What IT Professionals Need to Know About Hadoop in National Security Missions

Bob Palmer | Senior Director, SAP National Security Services™ (SAP NS2™)
June 2012 | www.SAPNS2.com

Despite the funny name, Hadoop is not the latest music group or iPhone game. It *is* one of the hottest new technologies for managing very large data sets, which is a growing challenge in large commercial operations such as Yahoo!, eBay and Amazon, as well as in the U.S. Government, especially the Intelligence Community and Department of Defense.

Consider that:

- From the beginning of time to 2003, humanity created an estimated 5 exabytes of information, but the same quantity of data is now created every 12 hours.
- More than 2.1 billion people are active users of the Internet, and that number is still climbing at a healthy pace every year.
- The number of sensors, cameras and machines that are networked and producing data for business and mission uses is skyrocketing.

How can Hadoop help? What are its strengths and limitations, especially in national security missions? This paper[1] seeks to provide readers with a basic understanding of:

- The architecture of Hadoop, in plain terms;
- Which features and benefits are driving its adoption;
- Misconceptions that may lead to the idea that Hadoop is the "hammer for every nail;" and
- How SAP solutions, delivered by SAP National Security Services (SAP NS2), can extend and complement Hadoop to achieve optimal mission or business outcomes.

What Is Hadoop?

Hadoop is a "distributed file system," not a database. Hadoop manages the splitting up and storage of large files of data across many inexpensive commodity servers, which are known as "worker nodes." When Hadoop splits up the files, it puts redundant copies of the chunks of the file on more than one disc drive, providing "self-healing" redundancy if a low-cost commodity server fails. Hadoop also manages the distribution of scripts that perform

[1] This paper is a condensed version of "Hadoop: Strengths and Limitations in National Security Missions," available at http://www.sapns2.com/news/white-papers-videos.html.

business logic on the data files that are split up on those many server nodes. This splitting up of the business logic to each of the CPUs and RAM on many inexpensive worker nodes is what makes Hadoop work well on very large "Big Data" files. Analysis logic is performed in parallel on all of the server nodes at once, on each of the 64MB or 128MB chunks of the file.

Hadoop software is written in Java and is licensed for free; it was developed as an open-source initiative of the Apache Foundation.

Companies such as Cloudera and Hortonworks sell support for Hadoop; they maintain their own versions and perform support and maintenance for an annual fee based on the number of nodes the customer has. They also sell licensed software utilities to manage and enhance the freeware Hadoop software. Thus, we can say that Cloudera and Hortonworks are to Hadoop as Redhat and SUSE are to Linux.

The Secret Sauce: MapReduce

The exploration or analysis of data in Hadoop is done through a software process called MapReduce. MapReduce scripts are written in Java by developers on the "edge server." The MapReduce engine breaks up jobs into many small tasks, run in parallel and launched on all nodes that have a part of the file; this is how Hadoop addresses the problem of storage and analysis of Big Data. Both the data and the computations on that data are in the RAM memory of the node computer.

The *map* function of the MapReduce engine distributes the logic that the developer wants to execute on a given data file to each of the CPU processors that has a small chunk of that file. The *reduce* function takes the raw results of the mapping function and performs the business logic computations desired against them: aggregation, sums, word counts, averages, *et cetera.*

Scalability is simple to achieve; if the CPU or RAM memory is being taxed by the MapReduce scripts, just add more inexpensive commodity server CPUs and disc drives.

Perceived Business Benefits

The attractions of Hadoop are similar to those of other open source business drivers. The procurement process is simpler, without the need for capital expenditure, because of the absence of initial software licensing costs. Paid support from vendors like Cloudera and Hortonworks can be an operations and maintenance (O&M) budget expenditure. The skilled labor needed for MapReduce scripting (and updating those scripts to reflect changing end-user demand) may be seen as a sunk cost if a large employee base of developers already exists in a client's shop, or amongst the system integrator staff. Aside from the initial cost, the freedom and flexibility of deployment unconstrained by licensing considerations makes Hadoop attractive to customers.

Hadoop's Limitations

Since Hadoop is a file system and not a database there is no way to change the data in the files.[2] There is no such thing as an "update" or "delete" function in Hadoop as there is in a database-management system, and no concept of "commit data" or "roll-back data" as in a transactional-processing database system. Hadoop is a "write-once, read-many-times" affair.

[2] There are schemes which can append log files to Hadoop files to emulate the deletion or changing of data elements in Hadoop files. On query, the log file is invoked to indicate that the result of the MapReduce should not include a given element.

Therefore, Hadoop is best used for storing large amounts of unstructured or semi-structured data from streaming data sources like log files, Internet search results (click-stream data), signal intelligence, or sensor data. It is not as well suited for storing and managing changes to discrete data elements that have precise meta-data definitions. In order to do analysis or exploration of a file in Hadoop, the whole file must be read for every computational process because by its nature there are no predefined data schema or indexes in Hadoop.[3] Hadoop is essentially batch-oriented; a developer builds a MapReduce script to look at a one- or two- terabyte file, and he expects it to run for 20 minutes to an hour or longer. The idea of Hadoop is to be cheap enough per terabyte that you can store all of the raw data,[4] even data that currently has no anticipated uses.

Since Hadoop is built to accommodate very large data files by splitting them up into small chunks over many worker nodes, it is not a good way to handle large numbers of smaller files. Nor will Hadoop return query results in the sub-second response times that business intelligence users expect for queries based on well-structured and indexed data warehouses.

The Scalability Problem of Hadoop: It's Not in the Software

Mission-driven requirements for data analysis are not static, because the enemy is not static. This creates a need to constantly revise the analysis logic in MapReduce to create evolving views of the data set. Highly skilled data scientists are essential in this construct, as is the need for accurate and frequent communications between the non-technical subject matter analyst and the data scientist. This could produce a bottleneck in data analysis that has nothing to do with data-node scalability, as the limited number of data scientists may be unable to keep up with the ever-changing needs of end-users[5].

Optimizing Mission Performance Using a Hybrid Solution

To the extent that end-user requirements to analyze data can be pushed to a more self-service front-end, the need for IT involvement to create new MapReduce scripts can be constrained. This can shorten the time between end-user requirements and fulfillment, resulting in a more agile, mission-responsive analysis process.

We propose that Hadoop could be used as a catch-all collection point for continuously generated data. Then, instead of executing the iterative analytical processes in the Reduce scripts, MapReduce could be used in an ETL (Extract, Transform and Load) process to load an "analytical data repository" with a very large domain of data that would be likely to contain the information needed for a given mission or business function. In addition, by empowering the non-IT analyst to create his or her own new analyses in a self-service, front-end visualization tool, the workload on the highly skilled data scientists can be managed more optimally.

The SAP Sybase columnar RDBMS is uniquely appropriate as an analytical data repository for the result of a MapReduce ETL, because there is no bottleneck in reporting performance for even extremely large data sets likely to be extracted for the data domain needed for the mission. Sybase IQ by its nature is "self-indexing", and

[3] Hadoop is "schema-on-read."

[4] For example, anecdotal rumor has it that a large telecommunications vendor couldn't figure out what caused a system-wide outage for several days because the log files were on too many different servers, and they could not analyze them together. Now they are collecting all the logs to one large Hadoop system.

[5] Total-cost-of-ownership discussions about Hadoop should include this labor it takes to write and constantly update the MapReduce scripts as driven by ever-changing end-user demands.

the columnar vs. row-based storage of data means that the "seek time" response is much faster, even when analyzing the very large data sets that may be the result of a MapReduce script.

SAP BusinessObjects could then be used to analyze and visualize that result to make it actionable for the mission, in a graphical and flexible user interface. This would empower analysts to get different analytical views and juxtapositions of the data elements—with sub-second response--without the need to go back to the IT shop for a revision to a Reduce job, or a new Java report. Google-like keyword searches of the data in the analytical repository is provided in the BusinessObjects system, as is the ability to create new reports, visualizations and dashboards without IT skills.

The capability of Sybase IQ to serve as an analytical repository with sub-second response times, combined with SAP BusinessObjects solutions' self-service graphical interface, means that organizations can be more agile and responsive to never-ending demands for new analyses. And if desired, SAP Sybase mobility solutions can enable the secure delivery of analyses to mobile devices in the field.

The analysis repository can also be the nexus between Hadoop data and other systems in the enterprise, giving users the ability to seamlessly present analyses of data from heterogeneous systems, data warehouses, and transactional applications.

Summary

With this paradigm, SAP NS2 can help clients optimize mission performance by leveraging the best of both worlds: the ability to store *all* incoming data without regard to size or structure in a cost-effective Hadoop file system; and then the use of MapReduce to create data domains that can be delivered to non-technical end-users for analysis with *ad hoc* flexibility and sub-second response times using SAP's industry-leading solutions for reporting and analysis.

Author:

Bob Palmer
Senior Director, SAP National Security ServicesTM (SAP NS2TM)
bob.palmer@sapns2.com
301.641.7785