Enterprise Historian for Efficient Storage and Analysis of Industrial Big Data

Ernest C. Puig
Software Engineer
GE Power & Water
4200 Wildwood Pkwy
Atlanta, GA 30339
ernest.puig@ge.com

John Interrante
Computer Scientist
GE Global Research
1 Research Cir
Niskayuna, NY 12309
interran@ge.com

Kareem S. Aggour
Senior Engineer
GE Global Research
1 Research Cir
Niskayuna, NY 12309
aggour@ge.com

Jie Lu
Senior Engineer
GE Intelligent Platforms
325 Foxboro Blvd
Foxboro, MA 02035
jie.lu@ge.com

Eric Pool
Consulting Engineer
GE Power & Water
4200 Wildwood Pkwy
Atlanta, GA 30339
eric.pool@ge.com

Mina Botros
Software Engineer
GE Intelligent Platforms
325 Foxboro Blvd
Foxboro, MA 02035
mina.botros@ge.com

ABSTRACT

With a massive increase in time series sensor data being generated by an ever-growing number of industrial equipment such as gas and steam turbines, new systems are required to store and analyze this machine-generated "Industrial Big Data." GE Global Research, GE Intelligent Platforms, and GE Power & Water designed and built Enterprise Historian to address this challenge with the goal of enabling GE Power & Water to perform deep historical analysis and data mining on terabytes of time series sensor data. Enterprise Historian is built on top of Apache Hadoop, an open-source framework for executing batch analytics on big data sets in a distributed computing environment. An instance of Enterprise Historian has been deployed on a 48-node, 280TB Hadoop cluster at GE Global Research and loaded with 3 years of Thermal RM&D sensor data. With Enterprise Historian, analytics that used to take weeks to months to run can now be executed in minutes to hours. Enterprise Historian has also been commercialized by GE Intelligent Platforms as Proficy Historian HD.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*distributed systems*; G.3 [**Probability and Statistics**]: Time series analysis

General Terms

Experimentation, Performance

Keywords

enterprise historian, industrial big data, remote monitoring and diagnostics, time series sensor data

1. INTRODUCTION

GE's installed base of industrial equipment, including gas turbines, aircraft engines, and medical imaging devices, can be found all over the world. Each piece of equipment contains sensors that continuously generate time series data for monitoring and diagnostics purposes. For a large percentage of that equipment, a subset of their data is sent to a GE Remote Monitoring and Diagnostics (RM&D) Center to proactively monitor for customers that have a CSA (customer service agreement) and automatically detect maintenance needs. As ever-smaller sensors become ubiquitous and network connectivity improves, these pieces of equipment are incorporating increasingly larger sets of sensors and sending data at increasingly faster rates. In order to support and take maximal advantage of this growing Industrial Big Data, it is critical that the infrastructures at GE's RM&D centers be able to efficiently store and analyze quickly growing volumes of time series sensor data.

Historically it was possible for a single machine or a handful of machines to receive and store much of this data. However, the growth in sheer volume of devices and sensors coupled with the desire to perform rapid data mining on larger amounts of historical data (requiring that they be kept in memory or on disk) requires development of systems with new technical approaches to address these challenges. Enterprise Historian was designed and built by GE Global Research, GE Intelligent Platforms, and GE Power & Water to address this challenge with the goal of enabling GE Power & Water to store and analyze tens to hundreds of terabytes of time series sensor data from gas and steam turbines at its Thermal RM&D center.

Historians are special-purpose database applications designed to efficiently store and analyze large quantities of time series data with extremely fast read and write performance. This enables historians to store anywhere from thousands to millions of data points per second, capturing streams of data in real-time from sensors located across a manufacturing facility, power plant, or other such sensor-rich environment. While very efficient, most of the historians in the marketplace are single server solutions, thus limited by the memory and disk capacities of a single machine. Our objective was to design and develop a next-generation historian solution capable of efficiently storing and performing rich historical analysis on hundreds of terabytes of time series data. Such a Big Data-enabled "Enterprise Historian" system would be nearly seamlessly scalable in how much data could be stored and analyzed, moving from months of data stored on a single server to decades of data stored on a cluster of low-cost commodity servers.

The Enterprise Historian system was built to complement Proficy Historian^{TM}, an existing operational historian from GE Intelligent Platforms, with the ability to store Proficy Historian archive files in a Hadoop-based, distributed environment for massive storage and parallel processing. Apache Hadoop^{TM} is an extremely popular open-source framework for executing batch analytics on big data sets in a distributed computing environment due to its near-linear scalability and built-in fault tolerance. Because of how naturally Enterprise Historian complements Proficy Historian, GE Intelligent Platforms is making Enterprise Historian commercially available outside of GE under the moniker Proficy Historian HD [7].

An instance of Enterprise Historian has been deployed on a 48-node, 280TB Hadoop cluster at GE Global Research with 3 years of Thermal RM&D sensor data loaded from the years 2004, 2006, and 2011. With this instance, analytics that used to take weeks to run can now be executed in minutes. In practice, such analytics were very difficult to execute in the operational historian data store because it would have to slow down its processing and analyses of the recently arriving time series data points in order to run analytics on the older historical time series data. Other instances of Enterprise Historian now exist within GE Intelligent Platforms and GE Software as well, but this paper discusses the instance set up at GE Global Research.

This paper is organized as follows. Section 2 gives a background on Big Data in general, and highlights some common Big Data management and analysis challenges faced by most organizations. Section 3 describes related work in the space of time series data management, and in particular big time series data. Section 4 provides an overview of the GE Power & Water Thermal RM&D Center and why Enterprise Historian is needed. Section 5 outlines the specific use cases for which the system was designed and tested, followed by Section 6 which describes the Enterprise Historian software architecture. Section 7 describes the hardware infrastructure used for performance testing, and Section 8 details the performance results for the use cases described in Section 5. Section 9 then describes the impact of the system. Finally, Sections 10 and 11 describe our future vision for Big Data management for RM&D and presents general conclusions on the system.

2. BIG DATA BACKGROUND

Big Data can be most succinctly described as data that is too big, moving too fast, and/or too poorly structured to be stored and analyzed in its entirety through traditional computing approaches. In order to extract meaning and value from Big Data, new systems are required to handle the challenges posed by the volume, velocity, and variety of these big data sets.

While Big Data challenges are not particularly new (especially to businesses such as GE), highly scalable and fault-tolerant software systems that enable the cost-effective storage and analysis of big data sets are. Historically, data sets that were too large to analyze were typically stored on disk for a period of time, and eventually were then rolled off of disk and onto a tape archive for long-term storage, with very little chance of ever being used again. This write-once-read-never strategy for handling Big Data produced a lot of tape back-ups (and commensurate storage costs), but did not produce a lot of value from that data. Hardware advances from reduced disk and memory costs to increased network bandwidth, coupled with software improvements from improved cluster management software to the development of Big Data systems, have come together to support the handling of increasingly large volumes of data that is growing and changing at increasingly faster rates. Replacing the traditional archive-and-ignore model of data management, together these hardware and software advances are ushering in a new era in which all data for all time is online and available for analysis and mining, giving researchers and data scientists access to larger, faster data sets than realistically possible before.

2.1 Hadoop Background

Apache Hadoop^{TM} was originally created by Doug Cutting and Mike Cafarella [16]. They took the best practices of Nutch, their web search engine project (now an Apache project), and combined it with ideas from articles released by Google on their Google File System [8] and MapReduce [5]. The success of Hadoop has been staggering, with a market research institute predicting Hadoop will be worth $813 million by the year 2016 [4].

Hadoop enables analytics to be executed in parallel across potentially thousands of machines (which are called nodes when they are assembled into Hadoop clusters). Hadoop is designed to run on clusters of low-cost, commodity servers and is comprised of two main components. The first component is the Hadoop Distributed File System (HDFS), an infrastructure for storing files across a cluster of machines [20]. Files are chunked into blocks and each block is stored with a replication factor (3x by default), meaning multiple copies of each block are stored on different nodes in the cluster for fault tolerance. Because Hadoop is designed to run on commodity servers, it is designed to handle node failures without losing data or canceling jobs.

The second component of Hadoop is the MapReduce parallel computing paradigm [5], consisting of two stages. The Map stage involves dividing work across the cluster of nodes, performing initial processing of the data local to each node, and then shuffling the data so that related data points are grouped together on the same node for further processing. The Reduce stage then executes further operations on the logically grouped data, typically producing an aggregated result. If an individual execution fails on any given node due to hardware or other failures, it is automatically rerun on a different node in the same cluster. Hadoop guarantees that even if some nodes in the cluster fail, the execution will still complete and the engineer can have confidence in the results.

Together, the distributed storage provided by HDFS combined with the parallel computing paradigm of MapReduce makes Hadoop a highly scalable and fault-tolerant storage and computing platform. For many applications, Hadoop is nearly linearly scalable; meaning that doubling the performance of a Hadoop cluster requires simply doubling the number of nodes in the cluster. For example, if a typical Hadoop job takes 2 minutes to run on 10 nodes, it will take approximately 1 minute to run on a comparable 20 node cluster. This near-linear scalability, coupled with the built-in fault tolerance, has made Hadoop a popular platform for executing batch analytics on massive quantities of data.

Today, Hadoop provides the backbone of Facebook, Twitter, eBay, Amazon, Google, IBM, LinkedIn, and more with several vendors supporting their own commercial distributions of Hadoop including Cloudera, MapR, Hortonworks, Intel, and Pivotal to name a few [16, 1].

With Hadoop, businesses may now process huge volumes of unstructured data sets in a relatively cost effective and timely manner. Instead of representative samplings of their data, companies can now process any and all data associated with their business problems [11].

2.2 Big Data Management and Analysis Challenges

With Big Data, there are three core challenges that should be addressed in tandem: 1) data storage, 2) data management, and 3) data processing/analysis.

Today, unstructured data comprises approximately 80% of all data [2]. Hadoop and similar solutions attempt to address the challenges posed by unstructured data by providing a single infrastructure that can enable both long-term storage and analysis of big data sets.

With Hadoop, organizations can solve the data storage challenge by purchasing more disks [2]. From an analysis perspective, many efforts are underway to augment existing analytic systems (e.g., SAS, R, Matlab, etc.) to enable them to execute on Hadoop-based Big Data, where it makes the most sense financially and from a business operations standpoint [11].

According to McKinsey & Company, the United States alone is likely to "face a shortage of 140,000 to 190,000 people with deep analytical skills as well as 1.5 million managers and analysts with the know-how to use the analysis of big data to make effective decisions' by 2018" [11]. Thus, developing novel Big Data systems is not enough; universities and companies must begin training students and employees today to take full advantage of these scalable systems being developed to run analytics on Big Data.

Unquestionably, the biggest Big Data challenge facing GE is from Industrial Big Data. GE has a large installed base of industrial equipment, a significant portion of which is monitored through sensors in real or near real-time. The industrial equipment being built and sold today is equipped with sensors that capture an order of magnitude more data than the equipment put into production just a few years ago. And with many of the assets in GE's installed base sending data for remote monitoring and diagnostics, the time series sensor data GE manages is already significant and growing rapidly [12].

3. RELATED WORK

The vast majority of the Big Data generated today is Industrial Big Data. According to McKinsey & Company, "manufacturing stores more data than any other sector—close to 2 exabytes of new data stored in 2010" alone [15]. This number is impressive for a single sector and speaks to the massive volume of Industrial Big Data being generated when one recognizes that manufacturing is just one of many sectors that generates time series sensor data. Other such sectors are as diverse as financial trading markets, industrial equipment monitoring and diagnostics, and patient health monitoring, to name a few. Despite its size and significant growth potential, big time series data has not received nearly the attention that other Big Data challenges have, however.

Some approaches to time series storage and analysis involve embedding time series data into relational or columnar data stores. And while some columnar stores in particular have proven to be reasonably effective at storing time series data, they often are suboptimal for querying time series data. As outlined by Shafer et al at Carnegie Mellon University [19], one of the many reasons is that generic data stores do not by default store time series data in time sorted order, and instead require the data to be sorted as part of every query executed on the system. This aspect of database storage can significantly limit the read performance of a traditional database storing billions of time series data points.

There are a suite of operational historian solutions specifically optimized for time series data in the commercial marketplace, including GE's own Proficy Historian product from GE Intelligent Platforms [6], which give both very high read and write performance. Beyond their high read and write rates, these operational stores are further enhanced to reduce the data footprint, utilizing proprietary file formats that give very high compression rates on time series data. However, these traditional historian solutions are single-server operational data stores that are not natively distributed and are thus not easily scalable once the volume of data exceeds the capacity of a single machine.

Beyond the operational historian market, there has been a body of research specifically focused on Big Data-enabled storage and processing of time series sensor data similar to our Enterprise Historian solution. Pure Big Data time series solutions include the Open Time Series Database (also called OpenTSDB) built on top of HBase [21]. OpenTSDB is a distributed system built primarily for storing and graphing time-stamped data, particularly log file data, and is not built to enable meaningful time series analytics on the data. KairosDB is a recent rewrite of OpenTSDB built on top of Cassandra, with its first beta release in April of this year [10]. At least two commercial solutions also exist. TempoDB provides a database service for storing and analyzing time series data, but with a maximum resolution of 1ms [3]. Seeq is a Seattle-based start-up focused on building Big Data platforms for industrial process data [22]. In contrast to some of those systems, Enterprise Historian merges the time series optimizations available in traditional historian products with the near-linear scalability and fault tolerance supported by today's Big Data systems.

4. THERMAL RM&D OVERVIEW

GE Power & Water's Thermal RM&D Center can monitor over 1,500 turbines globally, receiving a real-time stream of time series data from approximately 2.8 million unique sensors across those assets worldwide. Data points arrive from the global assets at an average rate of 5 points per minute per sensor for an average total of 100K data points/second (some data points may already have been compressed by intermediate Historians). This data can be collected in a central time series store and made available for monitoring and diagnostics analytics. Typical analytics currently run in production at the Thermal RM&D Center may use anywhere from a single recent time series point to several days' worth of values from a particular unit, and may calculate items such as corrected values, aggregated metrics, or performance characteristics for a given unit. Analytics results are often stored back to the time series data store as derived data and may serve as inputs to other calculations. Enabling fast execution of these analytics while retrieving years of time series data for deep historial analytics at the same time can allow Thermal RM&D to quickly detect potential issues with gas turbines and related equipment in the field and to make quick corrections to improve turbine performance, to name a few capabilities.

The typical model for RM&D solutions for industrial machinery is to process data locally (at the location of the equipment) and only send a reduced subset of the data to the central infrastructure. This traditional site-based model is very effective for a small number, for example tens, of sites. It reduces network bandwidth, provides access to real time data streams from the machinery source, and lowers central storage costs. However, this model does not scale well to hundreds or thousands of sites. It is challenging to support a large number of independent remote hardware and software systems at customers' sites while ensuring that they are current and equivalent. Access to the equipment is often difficult, as the hardware is typically required to reside on the customer network to which remote access is often restricted. Because of the many independent installations, there is limited ability to continually add capability and leverage the latest data storage and analytics technologies. Most importantly, there may be no straightforward capability to perform fleet level analytics since the data for different assets resides in separate locations.

With the advent of Big Data and Cloud technologies, a near real-time response from a cloud implementation has become feasible and cost effective. Such a solution offers multiple important advantages:

- Reduced complexity of onsite components, moving focus from analytic execution to data collection and transport
- Reduced management and support costs of onsite hardware and software
- On-demand elastic and scalable compute infrastructure that can grow with the installed base of industrial equipment, allowing for better return on investment (ROI)
- Enabling fleet-wide analytics, as well as continuous improvement processes with the ability to measure and understand data quality issues rapidly
- Ability to modify analytics in minutes rather than days or weeks
- Enhanced support model with analysts having direct access to the raw equipment data for deeper data mining, leading to faster problem resolution

One of the primary drawbacks to a centralized solution is network bandwidth costs associated with streaming large amounts of data from each site to the central location. This impact on bandwidth requirements can be reduced by implementing the RM&D services and resources regionally.

As shown in Figure 1, the GE Power & Water Thermal RM&D Center utilizes such a centralized architecture. Onsite operational historians collect time series data in real-time and transmit that data in real or near real-time to the centralized RM&D Center's data infrastructure. Analytics

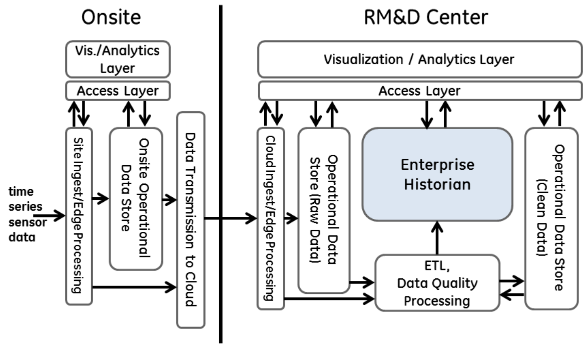

Figure 1: Centralized Architecture for RM&D

and visualizations are executed against the centralized operational historian for near real-time analysis. As data ages it is moved into the Enterprise Historian system for deep historical analysis and data mining.

5. THERMAL RM&D USE CASES

The ability to operate against historical machinery data is important to all aspects of RM&D and machinery life cycle management. Through the support of Hadoop and MapReduce, all historical machinery data is made available to massive parallel operations by equipment experts and data scientists. As data sizes have exploded, the traditional model of extracting data from a database and downloading it for desktop analysis is no longer feasible. The objective of Enterprise Historian is to improve the overall capacity and speed with which the various users can learn from time series data.

The following monitoring and diagnostics use cases have been used to evaluate the performance of Enterprise Historian as compared to the current GE Power & Water Thermal RM&D production environment. Although we have limited the size of the input data for those use cases in order to compare the performance of Enterprise Historian to the current production historian environment more fairly, each analytic can execute over the entire fleet at the same time if given enough input data. The use cases are as follows:

5.1 Plant Operations Support

5.1.1 Automated Data Requests

Compared to the other three use cases below which are mainly focused on testing the integration of Hadoop and Historian technologies, this exemplary use case is aimed at evaluating the Enterprise Historian system under significant load. Several MapReduce queries have been constructed based upon automated data requests (ADRs) we have received from internal customers. The largest such query will request several months of data for one fourth of our entire monitored fleet, and interpolate this data to a time-aligned resolution of five minutes.

5.2 Condition Based Maintenance

5.2.1 *Overspeed Analysis*

Overspeed analysis scans through time-aligned 1-second resolution data from two tags (shaft speed and compressor discharge pressure) to determine if an overspeed condition has ever occurred. The data mining MapReduce job developed for this use case processes sequential raw data points from both tags while performing "on the fly" interpolation between each set of two points for the 1-second values it needs, so it runs much more quickly than the data density analysis job which has to read and count every tag's time series instead of processing just two tags' time series.

5.3 Enterprise Machinery Data Management and Quality

5.3.1 *Data Gap Analysis*

Data gap analysis checks to determine if there are literal "gaps" of data in our time series records. The MapReduce job that has been developed for this use case checks each individual tag to see if the time differential between two subsequent records exceeds the maximum time quantum that has been set by the business. The results file will contain the timestamps of each pair of such records. The result file is then ingested into Thermal RM&D's production systems for reconciliation.

5.3.2 *Data Density Analysis*

Data density analysis is very important to ensure that the on-site equipment, as well as our central operations center, are configured and synced properly. This allows us to determine if we are collecting too much or too little information on one of our assets. This is a MapReduce query that performs a daily count of all records specific to a particular sensor. When queried with multiple assets, a comparison can be made between them to determine if our data collection operations need to be modified. The raw results can be fed into a subsequent process to generate a heatmap for better analysis, although this heatmap capability is not currently integrated within the Hadoop framework.

6. SOFTWARE ARCHITECTURE

Enterprise Historian builds on the strengths of Proficy Historian to develop a scalable big data time series storage and analytic platform as shown in Figure 2. A historian archive file is bounded by size. In the current RM&D setup, these files are approximately 9GB in size. When Proficy Historian creates a new historian archive file to write data to it, it associates a "start time" with the file and will not store any data prior to that time in the current file. Once the size limit is reached, Proficy Historian automatically closes the file, associates an "end time" with the file, and creates another historian archive file to become the current file. The archive importer (also called the ingestion service) copies the Historian archive files (converting them to the latest Historian archive format if they are archives from previous years in older formats) to the Hadoop cluster's filesystem (HDFS) and stores information about each file in a metadata store, a PostgreSQL 9.2 database, such as the file's location within HDFS, its start and end times, and which tags are stored in the file. This information allows the query planner to limit which files are needed to process the user's query. The query planner passes the query and the list of

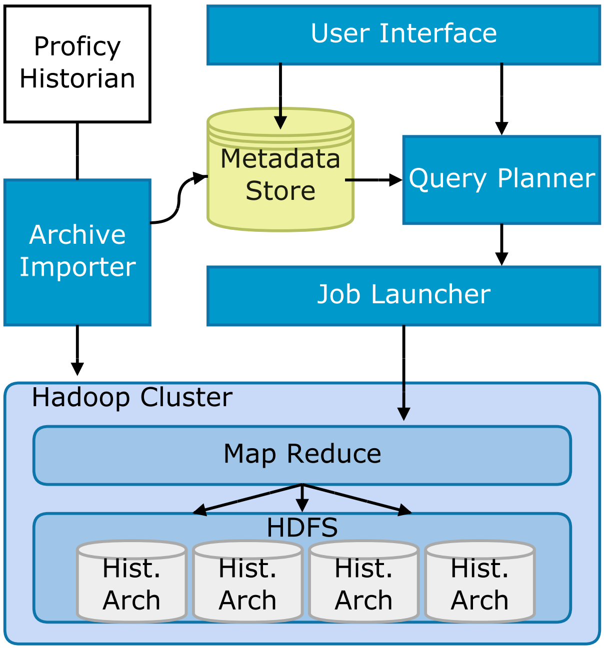

Figure 2: Enterprise Historian Architecture

files needed to process the query to the job launcher, which kicks off a MapReduce job on the cluster of low-cost commodity servers. The MapReduce job runs many tasks simultaneously on large numbers of machines to parallelize the reading and analysis of the historian archive files across each machine in the Hadoop cluster. Each sub-task reads only one part of one historian archive file, allowing large volumes of historical data to be collected, sorted, merged, and analyzed more quickly in parallel.

The MapReduce job invokes two components in series, a Historian Mapper component and a Historian Reducer component with an optional Custom Calculation and Custom Writer. The Historian Mapper reads the time series tuples from the various Historian archive files, groups them into lists by tagnames, and sorts the lists by timestamps. The Historian Reducer then reduces each list to the desired query results using a given query mode (raw, count, average, interpolation, and/or custom calculations) before outputting the query results to HDFS or a custom writer as shown in Figure 3. Queries can specify their own custom calculation and custom writer classes to be invoked by the MapReduce job to perform special computations on the query results or write the results to external non-HDFS stores or special destinations.

Enterprise Historian is designed and built with three core requirements in mind:

1. Scalability—the system must be able to scale in both data storage capacity and analytics performance. That is, adding nodes to the cluster should increase both the amount of data able to be stored and the speed of execution of the analytics (given the same amount of data).

2. Fault-tolerance—all data must be backed up such that

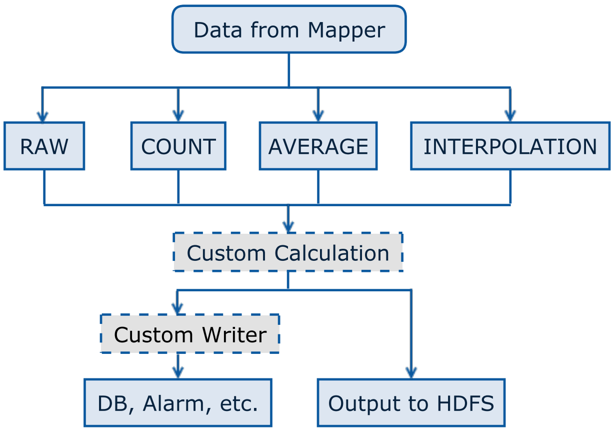

Figure 3: Historian Reducer Component with Custom Calculation and Custom Writer

the loss of up to 10% of the nodes in a cluster at any single time should result in no loss of data.

3. Rich Analytics—both predefined and custom analytics must be able to execute against the data repository, to enable in-depth mining of the historical data stored in Enterprise Historian.

To date, Enterprise Historian has been run on the Standard Edition of Cloudera's Distribution Including Apache Hadoop (CDH) and also on Pivotal HD. At its core, Enterprise Historian is built from Hadoop, a Metadata Store, and a set of Web Services.

6.1 Proficy Historian

Proficy Historian is the operational historian product offered by GE Intelligent Platforms. It features a very efficient storage system for time series data, provides high throughputs for read and write operations, and serves the need to manage short-to-mid-term time series data very well, but is limited by the computing power of a single server at present time. As the size of collected data grows, Proficy Historian will need to hand off data to a long-term storage and query platform, which is Enterprise Historian.

Proficy Historian stores data by time, but also allows users to partition all data into multiple data sets. Each data set has its own stream of archive files organized by time. Partitioning can be based on application, so separate applications can use their own data sets. It can also be based on data characteristics such as collection rates, which can be used to increase data density in archive files and improve the I/O efficiency of queries.

Proficy Historian can be configured to send archive files (closed files for older data or a snapshot of the "current" file) to be consumed by Enterprise Historian. During the process, metadata for archive files is also generated and made available as well. The archive importer, the ingestion service of Enterprise Historian, picks up the archive files and their metadata and moves them along to the Hadoop cluster and a metadata store respectively. Enterprise Historian is designed to accept data from multiple Proficy Historians.

6.2 Data Files and Storage Size Considerations

Different options, in terms of file format, have been considered for storing time series data within Enterprise Historian. One options is to store the native Proficy Historian archive files as is, and another option is to convert them into plaintext files, such as tab separated value (TSV) files. Proficy Historian archive files, compared with their equivalent plaintext files, are highly compressed and more efficient in terms of storage need. Archive files are "splittable" in a sense that they can be broken into smaller pieces and then processed in a parallel manner. Such characteristics make archive files an appealing option for storing large amounts of time series data. It has been found that on average an archive file is eight times smaller in required storage space than its corresponding plaintext file. Since files are typically replicated in the Hadoop Distributed File System, the potential saving in storage is even more significant.

Sensor data used for big data analytics is expected to be stored permanently and new data is added to the data set continuously. Efficiency in data storage on disk is one of the key factors addressed by Enterprise Historian to lower the cost of implementing the solution. Historian archive files require less disk space than alternatives, and this is a key differentiator of Enterprise Historian.

6.3 Hadoop

Within Enterprise Historian, all of the time series data is stored within the Hadoop Distributed File System. Once ingested, Proficy Historian archive files are loaded into HDFS and automatically "chunked" into 128MB blocks, which are replicated into three copies stored on three different nodes, with one node in a different rack than the first two nodes. Enterprise Historian has a standard set of Map operations for parsing the Proficy Historian Archive files and a set of different Reduce operations to suit specific time series analytic requirements. Specifically, reducers have been built within the MapReduce framework to return the raw data, to return interpolated data sets at fixed intervals, to provide minimums, maximums and means over time ranges, and more.

6.4 Metadata Store

To enhance query performance, a PostgreSQL based metadata store is integrated into Enterprise Historian. The store maintains information on each of the archive files maintained within the system, such as the archive file locations in HDFS, the start and end times for the data in each archive file, as well as relations between tags and archive files. For each query job executed against the system, the metadata store is first consulted to identify which archive files are needed to complete the job. This allows the system to avoid scanning every archive file for every query, as would be done otherwise in a default Hadoop implementation.

With information from the metadata store, the query planner can consolidate multiple queries into one MapReduce job and therefore eliminate the need to scan archive files in multiple passes for multiple queries. Per-job metadata generated by the query planner also serves as a guide for map tasks to more efficiently scan input archive files, skipping over large data units that are irrelevant to the queries.

6.5 User Interface and Web Services

Enterprise Historian exposes several RESTful web services

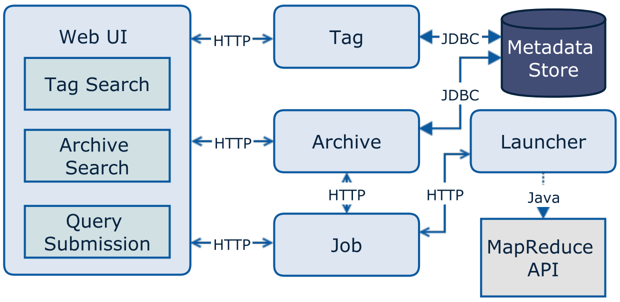

Figure 4: Enterprise Historian Web UI and Web Services

```
<job>
<jobName>Overspeed</jobName>
<inputType>HistorianArchiveFormat</inputType>
<outputPath>/hdfs/output/path</outputPath>
<outputType>text</outputType>
<queries><query>
<name>overspeed</name>
<startTime>01/01/2011 00:00:00</startTime>
<endTime>12/31/2011 23:59:59</endTime>
<mode>INTERPOLATED</mode>
<window>1000</window>
<tagFilterList>some-regular-expression</tagFilterList>
<type>overspeed</type>
<tagMatching>5</tagMatching>
</query>
</queries>
</job>
```

Figure 5: Enterprise Historian XML Job Configuration Document

(Tag, Archive, Job, and Launcher) within its web-based user interface (Web UI) to access the Metadata store and support query execution. These services and the Web UI have been implemented together in a web application run by an Apache Tomcat container. The services can be utilized for read accesses to the Metadata store, while the Web UI itself provides basic functionality for interactive query submissions using web-based forms and job tracking using links to Apache Hadoop's standard web tools. The Web UI also provides the ability to search for tags and find archive files using the RESTful services. Figure 4 shows the one-to-one mappings between the Web UI's capabilities and the RESTful web services that implement these capabilities.

The Web UI is in fact merely a presentation layer on top of the RESTful services. The Job service receives the user's inputs from the Web UI and converts the form parameters into an XML document for processing by the Launcher service. An example of the XML document can be seen in figure 5. The Launcher service takes an XML document representing the query job to be executed and launches the query job using the MapReduce API and configuration information to allow it to connect to the Hadoop cluster. With the RESTful web services as the API, Enterprise Historian can become a platform for larger and more complex applications that execute analytics, track their progress, and query Enterprise Historian for generally useful information.

As stated above, Enterprise Historian job configurations are converted to XML from the various Web UI input fields. This XML content is used to generate the Hadoop MapReduce jobs. The various XML tags are described below:

- **job:** The main XML tag
- **jobName:** Assigns a name to the MapReduce job
- **inputType:** Indicates the format of the input data
- **outputPath:** File path in HDFS to write out results
- **outputType:** Specifies the format of the output data
- **queries:** The start of one or more specific queries
- **query:** The start of an individual query
- **name:** The name of a particular query. Appended to the end of the output path folder specification
- **startTime:** The beginning time of the input data to query
- **endTime:** The end time of the input data to query
- **mode:** Specifies the exact "Historian" function
- **window:** Used when INTERPOLATION mode is set. Specifies in milliseconds the data resolution to interpolate down to.
- **tagFilterList:** A comma-separated list of all assets to query against
- **type:** Specifies a particular advanced analytic mode
- **tagMatching:** Specifies if the tagFilterList contains regular expressions, or literal strings.

7. HARDWARE INFRASTRUCTURE

GE Global Research has provisioned a cluster of commodity servers for testing Enterprise Historian in a preproduction staging environment. The cluster specifications can be seen below:

- 48 HP ProLiant DL380 G8 nodes (1 name node, 47 data/task nodes), each with:
- 2 Intel Xenon E5-2690 @ 2.9 GHz sockets with hyperthreading enabled
- 32 CPUs (2 sockets × 8 cores/socket × 2 threads/core)
- 384GB RAM (on each node; was supposed to be 96GB per node)
- Red Hat Enterprise Linux 6.3 with Linux kernel 2.6.32-279
- CDH 4.2.1 (Cloudera's Distribution Including Apache Hadoop)
- 280TB total disk space for HDFS storage

8. PERFORMANCE RESULTS

Below are the performance results that we obtained from our Hadoop staging environment.

Table 1: Traditional Historian vs. Enterprise Historian Performance Results

Analytic	Duration	Single / Fleet	Input Data (GB)	Output Record Count	Cycle Time (Traditional Historian)	Cycle Time (Enterprise Historian)
Perf ADR (82 Tags)	3 Months 5-min Interpolated	Single Asset	611	10,628,640	15 minutes	3 minutes
Vib ADR (44 Tags)	3 Months 5-min Interpolated	Single Asset	611	7,437,120	18 minutes	4 minutes
Fleet ADR (980 tags per asset)	3 Months 5-min Interpolated	1/4 of Fleet	611	714,736,512	2 weeks	112 minutes
Data Gap	1 Month Raw	1/4 of Fleet	121	2,340,417	Not practical to run	28 minutes
Data Density	2 Months Raw	1/4 of Fleet	300	47,401,042,413	Not practical to run	105 minutes
Overspeed	1 Year 1-Sec Interpolated	1/4 of Fleet	1.5 TB	9,383	Not practical to run	9 minutes

8.1 Side-by-Side Comparison of Traditional vs. Enterprise Historian Approaches

Table 1 is a performance comparison of our use cases between our traditional historian and Enterprise Historian environments. The use cases executed are a representative sampling of the type of queries that are executed on a regular basis in our traditional historian environment. The range of input data for these use cases differ greatly, from as little as 121 GB to 1.5 TB of compressed raw data. The size of the input data is invariant whether a single asset or 1/4 of the fleet is being queried (it changes only when more months are queried, or when the entire fleet is queried instead of one fourth). It should also be noted that some of the input data sets hold only 3 months of our total data collection operation. Others contain as much as an entire year's worth of data. In addition to this, since we cannot perform experimental studies in the Thermal RM&D production system without impacting 24x7 support of customers, the execution times for the traditional historian environment have been normalized from actual jobs to match the input characteristics of the use cases executed within the Enterprise Historian system. Here are brief definitions of the use cases detailed above:

ADR Automated Data Request, a standard process for requesting historical data from the system

Data Gap Quality check that looks at every tag and identifies marked and unmarked gaps in the sensor data

Data Density Quality check that performs a daily record count of every tag in the database

Overspeed Data mining analytic that evaluates 1-second resolution data from two tags to determine whether an overspeed condition has occurred

1/4 of the Fleet Approximately 375 gas and steam turbine assets

The results in Table 1 tell quite a compelling story. The first three use cases are all "ADR" type of queries. All three were executed over the same three months of data. In addition to this, the first two use cases were executed using the same asset's records. Although one can see the gains in the first two use cases, where we can reduce our cycle times by about 80%, the real performance of this system can be seen when comparing these first two use cases against the third ADR use case. It was mentioned earlier in this discussion that a Hadoop system is near-linear in performance with respect to adding additional nodes and seeing an almost proportional decrease in overall cycle time of a job. This should not be confused with the performance characteristics that can be seen when adding additional assets for analysis to a particular Enterprise Historian job. If there is some base job such as the first use case in Table 1 that takes three minutes to complete, we do not see this same job execute in 6 minutes when an additional asset is queried for that same job (twice the number of inputs). The performance characteristics take on much more of a logarithmic behavior. One of the primary differences between use case three and the first two use cases is that use case three is querying 1/4 of our fleet, rather than just a single asset. If this system's performance was linear with respect to the number of queried assets, and if we assume for a moment that we are querying the same number of tags as we are in use case one (which, we are actually querying for significantly more in use case three), we would expect use case three to take about (3 minutes * 375 assets) / 60 minutes = $\sim$ 19 hours to complete. However, it turns out that use case three takes just under 2 hours to complete. This is because all of the input data contained in these archives that match our given input time range are being processed in the Mapper phase, regardless of the number of assets that are queried. It is the larger queries, like use case three, that make use of all of the input data. The smaller jobs simply ignore the time series data unrelated to the queried assets.

The remaining three use cases have never before been attempted in our traditional environment at the level that these execute within Enterprise Historian. Due to various combinations of input data size and analysis logic, their equivalent jobs in our traditional system would take an impractical amount of time (weeks to months) to execute and would starve our other processes. With Enterprise Historian we can now execute these important use cases over our entire fleet in a more timely and efficient manner.

8.2 Archive vs. Plaintext File Comparison

Table 2: **Archive vs Plaintext Query Performance Results**

File Type	Map Time	Reduce Time	Shuffle Time	Merge Time	Elapsed Time
Plaintext (10%)	0.3833	5.8	8.5167	8.75	16.8833
Archive (10%)	1.1667	5.7833	2.6	2.7833	10.8
Plaintext (50%)	0.9167	30.6333	18.9333	19.1833	56.5167
Archive (50%)	6.1833	30.3667	12.6833	12.85	51.2
Plaintext (100%)	1.5	59.5	30.25	30.45	99.3167
Archive (100%)	12.0833	62.2167	26.2167	26.55	109.6833

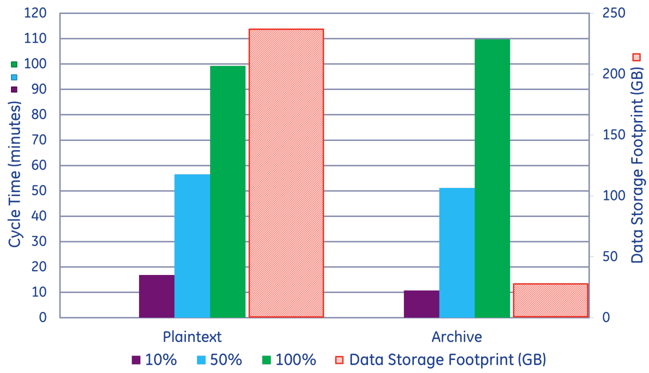

Figure 6: Archive vs Plaintext Query Performance and Storage Chart

As stated earlier, Proficy Historian archive files bring a significant advantage in storage requirement. However, it needs to be studied how the query performance is affected. To answer that question, a number of tests were carried out to compare the performance of querying archive files to their equivalent plaintext files (i.e., the same time series data samples in binary and text formats).

All tests compared the speed of querying three archive files against the speed of querying their equivalent plaintext files. Note that each archive file and plaintext file has information about 209,606 tags containing a total of approximately 1,512,509,499 data samples.

Two kinds of tests were run:

Historian archive processes Proficy Historian archive files and performs timestamp and tag name filtering

Plaintext processes plaintext files and performs timestamp and tag name filtering

A primary difference between archive and plaintext is the use of a plaintext file Mapper as opposed to a historian archive Mapper for reading the time series data.

This test set is composed of three tests, each containing two MapReduce job executions. The first job extracts 462,153,541 samples which represents approximately 10% of the total number of samples in the input data set. The second job extracts 2,322,026,687 samples which represents approximately 50% of the total number of samples in the input data set. While the third job extracts 4,515,471,135 samples which represents approximately 100% of the total number of samples in the input data set.

The input size to the archive jobs is 28.24062 GB, so the jobs used 227 Map tasks to process the input data set. The input size to the plaintext jobs is 236.8456 GB, so the jobs used 1902 Map tasks to process the input data set. The results are presented in Table 2 and Figure 6.

The columns of the results table represent the following:

File Type the format of the input data set, whether plaintext (simple text format) or archive (Proficy Historian archive file binary format).The percentage of the number of samples involved in the query is specified in brackets.

Input Size the total size in GB of the input data set

Mappers the number of Map tasks that were executed to process the input data set

Map Time the average time taken (in minutes) by the individual Map tasks

Reduce Time the average time taken (in minutes) by the individual Reduce tasks

Shuffle Time the average time taken (in minutes) by the individual Shuffle tasks

Merge Time the average time taken (in minutes) by the individual Merge tasks

Elapsed Time the overall time taken (in minutes) by the job to process the input data set and write the output results to HDFS

The tests were run on a four node Pivotal HD DCA cluster, where each node had the following configuration:

- Intel(R) Xeon(R) CPU E5-2660 0 @ 2.20GHz
- 32 cores
- 64GB RAM
- Red Hat Enterprise Linux Server release 6.2 (Santiago)

Each job ran with 20 Reducer tasks.

The results show that if the output ratio (number of samples extracted from archive files divided by total number of samples) is low to median, archive files either outperform or are on par with plaintext files in queries. It is partially attributable to the I/O efficiency associated with compressed input files. Only when the output ratio is higher, does the processing load needed to decompress more samples start to outweigh the I/O efficiency. Nonetheless, the performance penalty in a typical multi-archive query (as shown) is no more than ten percent. Even in the worst case in the study (output ratio at 100 percent in a single-archive query, not shown in this paper), it takes an acceptable thirty percent more time than plaintext.

8.3 Fleet Analyses

By reducing the time to analyze fleet level data sets from weeks to hours, Enterprise Historian will enable the following use cases:

- Calculate historical summary metrics for the entire fleet for all time
- Execute diagnostics rules on the entire fleet for all time
- Calculate historical baselines for the entire fleet

8.4 Operational Analyses

Enterprise Historian will also enable the following engineering analysis use cases:

- Perform data analyses in support of RCAs (root cause analyses)
- Perform historical market analyses
- Perform ad-hoc data mining investigations
- Look for common historial patterns

9. BUSINESS IMPACT

As Thermal RM&D monitors its global fleet of gas and steam turbines, 6-8 terabytes of highly compressed time series sensor data are collected and stored per year. The introduction of Big Data technology is enabling RM&D to provide more timely and accurate analysis and diagnostics of this collected data. Being able to keep over a decade's worth of data online for deep historical mining and analysis is expected to result in direct improvements in the reliability and performance of customer equipment. For example, this data can be used to back-test existing and new analytic models, run online condition-based lifing models across the entire fleet, and provide deep data mining capabilities for outage management.

With Enterprise Historian, these fleet-level data mining efforts are able to run within a matter of minutes to hours, as opposed to weeks to months on a single machine today. Furthermore, the near linear scalability of Hadoop allows this platform to scale out across new hardware as the volume of data grows over time. Thus, the entire 20+ years of Industrial Big Data generated from the installed base of equipment will be able to be stored online and mined on demand, replacing what was in many instances months of significant manual effort to explore much smaller data sets.

The qualitative benefits provided by Enterprise Historian to GE Power & Water's Thermal RM&D Center can be summarized as follows:

- Increased Productivity—reduce decision times and minimize effort spent managing data. Finding cross-fleet patterns and creating/testing rules is very inefficient (weeks to months) and can be reduced to minutes or hours.
- Higher Quality Analytics—data-driven decisions are correct and optimal. Larger data sets can be used for creating rules, thus reducing rule errors on unseen issues.
- Increased Customer Satisfaction—the customer sees our greater speed and consistency.
- Elevated Team Effectiveness—digitized knowledge reduces training needs and increases staff flexibility. Experienced engineers with valuable knowledge can be more productive, and new engineers with less historical knowledge can also be more effective.
- Enabled Growth—digitized knowledge allows for the creation of new digital products and services.

The value of exposing machinery data to massively parallel analysis and data mining operations are expected to lead to significant amounts of productivity for GE and even more significant value to GE's customers through better management and operation of their equipment (avoiding unplanned downtimes). To date, the team has filed 15 patents on the Enterprise Historian system and adjacent technologies.

10. FUTURE OUTLOOK

Today's software infrastructures for RM&D are moving in at least two distinct directions. On one end, high throughput infrastructures are being built for near real-time to true real-time analysis, using technologies such as Data Distribution Service (DDS) [17] for fast data movement and Complex Event Processing (CEP) for stream processing [14]. On the other end, Big Data infrastructures are being built for storage and batch processing of extremely large volumes of data sets.

As the needs grow for near real-time to true real-time analysis, it will no longer be acceptable for these two infrastructures to be kept in distinct silos. Fast processing capabilities and Big Data storage capabilities will have to be merged into one hybrid system [9]. Through multiple different research efforts, progress has been made towards building such a hybrid fast big data platform. Efforts such as Impala from Cloudera [13], Storm from Twitter [23], Spark from the Berkeley AMP Lab [24], and most recently HAWQ from Pivotal [18] have each been meaningful, although very different, attempts in this direction. Impala and HAWQ both focus on enabling near real-time SQL-like querying of data in Hadoop. Storm is meant as an infrastructure for Big Data complex event processing, and Spark is meant as an in-memory infrastructure to speed-up the performance of traditional Hadoop jobs. Without in-memory technologies, I/O bottlenecks may forever plague Big Data applications, limiting their real-time performance. While one could argue that solid-state drives (SSDs) are a viable alternative due to their considerably faster performance than traditional disk drives without the volatility of RAM, their latencies may still be orders of magnitude slower (microseconds) than even average RAM speeds (nanoseconds).

Looking forward, Big Data architectures may be a tightly coupled combination of a CEP-like engine, an operational in-memory historian, and a disk-based Hadoop store, integrated to achieve the goals of real-time analytic execution on high velocity Big Data. It can be envisioned that a high-throughput CEP-style framework capable of processing thousands to millions of data points per second will be used to capture data sent from equipment sensors, immediately performing operations such as data cleaning, transformations, and simple analytics. The data should then be ingested into an operational in-memory historian for mid-term storage, supporting fast analytics on the most relevant subset of near-term historical data, as well as interactive visualizations and dynamic report generation. Once the data has aged such that it is no longer required for near real-time analysis, the data can then be transitioned into a long-term, disk-based storage platform such as Enterprise Historian. A single interface layer should be implemented such that these three components (CEP-like engine, operational historian, disk-based store) all work in tandem to solve complex challenges and are viewed and interacted with as a single system. End users should not need to know where data resides when inserting data, running a query, or invoking an analytic, and thus would experience the infrastructure as one holistic hybrid high-velocity, high-volume Big Data platform.

11. CONCLUSIONS

GE Global Research, GE Intelligent Platforms, and GE Power & Water have designed and built Enterprise Historian to address the challenge of efficiently storing and analyzing Industrial Big Data with the goal of enabling GE Power & Water to perform deep historical analysis and data mining on tens to hundreds of terabytes of time series sensor data. Enterprise Historian is built on top of Apache Hadoop, an open-source framework for executing batch analytics on big data sets in a distributed computing environment. An instance of Enterprise Historian has been deployed on a 48-node Hadoop cluster at GE Global Research and with this instance, analytics that used to take days to weeks to execute can now be run in minutes to hours.

Enterprise Historian has been performance tested against a diverse suite of GE Power & Water gas and steam turbine RM&D analytics. These tests have demonstrated a 5x-180x speed-up for analytics that were possible to run in the existing Thermal RM&D environment. Perhaps even more importantly, analytics that were not feasible to run before (that would take weeks to months of largely manual effort), have been demonstrated to run in minutes to hours in the system.

Further testing was performed to evaluate the benefits of using the Proficy Historian archive file format vs. simply storing the time series data in plaintext files in Hadoop. These tests demonstrated comparable performance in query execution times, with the archive file outperforming plaintext when the queries required small to medium amounts of data, and the plaintext format outperforming the archive file when most of the data is being requested. While the query performance is comparable between the two storage models, the data footprint is substantially different—the archive file gives an 8x reduction in the size of the data over plaintext, significantly reducing storage costs.

Overall, placing tens to hundreds of terabytes of Thermal RM&D's Industrial Big Data in Enterprise Historian for massively parallel analysis and data mining operations is expected to lead to significant amounts of productivity for GE Power & Water and significant value to GE's customers through better management and operation of their equipment.

Disclaimer

12. REFERENCES

[1] Apache. Poweredby - hadoop wiki. http://wiki.apache.org/hadoop/PoweredBy, Nov. 2013.

[2] A. Ashutosh. Best practices for managing big data. http://www.forbes.com/sites/ciocentral/2012/07/05/best-practices-for-managing-big-data/, July 2012.

[3] A. Cronk, M. Yagley, and J. DeLay. Tempodb - time series database. https://tempo-db.com/, Aug. 2013.

[4] P. M. Davis. The history of hadoop: From small starts to big data. http://blog.gopivotal.com/news-2/the-history-of-hadoop-from-small-starts-to-big-data, Mar. 2013.

[5] J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, Jan. 2008.

[6] GE Intelligent Platforms. Proficy historian. http://www.ge-ip.com/products/proficy-historian/p2420, Aug. 2013.

[7] GE Intelligent Platforms. Proficy historian hd. http://www.ge-ip.com/products/proficy-historian-hd/p3714, June 2013.

[8] S. Ghemawat, H. Gobioff, and S.-T. Leung. The google file system. In *Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles*, SOSP '03, pages 29–43, New York, NY, USA, 2003. ACM.

[9] D. Harris. 5 reasons why the future of hadoop is real-time (relatively speaking). http://gigaom.com/2013/03/07/5-reasons-why-the-future-of-hadoop-is-real-time-relatively-speaking/, Mar. 2013.

[10] B. Hawkins and J. Sabin. Kairosdb - fast scalable time series database. https://code.google.com/p/kairosdb/, Aug. 2013.

[11] J. Kelly. Big data: Hadoop, business analytics and beyond. http://wikibon.org/wiki/v/Big_Data:_Hadoop,_Business_Analytics_and_Beyond, Sept. 2013.

[12] J. Kelly. The industrial internet and big data analytics: Opportunities and challenges. http://wikibon.org/wiki/v/The_Industrial_Internet_and_Big_Data_Analytics%3A_Opportunities_and_Challenges, Sept. 2013.

[13] M. Kornacker and J. Erickson. Cloudera impala: Real-time queries in apache hadoop, for real. http://blog.cloudera.com/blog/2012/10/cloudera-impala-real-time-queries-in-apache-hadoop-for-real/, Oct. 2012.

[14] D. C. Luckham. *Event Processing for Business: Organizing the Real-Time Enterprise*. Wiley, Dec. 2011.

[15] McKinsey. Big data: The next frontier for innovation, competition, and productivity. http://www.mckinsey.com/insights/business_technology/big_data_the_next_frontier_for_innovation, May 2011.

[16] C. Metz. How yahoo spawned hadoop, the future of

big data. http://www.wired.com/wiredenterprise/2011/10/how-yahoo-spawned-hadoop/, Oct. 2011.

[17] G. Pardo-Castellote. Omg data-distribution service: Architectural overview. In *Proceedings of the 23rd International Conference on Distributed Computing Systems*, ICDCSW '03, pages 200–, Washington, DC, USA, 2003. IEEE Computer Society.

[18] Pivotal. Pivotal hd: Hawq, a true sql engine for hadoop. http://www.gopivotal.com/sites/default/files/Hawq_WP_042313_FINAL.pdf, Sept. 2013.

[19] I. Shafer, R. R. Sambasivan, A. Rowe, and G. R. Ganger. Specialized storage for big numeric time series. In *Proceedings of the 5th USENIX Workshop on Hot Topics in Storage Systems*, 2013.

[20] K. Shvachko, H. Kuang, S. Radia, and R. Chansler. The hadoop distributed file system. In *Proceedings of the 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, MSST '10, pages 1–10, Washington, DC, USA, 2010. IEEE Computer Society.

[21] B. Sigoure. Opentsdb - a distributed, scalable monitoring system. http://opentsdb.net/, Aug. 2013.

[22] S. Sliwa, J. Peterson, and M. Derbecker. Seeq: Transforming industrial process data into actionable intelligence. https://www.seeq.com/, Nov. 2013.

[23] Twitter. Storm, distributed and fault-tolerant realtime computation. http://storm-project.net/, Sept. 2013.

[24] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica. Spark: Cluster computing with working sets. In *Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing*, HotCloud'10, pages 10–10, Berkeley, CA, USA, 2010. USENIX Association.