Big Data Engine

A Solution for Big Data Storage, Mining, and Analytics

Stephen V Moccio, Philip A Grim II

Abstract

In today's public and private industries, the volume of data being collected increases twenty five percent per year (Anderson, 2012). Many organizations are facing challenges to keep up with the volume and velocity of data collections because current storage capabilities are costly. In addition to the data being collected, even more data is created due to support advanced analytics as the richness of the data is extracted. This white paper will provide an approach that leverages widely available open source solutions in creating a data fusion and analytics platform for the storage and dissemination of the data while achieving the lowest total cost of ownership (TCO).

The Big Data Engine Platform (BDE) provides an out of the box, turnkey solution that allows for data fusion of structured, unstructured and semi-structured data sets regardless of modality. By leveraging the Hadoop NoSQL framework integrated with open source capabilities, BDE extends beyond a simple Hadoop distribution by including elastic ingest, data fusion, search and discovery, system monitoring, data management, data privacy, geospatial integration, security framework, data visualization, standard analytics, and more. This solution provides a new concept of storing, managing, and protecting content that scales with data and provides a platform for advanced analytics.

L3 Data Tactics' direction in creating BDE was to provide a quick solution to our customers so that they can start using their data on day one as opposed to creating a new platform each time. This platform reduces the twelve to eighteen months of engineering in order to stand up a solution. Our approach provides the platform to jumpstart your Big Data Strategy to start leveraging your data now. Our solution is designed to demonstrate that big data integration is accessible due to advances in innovative technology i.e., Hadoop.

Keywords

Big Data, Analytics, ETL, Storage, Visualization, Hadoop, Accumulo, MapReduce, Solr

About the Authors

Stephen V. Moccio is the L3 Data Tactics Technology Division Manager.

Philip A Grim II is the L3 Data Tactics Technology Division Chief Engineer. He has been involved in cloud computing efforts since the dawn of Hadoop, and participated in the architecture, design, development, and implementation of the first U.S. Department of Defense Data Fusion Cloud. His work has included ETL, Natural Language Processing, semantic data representation, analytics and visualization. He holds a BS in Computer Information Systems from St. Leo University, and an MS in Analytics from Harrisburg University of Science and Technology.

Contents

1. Introduction

Data growth is forcing the public and private industries to take a step back and identify strategies in handling the Velocity and Volume of data growth; The ability to store large amounts of data; And providing capabilities and advanced analytics in support of the business or mission critical objectives that will be supported. In addition to Velocity and Volume of data, large scale platforms also address the Variety and Complexity of Data.

Variety is the next challenges facing Big Data Solutions. Variety of data is defined by the type of data that is available in your date silos today. These data sets are classified under three modality types namely structured, semi-structured and unstructured. Structured data is the easiest to handle but in current solutions you have to provide a data model in which to load your data into traditional Relational Database Management Systems (DBMS) and defining models around unstructured and semi-structured is complex and a time consuming process.

Today, we define several key domains of Big Data Solutions. First Data Fusion of transactional and historical data sets which include social media, population make up, etc. Second, a robust search framework that leads to activity based intelligence (ABI) that is used to discover entity and objects on data sets. Third, Analytic Domain to describe both real time, analytics that are displayed during real time data capture to provide trends and analysis and batch analytics that are able to extract information across all of the data sets to provide link and relational analysis.

Apache Hadoop was created in 2005 to provide a way to handle large data volumes by providing a distributed storage framework for the distribution of data in a redundant and economical way by leveraging common off the shelf hardware components eliminating the need for customized hardware and software high cost solutions.

In addition to the distributed file system, Apache also released MapReduce, a distributed data processing framework that allows for distributed batch process analytics across the entirety of the data sets leveraged in Hadoop.

2. Big Data Engine

Our solution is based on the ability to provide a full-featured Hadoop-based analytics platform that will allow for a fast implementation process giving the customer a very short time-to-value approach to a Big Data solution. This base architecture allows the organization to continually add additional Analytics and Visualization features. We provide an Activity-Based Intelligence (ABI) User Experience focused on creating an environment that assists users in making analytic connections that are non-obvious outside of Big Data architecture. This allows use case discovery and data source identification with Data Source Investigation. With this data fusion theme the end user can conduct discovery by investigating structured and unstructured data in combination with the ability to fuse the data using geospatial, temporal and entity based controls. This is a complete end-to-end solution that can be rapidly deployed and provides the framework and tools necessary to provide real-time analysis in support of business-critical objectives.

Applying analysis to data is the first step in extracting value. Big Data has taken the market by storm yet is clearly still on the on the rise with additional analytics and engineering tools. In addition, the engineering expertise with open source Big Data architecture is becoming mainstream. Enterprises

are realizing the need to create order within their data and utilize analytics to answer specific questions, address specific issues, or identify trends that might affect their business going forward. BDE acknowledges these parameters and provides an offering that allows different analytic packages to be added based on customer needs.

2.1. Software Architecture

The success of Open Source Software (OSS) has been remarkable, forcing even the largest commercial software vendors to acknowledge its influence and, in some cases, adopt its methods. It seems likely that most organizations are familiar with, if not actively using, open source products on a daily basis. After about a decade of proving itself, OSS stacks have been moving from edge servers on the internet and department servers for branch offices to core business applications.

With OSS, we get freedom and flexibility. OSS allows you the ability to edit the underlying code, providing the flexibility to fit your needs. This flexibility also makes OSS more versatile in its interoperability with other software.

OSS is often the product of many people providing coding, ideas, feedback, and suggestions. More people review and test the software for the functionality that most concerns them. The net result is potentially wider-ranging Quality Assurance (QA) than any single entity might be able to support. A common reason enterprises choose OSS is because of its reliable quality. Since more people are involved in the development of OSS, more people find bugs. Exploits in OSS are noticed and fixed quicker. This transparency provides OSS a higher security level and more rapid problem resolution.

Public access to the OSS infrastructure allows more developers to test it. Additionally, OSS is usually delivered when the developers are satisfied that it is ready. It is typically built without the pressure of completion deadlines and the no-rush advantage allows developers to take adequate time perfecting it. Anything built slowly but surely is often more stable than something built in a rush.

More and more companies see the advantages of OSS compared to commercial and proprietary software. Seeing and understanding the philosophy behind and the actual development process of OSS, it is the apparent better choice as more companies recognize its merits.

We understand the benefits of OSS and the community that supports its innovation. Our solution uses OSS as the core of the pieces that, when combined, make the most innovative, agile, flexible, and performing system available. With the open source community continuing to improve on these technologies and creating new capabilities, we believe our solution also has the most potential for future value.

1) Operating Systems
    a) The Community Enterprise Operating System (CentOS) - A Linux distribution that provides a free Enterprise class computing platform which has 100 percent binary compatibility with its upstream source, Red Hat Enterprise Linux (RHEL).
    b) Red Hat Enterprise Linux (RHEL) - The preferred Linux distribution used by industry

and the US Government. RHEL is an open-source Enterprise operating system with professional-grade support.

2) Infrastructure Components

   a) DNS - The Domain Name System (DNS) is a hierarchical distributed naming system for computers, services, or any resource connected to the internet or private network.

   b) DHCP - The Dynamic Host Configuration Protocol (DHCP) is a standardized networking protocol used on Internet Protocol (IP) networks for dynamically distributing network configuration parameters, such as IP addresses for interfaces and services. With DHCP, computers request IP addresses and networking parameters automatically from a DHCP server, reducing the need for a network administrator or a user from having to configure these settings manually.

   c) LDAP - The Lightweight Directory Access Protocol (LDAP) is an application protocol for accessing and maintaining distributed directory information services over an IP network.

   d) NFS - Network File System (NFS) is a distributed file system protocol originally developed by Sun Microsystems in 1984, allowing a user on a client computer to access files over a network, much like local storage is accessed. NFS, like many other protocols, builds on the Open Network Computing Remote Procedure Call (ONC RPC) system. The NFS is an open standard defined in RFCs, allowing anyone to implement the protocol.

   e) MySQL - MySQL is an open-source relational database management system (RDBMS).

   f) Puppet - Puppet is an open source configuration management tool from Puppet Labs. It is written in Ruby and released as free software under the GPL, until version 2.7.0 and the Apache 2.0 license after that.

3) Cloud Computing Environment Components

   a) Apache Hadoop - Hadoop is an OSS framework for storage and large-scale processing of data sets on clusters of commodity hardware. Hadoop is an Apache top-level project being built and used by a global community of contributors and users. The Hadoop ecosystem consists of a number of subprojects, including:

      i) HBase - HBase is an open source, non-relational, distributed database modeled after Google's BigTable and is written in Java. It is developed as part of Apache Software Foundation's Apache Hadoop project and runs on top of HDFS (Hadoop Distributed Filesystem), providing BigTable-like capabilities for Hadoop.

      ii) HDFS – HDFS is a distributed, scalable, and portable file-system written in Java for the Hadoop framework. Each node in a Hadoop instance typically has a single DataNode, a cluster of DataNodes form the HDFS cluster. The situation is typical because each node does not require a DataNode to be present. Each DataNode serves up blocks of data over the network using a block protocol specific to HDFS.

iii) Zookeeper - Apache ZooKeeper is a software project of the Apache Software Foundation, providing an open source distributed configuration service, synchronization service, and naming registry for large distributed systems. ZooKeeper's architecture supports high-availability through redundant services.

iv) Flume - Flume is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of log data. It has a simple and flexible architecture based on streaming data flows. It is robust and fault-tolerant with tunable reliability mechanisms and many failover and recovery mechanisms.

v) Oozie - Oozie is a workflow scheduler system to manage Hadoop jobs. It is a server-based Workflow Engine specialized in running workflow jobs with actions that run Hadoop MapReduce and Pig jobs.

vi) MapReduce - MapReduce is a programming model for processing large data sets with a parallel, distributed algorithm on a cluster.

vii) Apache Accumulo - Apache Accumulo is a computer software project that developed a sorted, distributed key/value store based on the BigTable technology from Google. It is a system built on top of Apache Hadoop, Apache ZooKeeper, and Apache Thrift. Written in Java, Accumulo has cell-level access labels and server-side programming mechanisms.

viii) Apache Solr - Solr is an open source Enterprise search platform from the Apache Lucene project. Its major features include full-text search, hit highlighting, faceted search, dynamic clustering, database integration, and rich document (e.g., Word, PDF) handling. Providing distributed search and index replication, Solr is highly-scalable. Solr is the most popular Enterprise search engine.

ix) Apache ActiveMQ - ActiveMQ is an open source message broker written in Java together with a full Java Message Service (JMS) client. It provides "Enterprise features" which in this case means fostering the communication from more than one client or server.

4) Web Service Components

a) CAS is an authentication system originally created by Yale University to provide a trusted way for an application to authenticate a user. It integrates with ActiveDirectory or LDAP to provide unified login to BDE services.

b) Apache Tomcat - Apache Tomcat is an open source web server and servlet container developed by the Apache Software Foundation (ASF). Tomcat implements the Java Servlet and the JavaServer Pages (JSP) specifications from Sun Microsystems, and provides a "pure Java" HTTP web server environment for Java code to run in.

c) Apache HTTPD - The Apache HTTP Server, commonly referred to as Apache is a web server application notable for playing a key role in the initial growth of the World Wide Web.

d) RStudio Server - RStudio is a free and open source integrated development environment (IDE) for R, a programming language for statistical computing and graphics.

e) Shiny Server - Shiny is an open-source web application plugin for the R programming language. It provides easy-to-deploy analytics visualization capability with user-configurable controls.

f) OpenGeo - OpenGeo is the geospatial division of OpenPlans which supports the development of a number of OSS packages for the Geospatial analysis, management and publication of geo-spatial information; these include PostGIS, GeoServer, GeoWebCache, GeoExt, and OpenLayers.

g) Ganglia - Ganglia is a scalable distributed system monitor tool for high-performance computing systems such as clusters and grids. It allows the user to remotely view live or historical statistics

5) Libraries and Programming Interfaces

a) Oracle Java

b) Apache Commons - The Apache Commons libraries provide commonly-used application programming interfaces for such functions as logging, data manipulation, encoding, parsing, and storage.

c) Apache Lucene - Apache Lucene is the indexing engine behind Solr, providing full text, temporal, and geospatial indexing for all data in the Unified Dataspace.

d) HTML5 - HTML5 is a markup language used for structuring and presenting content for the World Wide Web and a core technology of the Internet. Its core aims have been to improve the language with support for the latest multimedia while keeping it easily readable by humans and consistently understood by computers and devices (web browsers, parsers, etc.). It also defines a single markup language that can be written in either HTML or XHTML syntax. It includes detailed processing models to encourage more interoperable implementations; it extends, improves and rationalizes the markup available for documents, and introduces markup and application programming interfaces (APIs) for complex web applications. For the same reasons, HTML5 is also a potential candidate for cross-platform mobile applications. Many features of HTML5 have been built with the consideration of being able to run on low-powered devices such as smartphones and tablets.

e) jQuery - jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers.

f) OpenMap - BBN Technologies' OpenMap package is a JavaBeans based programmer's toolkit. OpenMap enables quick building of applications and applets that access data from legacy databases and applications, and provides the means to allow users to see and manipulate geospatial information.

g) R Statistical Programming Language - R is a free software programming language and software environment for statistical computing and graphics. The R language is widely used among statisticians and data miners for developing statistical software and data analysis.

h) General Architecture for Text Engineering (GATE) - GATE is an open source framework for the development of text analytics. It provides services for natural

language processing, entity extraction, semantic analysis, sentiment analysis, and many other text analytic functions.

6) ISO Standards
    a) ISO/IEC 23360 Linux Standard Base - Both the Community Enterprise Operating System (CentOS) and Red Hat Enterprise Linux (RHEL) are implementations of the Linux Standard Base.
7) IEEE Standards
    a) IEEE Standard 1003 POSIX Compliance - The Community Enterprise Operating System (CentOS) is classified as "mostly POSIX compliant. The Red Hat Enterprise Linux (RHEL) operating system is classified as "mostly POSIX compliant."
8) Data/Metadata Standards - The BDE does not natively implement any data or metadata standards. Instead, the unified dataspace provides the tools to implement any standards that the customer needs, including security and privacy standards such as Controlled Access Program Coordination Office (CAPCO) security marking, Privacy Act and other personally identifiable information standards, ACP128 military messaging standards, or any other standard that is required.

Our Development Team is committed on operating on or near the cutting edge of technology and staying there as technology changes.

2.2. System Infrastructure

The BDE platform provides elastic ingest capabilities to process structured, unstructured, and semi-structured data sources without the need for pre-defined data models that facilitate high-level quantitative analytics and searches. This solution provides thin client queries, search for resolved entities, and secures infrastructure with role and user-based access control.

This Big Data solution, built on the NoSQL Hadoop framework, supports petabyte-scale storage and processing capabilities, as well as the operation of customer-defined systems and service. It delivers high-speed data access to large data volumes, while providing the customer with the immediate insight they need.

2.2.1. Storage

Hadoop/HDFS - The core of the BDE storage solution is the Hadoop Distributed File System (HDFS). An HDFS cluster (Figure 2) primarily consists of a NameNode that manages the file system metadata and DataNodes that store the actual data. The secondary NameNode provides fault tolerance and backup for the primary. The BDE is capable of integrating with several different Hadoop implementations, including the Apache software foundation's reference implementation, Cloudera's Distribution of Hadoop (CDH), and the MapR file system (MapRFS), among others.

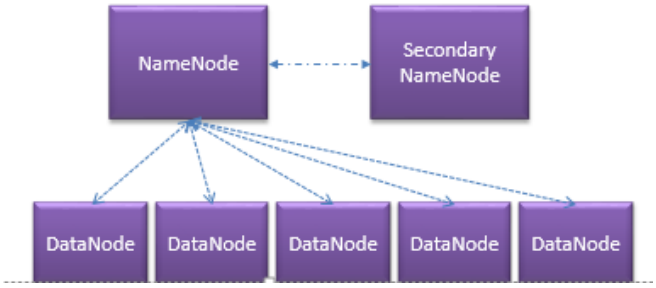


Figure 1

MapReduce is the distributed programming framework pioneered by Google that forms the basis for the BDEs analytic capabilities. MapReduce (Figure 3) works in concert with HDFS to perform highly parallel computation on the cluster nodes where the data is stored.

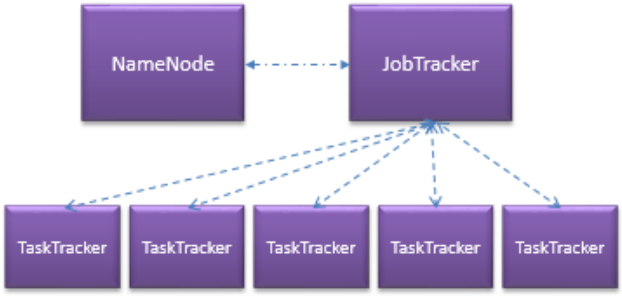


Figure 2

Accumulo - Apache Accumulo is a NoSQL columnar store (Figure 4). Accumulo was developed by the National Security Agency as a way to add security to a columnar store such as HBase, and was subsequently released as open source. The BDE uses Accumulo as the basis for its unified dataspace. It provides fine-grained access control to data based on users and roles.

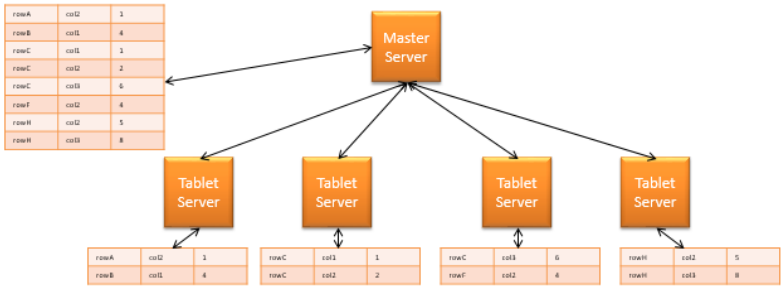


Figure 3

Solr - Apache Solr provides a distributed indexing capability that BDE leverages for full text, temporal, and geospatial querying, as well as faceted and model-based queries. Multiple Solr servers (Figure 5) provide replication and load balancing for fault tolerance and performance.

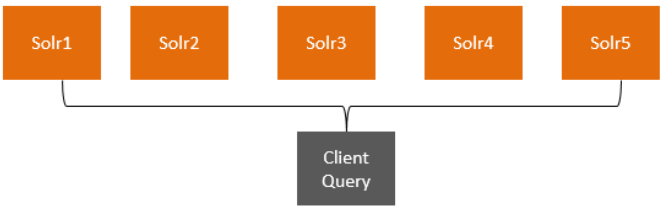


Figure 4

2.2.2. Elastic Ingest

Our elastic ingest capabilities (Figure 6) to process structured, unstructured, and semi-structured data sources transform customer-defined data models that facilitate high-level quantitative analytics and searches. This solution provides thin client queries, searches for resolved entities and secures infrastructure with role and user-based access control.

Figure 5

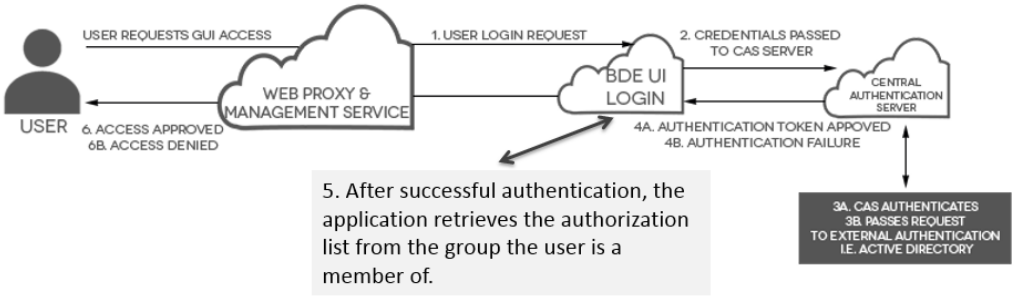


Figure 6

Natural Language Processing / Entity Extraction - Entity extraction is performed in various ways using a number of related technologies. Some types of named entities, such as telephone numbers and IP addresses are easily recognized using regular expression matching. Other types of named entities, such as proper names of people, places, organizations, and facilities require more advanced techniques such as gazetteer lookup, ontology, and grammar-based transduction. The combination of these technologies is generally referred to as Named Entity Recognition (NER), which is a subset of the field known as Natural Language Processing (NLP).

Our team has developed a set of technologies to perform some portions of the NER process using the General Architecture for Text Engineering (GATE), an open source alternative to commercial NLP applications.

GATE, developed by the University of Sheffield, is an infrastructure for developing and deploying software components that process human language. GATE helps scientists and developers in three (3) ways:

- By specifying an architecture or organizational structure for language processing software
- By providing a framework or class library that implements the architecture that can be used to embed language processing capabilities in diverse applications
- By providing a development environment built on top of the framework made up of convenient graphical tools for developing components.

The architecture exploits component-based software development, object orientation, and mobile code. The framework and development environment are written in Java and are available as open source free software under the GNU library license.

The ANNIE (A Nearly-New Information Extraction) system is a plugin distributed with GATE that performs the various functions of NER. ANNIE provides a configurable pipeline of functions that annotate a given corpus of text with the various types of tokens and entities found in each step of processing. ANNIE is highly-configurable to adapt to various domains of knowledge. Figure 8 illustrates the components of the ANNIE system.

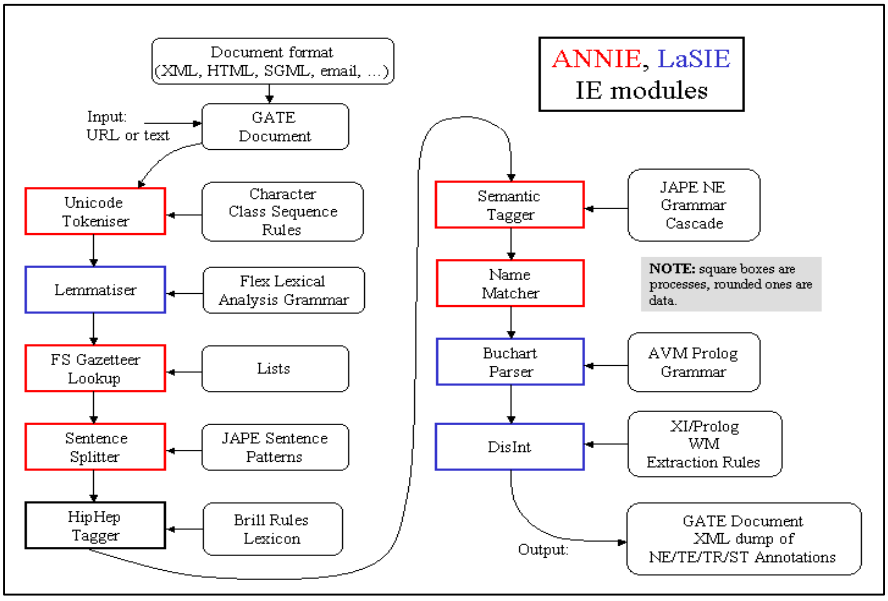


Figure 7

The focus of the BDE program's entity extraction development is on the gazetteer and semantic tagger portions of the ANNIE system. These components are easily configured and modified to adapt to the types of entities, which are useful to the analyst.

The gazetteer is based on dictionary lists, and uses a finite-state machine to match the tokens in the lists to words in the text being searched. These lists contain words in specific categories, as well as arbitrary feature data that can be assigned to the words.

The semantic tagger uses a rule-based system to analyze the grammar of the document. Input can be taken from any of the processes that occurred earlier in the pipeline, so annotations from the lemmatiser, sentence splitter, and gazetteer can be combined in rules to recognize specific entities. The left-hand side of the rules uses a grammar called the Java Annotations Pattern Engine (JAPE) to essentially allow regular expressions to be run over annotations. The right-hand side of the rules is Java, which is compiled at runtime and can modify annotations or create new ones as needed. JAPE rules compile into finite state machines.

ANNIE's JAPE rules are also very configurable, and when combined with the custom dictionaries in the gazetteer, they provide a powerful system to tune the entity extraction process to meet the needs of the customer.

The elastic ingest system included in the BDE can receive data directly from accessible data sources. This could mean, for example, that a site that collects data from sensors could allow BDE ingest to receive that data directly in concert with the local system, via a web service, RSS feed, socket connection, or similar methodology.

Trusted file transfer agents can be used to move larger blocks of data between sites. SafeMove is a government off the Shelf (GOTS) application used by the Defense Common Ground Station – Army (DCGS-A) program and several others to move data between sites reliably. Commercial options such as Aspera and Digital Fountain also perform this function.

The Big Data Engine (BDE) Platform provides a data ingest service that builds on Hadoop, Accumulo, and Solr as represented in Figure 9.

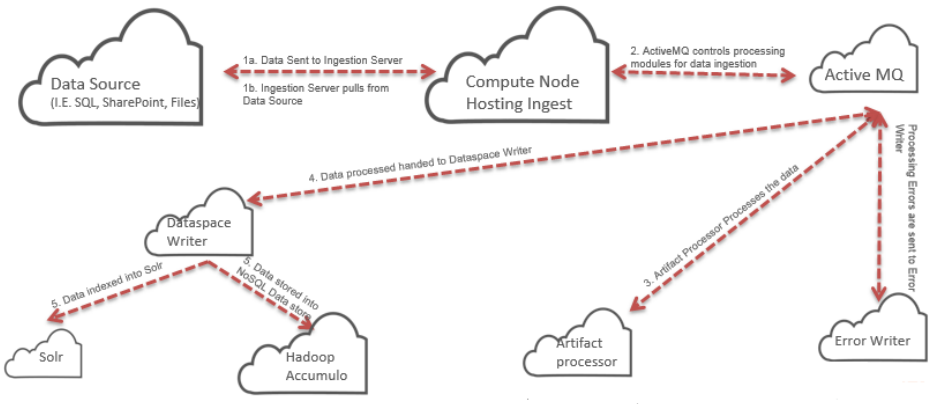


Figure 8

- Distributed - BDE ingest components are hosted on the same Compute Nodes as the Hadoop and Accumulo cloud processes. Apache ActiveMQ provides a

distributed work queue mechanism that allows the ingest processes to communicate.

- Scalable - The ingest system will automatically initiate more processes as the workload grows, and will release resources when they are no longer required.
- Fault-tolerant - The ingest system monitors and automatically restarts any failed processes. ActiveMQ persists data that is in process, so if a node fails, another node can take over processing that data.
- Extensible - The ingest systems is modular and presents an application programming interface to allow custom parsers for new data types, and custom processors to add functionality.
- Manageable – The system provides graphical tools for configuration, monitoring, and error handling. Easily add new data sources, configure the load balancing characteristics of the ingest system, view volume and performance statistics, and manage errors in data ingestion.

2.2.3. Security

The security architecture (Figure 7) used in the BDE platform was developed around two (2) security components: authentication and authorization. Authentication is the process by which the system validates the identity of users. BDE supports many different single sign-on authentication systems such as Active Directory and Lightweight Directory Access Protocol (LDAP). BDE also makes available its own single sign-on capability for those environments that either do not provide their own, or wish to segregate BDE authentication from the overall corporate architecture.

Authorization is the mechanism through which the system determines what services, applications, and data that each user may access. Authorization may be specified through groups in LDAP or Active Directory, directly within the BDE system or through other means, as determined by the needs of the customer.

The BDE platform provides operating system level security through Information Technology Infrastructure Library (ITIL) best practices compliance, Security Technical Implementation Guides (STIGs) as directed by DISA Field Security Operations (FSO), data space wide security, data level row and cell level security for classified information storage compliance, and Central Authentication Service (CAS) user role/user ID authentication based security through Lightweight Directory Access Protocol (LDAP) and Active Directory (AD).

The security architecture used in the BDE platform was developed around two (2) security components: User authentication and Data authorization. User Authentication is the process by which the system validates the identity of users. BDE supports many different single sign-on authentication systems such as Active Directory and Lightweight Directory Access Protocol (LDAP). BDE also makes available its own single sign-on capability for those environments that

either do not provide their own, or wish to segregate BDE authentication from the overall corporate architecture.

Data Authorization is the mechanism through which the system determines what services, applications, and data each user may access. Authorization may be specified through groups in LDAP or Active Directory, directly within the BDE system, or through other means as determined by the needs of the customer.

At the heart of any secure system is a secure data store. BDE uses cell-level security on every item stored within the system. Each atomic data value (text string, dollar amount, customer ID, etc.) is marked with a label expression describing the rules by which that data item may be accessed. Not only can each row of data in the system be protected uniquely, but each individual column within each row may have its own access control specification. This gives the customer the finest level of access control possible and allows all data, regardless of security requirements, to be stored in a single, unified data space.

Our team will works closely with customers to fully understand, document, and implement a security-marking scheme that satisfies all data protection requirements.

2.2.4. Visualization

Visualization is a critical accelerator for data exploration. Our system allows an analyst to see the data from a single dashboard allowing data discovery, proceeding on a journey guided by curiosity. The best Big Data integration technology allows visual exploration of data independent of the type of data or the source from which it came.

The more data sources you have, and the bigger those data sources are, the more you need effective visualization to help you with understanding exactly what data you have and how it can solve the business problem.

Blending data sources is key part of the analytics process. By visualizing this process in a tool that allows you to verify that the way you're blending data accurately reflects the meaning of the data sources. Enriching traditional data sources with relevant big data sources becomes an exciting visual experience rather than a tedious task.

Visualization leverages the incredible capabilities and bandwidth of the visual system to move a huge amount of information into a digestible format. These visualization tools take advantage of the human brain to identify patterns and communicate relationships and meaning. This ability to "see" the data can inspire new questions and further exploration.

Our use of intuitive visualization tools (Figure 10) aids the identification of trends and outliers, discovering interesting or specific data points with the use of textual, temporal, and spatial analytics and visualization.

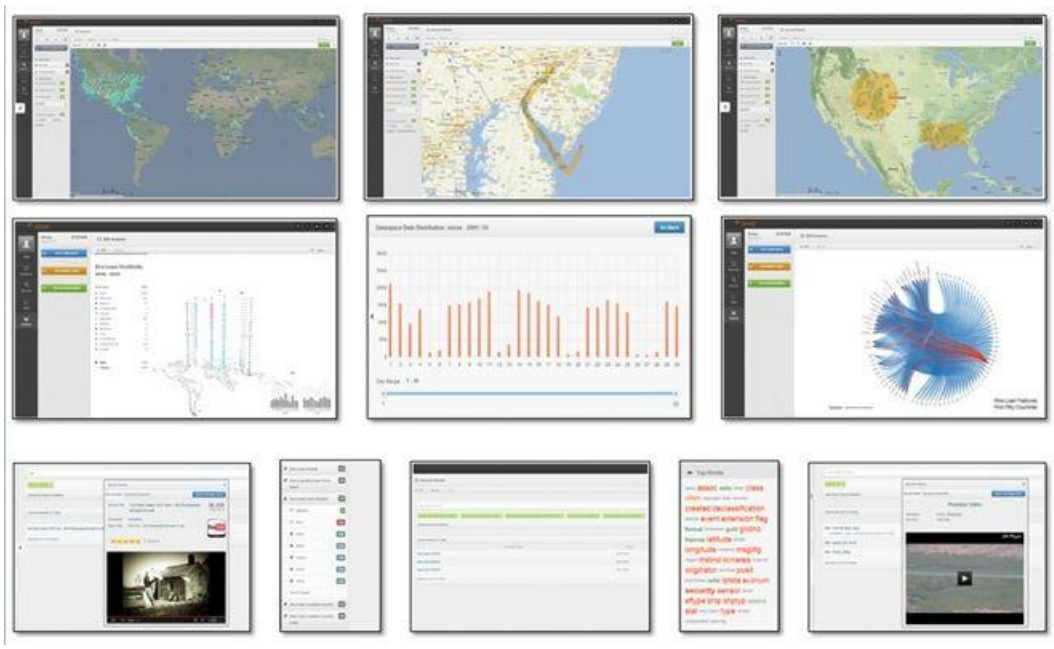


Figure 9

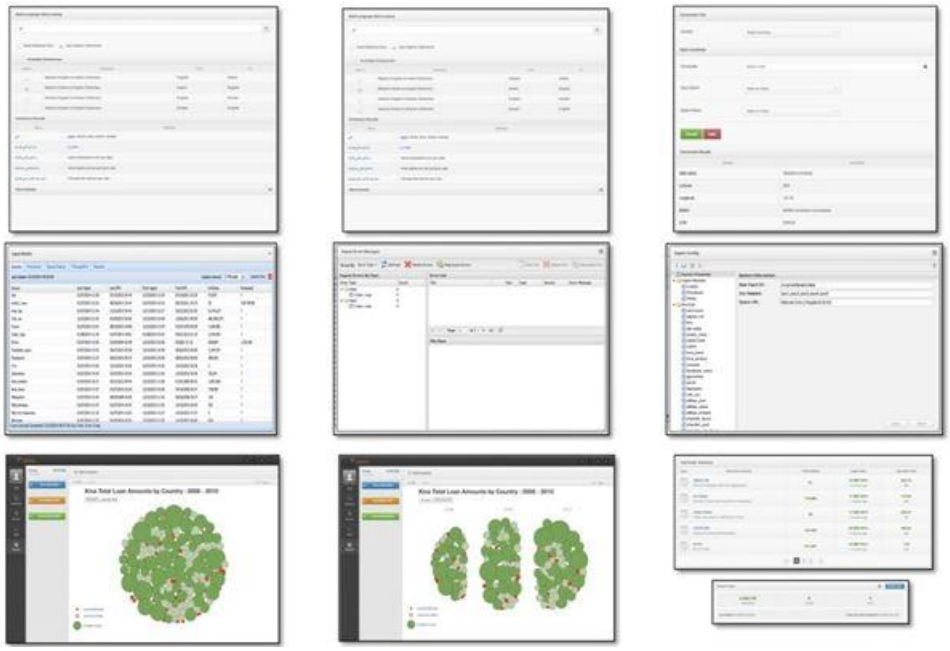
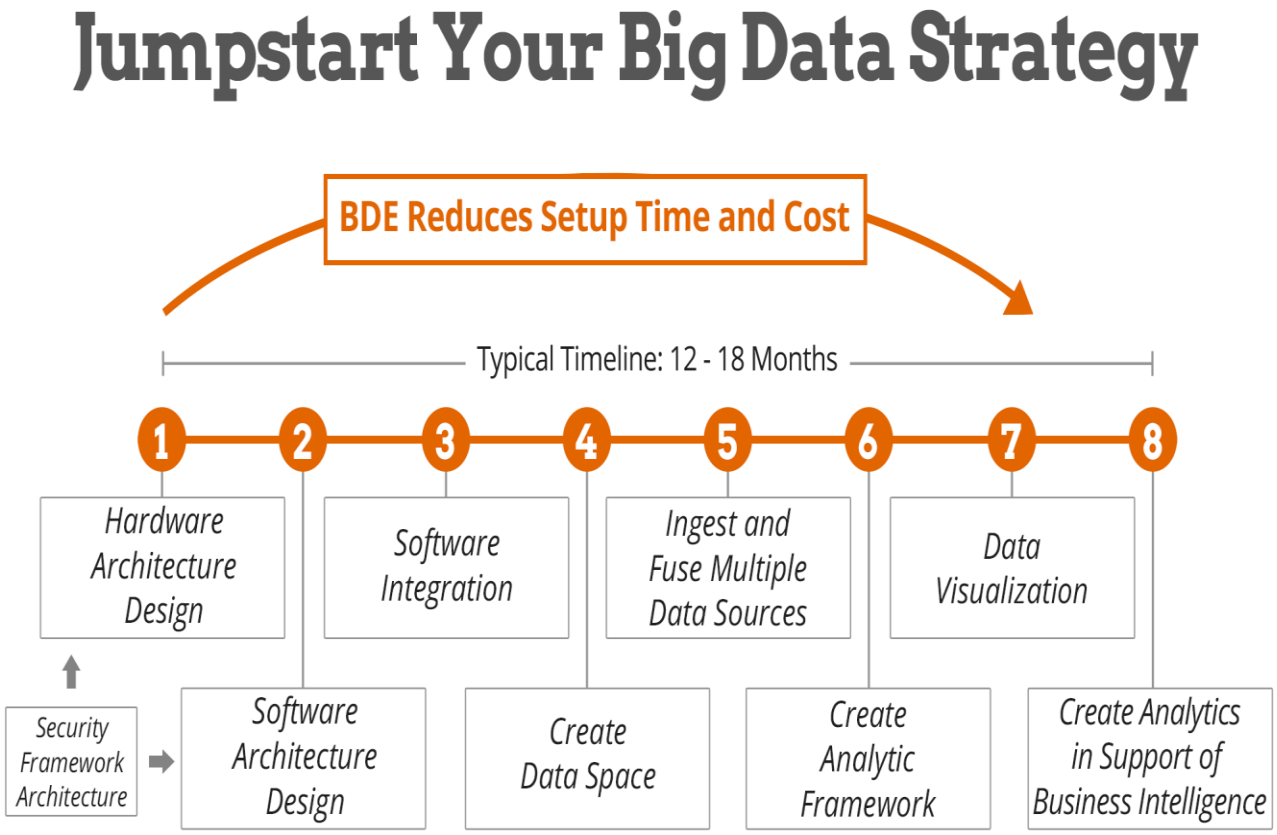


Figure 10

3. Conclusion

Many organizations today are starting big data projects that addresses the requirements of their business and mission objectives. The increase drive for scalable and flexible solutions are largely due to several factors. Current solutions are facing scaling challenges and those that can scale are doing so at a high cost and companies are trying to leverage Big Data at an affordable price.

Organizations trying to leverage open source solutions to manage data growth are facing engineering challenges in defining the hardware and software architectures needed to manage data and provide an analytic framework. As seen in Figure 1, there are 8 standard steps in building your NoSQL architecture and capabilities.



Steps 1 and 2 are focused on creating the hardware and software architectures required to support business and mission critical objectives in the public and private industries. The adoptive architecture needs to be robust enough to support growth and capabilities as intelligence needs evolve over time. There are several factors when considering a hardware architecture and they are mostly centered on the data and analytic support that is needed. For example, if the goal is to provide a distributed file store than the architecture chosen needs to support as many disk spindles as possible for fast I/O within the Hadoop cluster. If the goal is to provide analytic enrichment of static or small data sets then an optimized cluster would be one that has many CPUs available to process the data. There is also a hybrid

approach that marries a distributed file store and large scale computing that is a balanced approach for optimizing the hardware for both the number of disk spindles and CPUs.

The most overlooked process when creating your framework is defining the security framework that will enable data privacy, data authentication and data governance. This security framework applies to the hardware, software and extends to the applications and capabilities throughout the entirety of the platform. If over looked at the beginning, the defined architecture falls apart as you are unable to add security to the entire stack after the platform is created and most architects start over.

Also seen in Figure 1 are the additional steps in building an architecture to support business and mission critical objectives. Software integration; data space creation; ability to ingest and fuse data; analytic framework and data visualizations are necessary. These projects can take 12-18 months prior to having the ability to actually perform intelligence with your data. This requires subject matter experts in the field of big data to be able to build these platforms in house at a high cost based on the number of dedicated resources needed.