 3. A proposed hybrid telemetry post-processing analysis framework utilizing Hadoop. The "conventional" paradigm (left) is preserved, all expunged data goes into the Hadoop cluster for additional analysis.

WRAPPING UP

Hadoop technologies have revolutionized the way enterprises may manage "Big Data." While the security challenges remain the same when compared to conventionally (database) stored data; security implementation will be even more difficult. The schedules and safety concerns regarding flight (and terrestrial) vehicle safety will always dictate that some subset of a vehicle's telemetry product will always be managed with a traditional database warehouse technology. However, Hadoop will open the possibility for potentially all vehicle telemetry data to be "online and analyzable;” albeit at a slower speed. Such “low and slow” analysis may give test and safety engineers new insights into systems previously thought “understood.”

REFERENCES

[1] O'Reilly Radar Team, **Big Data Now**, O'Reilly, August 2011.

[2] White, Tom, **Hadoop: The Definitive Guide**, O'Reilly, May 2012.

[3] National Institute of Standards (NIST) Information Technology Laboratory, **FIPS 140-2, Security Requirements for Cryptographic Modules**, US Government Printing Office, May 2001.

[4] International Standards Organization (ISO), **Common Criteria for Information Technology Security Evaluation,** ISO, August 2005.