

Big Data Architecture

Bhashyam Ramesh

Abstract Big data is a broad descriptive term for non-transactional data that are user generated and machine generated. Data generation evolved from transactional data to first interaction data and then sensor data. Web log was the first step in this evolution. These machines generated logs of internet activity caused the first growth of data. Social media pushed data production higher with human interactions. Automated observations and wearable technologies make the next phase of big data. Data volumes have been the primary focus of most big data discussions. Architecture for big data often focuses on storing large volumes of data. Dollars per TB (Terabyte) becomes the metric for architecture discussions. We argue this is not the right focus. Big data is about deriving value. Therefore, analytics should be the goal behind investments in storing large volumes of data. The metric should be dollars per analytic performed. There are three functional aspects to big data—data capture, data R&D, and data product. These three aspects must be placed in a framework for creating the data architecture. We discuss each of these aspects in depth in this chapter. The goal of big data is data-driven decision making. Decisions should not be made with data silos. When context is added to data items they become meaningful. When more contexts are added more insight is possible. Deriving insight from data is about reasoning with all data and not just big data. We show examples of this and argue big data architecture must provide mechanisms to reason with all data. Big data analytic requires all forms of different technologies including graph analytics, statistical analytics, path analysis, machine learning, neural networks, and statistical analysis be integrated in an integrated analytics environment. Big data architecture is an architecture that provides the framework for reasoning with all forms of data. We end this chapter with such architecture. This chapter makes three points as follows: (a) Big data analytics is analytics on all data and not just big data alone; (b) Data complexity, not volume, is the primary concern of big data analytics; (c) Measure of goodness of a big data analytic architecture is dollars per analytics and not dollars per TB.

B. Ramesh (✉)
Teradata Corporation, Dayton, USA
e-mail: vembakkambhashyam@gmail.com

Keywords Data lake • Telematics • Data volumes • Business intelligence • HDFS • Sessionizing

1 Introduction

This chapter gives the architecture for big data. The intent is to derive value from big data and enable data-driven decision making. The architecture specified here is for the data-centric portions. It does not cover mechanisms and tools required for accessing the data. The choice of such mechanisms is based at least in part on the interfaces necessary to support a wide variety of access tools and methods. They must support all forms of applications including business intelligence (BI), deep analytics, visualization, and Web access to name a few.

Big data is a broad term. Wikipedia defines big data as “an all-encompassing term for any collection of data sets so large and complex that it becomes difficult to process using traditional data processing applications.” We agree with this definition of big data.

Benefitting from big data, that is, deriving value from big data requires processing all data big and not so big. In other words, big data processing is not an isolated exercise. Just as data understanding increases with descriptive metadata, value of data in general and value of big data in particular increase with context under which the data is analyzed. This context resides in many places in the organizations’ data repository. These must be brought under an overarching architecture in order to fully derive value from big data.

The rest of this chapter is organized as follows:

We discuss data and its key characteristics as foundations for understanding the data architecture. Sections 2–4 cover these discussions. Section 2 defines big data. It describes the three different components of big data. Section 3 defines different characteristics of big data and their evolution. It shows that data growth has been in spurts and has coincided with major innovations in storage and analytics. Big data is the next stage in this data growth story. Section 4 specifies some value metric as a motivation for evaluating the architecture.

We then discuss mechanisms for deriving value from data. Sections 5–7 cover these discussions. Section 5 specifies three functional components of data-driven decision making. It explains the phases that data goes through in order to become a product. Section 6 lays the foundation for the assertion that deriving value from big data is about reasoning with all data and not just big data. Section 7 focuses on data analytics and the evolutions in that space. It argues for an integrated analytics framework.

Section 8 gives the overall architectural framework by building on these earlier sections.

Finally, Sect. 9 puts them all together with a use-case.

2 Defining Big Data

Gartner defines big data in terms of 3Vs—volume, variety, and velocity [1]. These three Vs are briefly explained in the following paragraphs. Some add a 4th V for veracity or data quality [2]. Data quality is a huge problem in practice. Quality of human-entered data is typically low and even automatically collected data from mobiles and sensors can be bad—sensors can go bad, sensors may need calibration, and data may be lost in transmission. In addition, data may be misinterpreted due to incorrect metadata about the sensor. It is important that data should be checked for correctness constantly. We do not address the fourth V in this chapter.

The 3Vs can be viewed as three different dimensions of data.

Volume, the most obvious of the 3V, refers to the amount of data. In the past, increase in data volumes was primarily due to increase in transaction volume and granularity of transaction details. These increases are small in comparison with big data volumes. Big data volume started growing with user interactions. Weblogs was the starting point of big data. Size and volume of Weblogs increased with internet adoption. These machine logs spawned a set of analytics for understanding user behavior. Weblogs seem small compared to social media. Facilities such as Facebook, Twitter, and others combined with the increase in capability and types of internet devices caused a big jump in data production. The constant human desire to connect and share caused an even bigger jump in data production. Social media allows users to express and share video, audio, text, and e-mail in volumes with a large social audience. The corresponding sophistication in mobile technology makes sharing easier and increases the volumes even further. This is the second wave of big data. The next big jump in volume will be from automated observations. All kinds of biometric sensors (heart, motion, temperature, pulse, etc.) and motion sensors (GPS, cellular, etc.) and the ability to move them through cellular, internet, Wi-fi, etc., is the next increase in big data volume. Wearable technologies and the internet of things (IOT) is the next big thing that is waiting to happen. These are referred as observation data. Sensor data and observations will dwarf the volume we have seen so far. The following Fig. 1 shows the relative increases in volume as we move from transactions to interactions to observations.

Data variety is as diverse as the data sources and their formats. Unlike transaction data which is highly structured in relational form, other data forms are either relatively weakly structured such as XML and JSON or differently structured such as audio, video, or without structure such as text, scanned documents. Different data sources such as Weblog, machine log, social media, tweets, e-mails, call center logs, and sensor data all produce data in different formats. These varieties of data make analysis more challenging. Combining variety with volume increases the challenges.

Velocity is the speed at which data flows into the system and therefore must be handled. Data comes from machines and humans. Velocity increases with the number of sources. Velocity increases with the speed at which data is produced such as with mobiles and sensors.

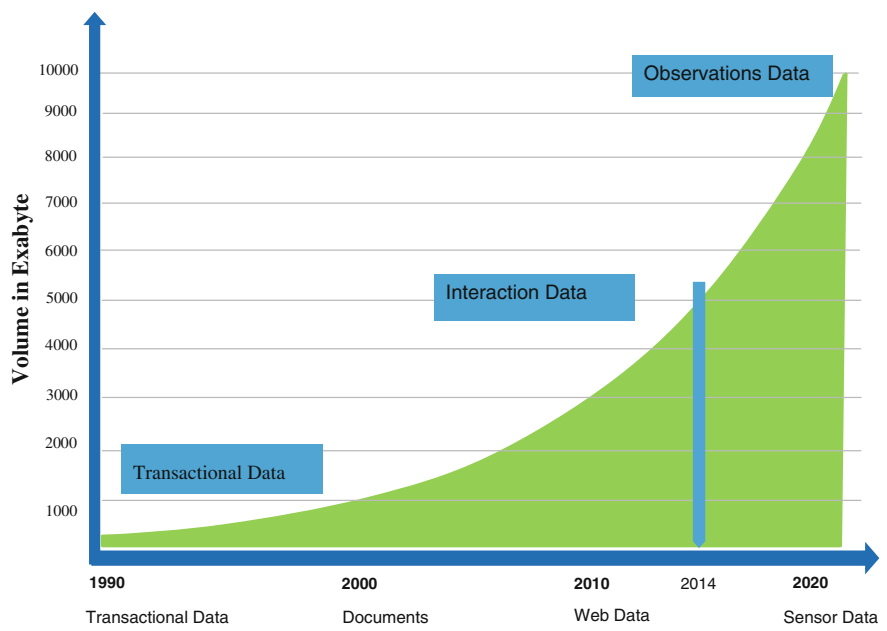


Fig. 1 Volume versus variety

We combine the 3Vs and call it the complexity of data. Complexity refers to the ability to analyze, derive insight, and value from big data. Deriving insight from big data is orders of magnitude more complex than deriving insight from transactional

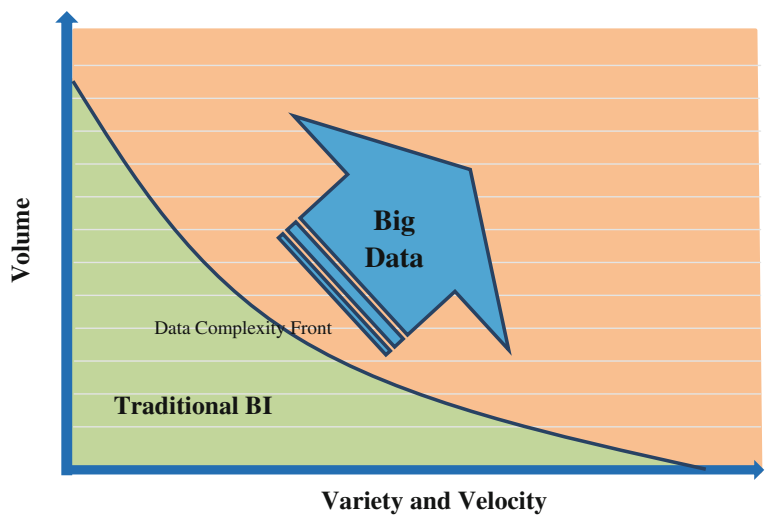


Fig. 2 Moving complexity front related to 3Vs

data. Deriving insight from sensor data is more complex than deriving insight from user-generated data.

Figure 2 shows variety and velocity in one axis versus volume in another axis. Complexity is the moving front. Complexity increases as the front moves to the right and top; complexity decreases as the front moves down and left.

One of the purposes of storing and analyzing big data is to understand and derive value; otherwise, there is no point in storing huge volumes of data. Complexity of gaining insight is directly related to the complexity of data. Analytics on complex data is much harder. It is complex in the type of analytic techniques that are needed and it is complex in the number of techniques that have to be combined for analysis. We cover these aspects later in this chapter. There is also a chapter dedicated to big data algorithms in this book.

3 Space of Big Data

Figure 3 shows the different types of data that make up the big data space. Each type of data exhibits different value and different return on investment (ROI).

Each shaded transition in this picture represents a major plateau in data growth. Each shaded region represents a quantum jump in analytic complexity, data capture, data storage, and management. The quantum jump is followed by a duration of incremental growth in such capabilities until the next quantum jump in such

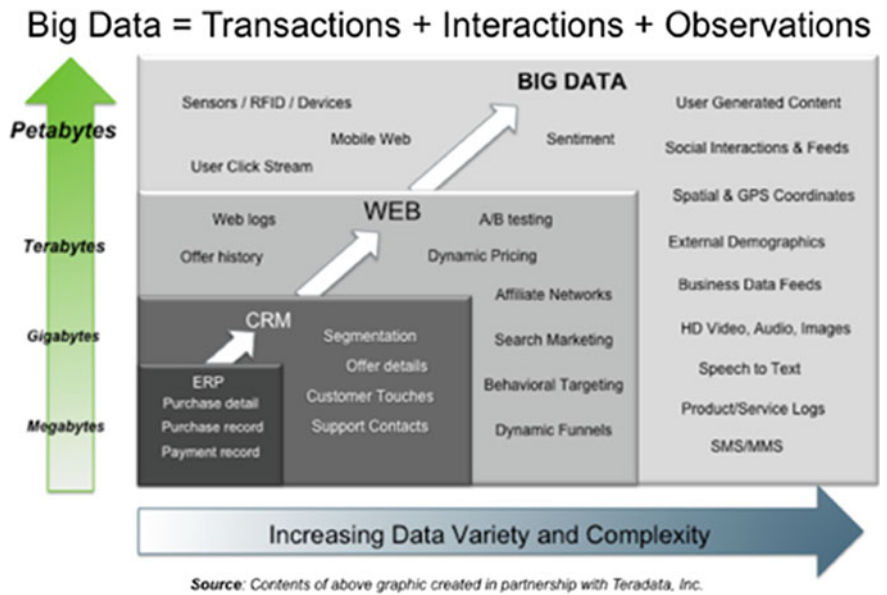


Fig. 3 Big data space

capabilities occur. There have been many plateaus over the years and many minor plateaus within each major plateau. Each has coincided with a jump in technology related to capture, transmission, management, and analytics. Each plateau has required some new forms of analytics. Each jump in technology has enabled a move to the next plateau. Each plateau has meant the ability to process much bigger volumes of data.

There is a ROI threshold for each plateau of data size. This means there is a balance between the cost associated with storing and managing the data and the value that can be derived from the data through application of analytics. Storing them is not useful when cost exceeds the value that can be derived. ROI determines whether the data is useful to be stored and managed. The ROI threshold is not reached until the technology and the cost of doing analysis is cheap enough to get a ROI from storing and analyzing the data. This notion is best described using transactional data as example. In the past retail applications used detail data at the level of store-item-week. Analysis used to be at this level of store-item-week granularity. Users were satisfied doing analysis at this summary level of detail when compared to what they were able to do before. However, users realized more detail than week-level summary will help, but the technology was unaffordable for them to store and manage at more detail levels. When technologies made it possible to go to store-item-day, then to market basket summaries, and then onto market basket detail, these new data plateaus became the new norm. At each point along the way users felt, they had reached the correct balance between cost and capability and any further detail, and data volumes were uneconomical since they did not see any value in them. However, leading-edge users made the conceptual leap to more detail first. They were visionaries in knowing how to get value from more detailed data. Now those plateaus have become common place; in fact, they are table stakes in any analytic environment. It is now relatively inexpensive to store at the smallest SKU level of detail, practical to do the analysis, and suitable to get value from such detailed data. There are such examples in every industry. In the communications industry, the plateau evolved from billing summary data which is few records per customer per month to storing and managing call detail records which are records of every call ever made! The new plateau now in the communications industry is network-level traffic data underlying each call. So there have been such plateaus before in all varieties of data.

Big data is just another plateau. Currently, we are seeing the leading edge of the big data plateau.

The capture and storage technologies appropriate for transactions is inappropriate for Weblogs, social media, and other big data forms. Technologies that can store large volumes of data at low cost are needed. Leading-edge users have found ways to derive value from storing interactions such as clickstreams and user navigation of Web sites. They determine the likes and dislikes of customers from these data and use them to propose new items for review. They analyze social graphs in order to differentiate between influencers and followers with a goal to tailor their marketing strategies. Such analyses are now moving from the leading edge to common place.

Currently, users attempt to reduce the volume of data in an effort to make it cost-effective to manage. They apply filters on acquired raw data in order to reduce its size for analysis. Technology will eventually become cheaper and bigger per unit of floor space and make it cost-effective to store, manage, and analyze all the detailed data instead of summaries of such filtered big data.

3.1 The Next Data Plateau

The next data plateau is on the horizon. The next jump will be from wearable technologies, sensors, and IOT. Many users wonder how to derive value, while leading-edge thinkers are finding innovative ways.

Social networks and the digital stream which are the current focus of big data are medium complexity in comparison with sensor data. Special scientific customers such as the supercollider already store sensor data. IOT will make such amounts common place. Mobile phones with biometric and other sensors collect large amounts of data and transmit them easily. These data are already far beyond any social networks. Wearable computing and IOT will increase this further. IOT is like an electronic coating on everyday things, a sensor embedded on everyday items that makes them seem sentient. IOTs can be linked together to form a network of sensors that can provide a bigger picture of the surroundings being monitored—a Borg-like entity without the malevolent intent.

Leading-edge companies and applications are starting to experiment with the beginning stages of this next plateau. Currently, sensor telemetry is an example of what they are doing. Insurance companies were considered innovators when they used credit rating scores with transaction data to determine premiums. Now they are innovating with big data. They create customized insurance policies based on individual driving behavior such as how far you drive, how well you drive, when you drive, what conditions under which you drive, and where you drive. They monitor every driver action to understand driving behavior. Clearly, a lot of sensors are involved in collecting and transmitting this data. Typically the driver has a small device in the car such as under the steering column. This device monitors the drivers' pattern of driving—speed, rash cornering, sudden stops, sudden accelerations, time and distance-driven, weather conditions under which the car is being driven such as rain and snow, light conditions such as day or night, areas where the car is being driven, the driving speed in different areas—and transmits them in real time to the insurer or aggregator. The insurer uses this data to create a comprehensive picture of the driver, his driving habits, and the conditions under which he drives, perhaps even his state of mind. This allows insurance premiums to reflect actual driving habits. The driver also has access to this data and can use to change his behavior with a potential reduction in premium. There are different names for this form of car insurance such as pay as you drive, pay how you drive, and usage-based premium. Soon sensor telemetry will move from the innovators to become

the norm in the insurance industry. Currently, adoption is restricted by such things as state and country local regulations, privacy considerations, drivers' willingness to be monitored, and other information sharing concerns.

Sensor telemetry is part of a larger movement called telematics. Telematics is the broad term for machine to machine communication and implies a bidirectional exchange of data between endpoints. The data is from sensing, measuring, feedback, and control.

4 Characteristics of Data

This section shows some properties of data that have a bearing on the architecture.

Characteristics of data differ with their type and storage format. Data can be stored in highly modeled form or in a form that lacks a model. There are strong data models for storing transactional data such as relational (ER) models and hierarchical models. There are weaker models for storing Weblogs, social media, and document storage. There are storage forms that lack a model such as text, social media, call center logs, scanned documents, and other free forms of data. The data model has a bearing on the kinds of analytics that can be performed on them.

1. Value Density

Value density or information per TB is a measure of the extent to which data has to be processed for getting information. It is different in different data forms. Transaction data has little extraneous data. Even if present, they are removed when they are stored in disk. Social media data on the other hand is sparse. Social media data is typically small amounts of interesting information within a large collection of seemingly repetitive data. Such data may contain the same thought or words multiple times. It may also contain thoughts and words outside the focus of discussion. Similar cases occur in other user produced data streams. Sensors produce data at fixed intervals and therefore may contain redundant data. The challenge is filtering the interesting information from the seemingly extraneous. Filtering useful data reduces the amount of data that needs to be stored. Such filtering occurs upfront in strong data model leading to higher information density. In weaker models, data is stored much closer to how they are produced and filtering occurs prior to consumption.

The information density or value density increases as data is processed and structured. Value density is a continuum that increases as it moves from unstructured and gains structure.

Value density of data has a bearing on storage ROI. For instance, low-cost storage may be appropriate to store low value density data.

2. Analytic Agility

Data organization affects the ability to analyze data. Analytics is possible on all forms of data including transactional and big data. Analytics on structured transactional data with strong data models are more efficient for business

analytics than analytics on unstructured or poorly structured data with weaker data models. However, analytics on weaker data models can lead to insights that were previously unknown, that is, new surprise moments are possible on weaker data models.

There are different types of analytic insights.

One is understand what happened and why. An organization may wish to identify why certain products sell better in one region versus another, or combine sales data with product placement data and marketing campaign strategy in order to determine whether a campaign can succeed, or find out whether a customer is likely to attrition by comparing his interactions with other customers' interactions with similar background and transaction history. Business analysts gain such new insight by finding new ways to traverse the data. In a relational model, this means joining new tables or joining tables in new ways and querying and combining tables in new ways. This is one form of predictive insight.

Another is to find new patterns in data or find the questions that are worth answering. The intent is to find new cause-effect relationships. For instance, knowing that product placement has an impact on marketing campaign, and knowing the factors that cause a customer to attrition away from the vendor, and knowing the factors that influence a customer to buy from a specific vendor or knowing how much influence each factor has on his buying decision are new patterns found in the data.

These two completely different types of insights depend upon the data structure and data organization. When structure is weak or nonexistent, a data scientist applies different learning techniques to reveal different patterns and different structures in the underlying data. Such patterns are further refined using more learning techniques. Such mining can uncover totally new insights. Storing data closer to their original form has the benefit of retaining seemingly noisy data but can become useful with new learning techniques.

3. Frequency and Concurrency of Access

Frequency and concurrency of data access is another characteristic that affects the data architecture.

Business users ask repeated questions of the same data. Thousands of such business users operate on the same data at the same time. Transactional data in strong models have high reuse and high concurrency. Transactional data modeled as strong models are more efficient for business uses.

Data scientists on the other hand ask mining kinds of questions on weakly structured data in order to sift through lots of data. Few data scientists operate on the same system at the same time when compared to business users. Schema-On-Read where the data is structured and a model is built each time the data is used is better for data scientists. Such users try out different models with each interaction in order to discover new structure in the data. Schema-On-Read, where no structure is presupposed, is easier with raw data storage. Data access for such interactions is typically low reuse and low concurrency.

High-reuse and high-concurrency forms of usage mean that it is acceptable to invest in efforts to structure the data. Low-reuse and low-concurrency forms of usage mean that it is acceptable to build structures each time the data is accessed.

One is better performing. The other is more flexible. One is appropriate for BI forms of analytics. The other is appropriate for big data forms of analytics.

Understanding these characteristics is critical to architecting the data solution. All data are important. Organizations cannot afford to throw away data merely because its value density is low. Mining it leads to valuable insight. This is a continuous process and there are quite a few different technologies for deep analysis of data (see Sect. 8.3).

Data has to be productized for broader impact. Data scientists perform investigations on raw data. In order to deliver more statistically accurate results, the data may be transformed to improve its quality, normalize the fields, and give it somewhat more structure for ease of further analysis by more users. Then as business insights are gleaned, the data may be further refined and structured to ease delivery of repeatable processes to deliver those insights. Finally, to integrate the data with the rest of the data in the enterprise, further refinement and structuring including extraction of attributes may be performed to add to traditional data models of the organization in order to enable widespread use of the derived results. At each stage, it is expected that the data size will reduce significantly. The reduction may be several orders of magnitude by the end of the phase, but it will have very large increase in usability by a general business user in the organization.

There are three distinct phases of data in a data-driven architecture—data capture, knowledge discovery, and knowledge usage. This is the focus of the next section.

5 Data-Driven Decision Making

Stephen Yu in his blog [3] divides analytics into four types:

- (a) BI. ROI, reporting, and dashboards come under this category.
- (b) Descriptive analytics. Dividing customers into segments and profiling profitable customers come under this category.
- (c) Predictive analytics. Future profitability of a customer, scoring models, and predicting outcomes such as churn come under this category.
- (d) Optimization. Optimization problems and what-if scenarios come under this category.

Data-driven decision making is about performing all these forms of analytics. The architecture must support all of them.

Gaining insight from data and making decisions based on that insight is the goal of data-driven decision making. Analytics creates information, understanding, and

finally insight, prediction, and foresight. Structure is core to analytics. Structure is either predetermined or it is built on the fly. The types of analytics on predetermined structures can be limiting. They are mostly derived from new ways of traversing the structure and combining entities in new and innovative ways to gain new insight. When structure is built on the fly new forms of insights are possible (see Sect. 4, Analytic Agility).

The velocity of big data requires that we capture it without loss. It should be transformed, cured, filtered, analyzed, validated, and operationalized in order to derive its value. Some forms of analytics may be possible to perform where the data lands. Some forms of analytics require structure. If the data fits one of the known structures, those structures are populated for use in every day analytics and decision making. Some forms of analytics require building structure on the fly. Such forms require learn–refine forms of analytic mentioned earlier. Deep analytics are possible using such forms and are performed to gather new structure. Such analytics create understanding of new patterns. All these require the management of three functional phases of data:

1. Data capture and storage. This is referred as data lake.
2. Discovery. This is referred as data R&D.
3. Productize. This is referred as data product.

Data lake is where data lands. It is the part that acquires and stores raw data. Some types of analytics are performed here. Data lake is alternately referred as data platform in this document.

Deep analytics on big data occurs at Data R&D. This is where insights hitherto unknown is gained.

Data product is where analytics such as BI, dashboards, and operational analytics on strong data model occur. This is also where every day analytics occur.

Our descriptions highlight the primary role of each phase. In real implementations, their roles and capabilities blur at the boundaries and are not as sharply differentiated as we have listed. All three aspects perform some form of analytic. The analytics vary in their depth. Data storage and data model have a bearing on the kind of analytics that are possible. The volume and variety of non-relational data have a bearing on where an analytic is performed. More types of predictive and optimization types of analytics occur in data R&D. BI and some forms of predictive analytics occur in data product. Some forms of descriptive analytics and predictive analytics occur in data lake.

Data goes through the following five stages as part of analyzing and gaining insight: acquire, prepare, analyze, validate, and operationalize.

Data lake acquires, stores, extracts, and organizes raw data. Some forms of reporting and some forms of descriptive analytics are performed here. Predictive and operational analytics are performed at both data product and data R&D. They differ on the forms of data on which they operate. Once structure is perceived in data then new structure can be created and loaded onto the data product portion. This in effect operationalizes the analytics. For data without strong structure or in

which structure is not apparent, structure has to be derived in order to perform analytics. The data R&D portion is used to create structure for analytics on big data.

Each of the three functional aspects is elaborated in the following sections.

5.1 Data Lake

Wikipedia defines a data lake as a “storage repository that holds a vast amount of raw data in its native format until it is needed.” It further says “when a business question arises, the data lake can be queried for relevant data, and that smaller set of data can then be analyzed to help answer the question.”

The above-mentioned definition means that data lake is the primary repository of raw data, and raw data is accessed only when the need arises. The accessed data is analyzed either at the data lake itself or at the other two functional points.

Raw data is produced at ever increasing rates and the volume of overall data an organization receives is increasing. These data must be captured as quickly as possible and stored as cheaply as possible. None of the data must be lost. The value density of such data is typically low and the actual value of the data is unknown when it first arrives. So an infrastructure that captures and stores it at low cost is needed. It need not provide all the classic ACID properties of a database. Until the data is needed or queried, it is stored in raw form. It is not processed until there is a need to integrate it with the rest of the enterprise.

Raw data must be transformed for it to be useful. Raw data comes in different forms. The process of raw data transformation and extraction is called data wrangling. Wikipedia [4] defines the process of data wrangling as “Data munging or data wrangling is loosely the process of manually converting or mapping data from one “raw” form into another format that allows for more convenient consumption of the data with the help of semi-automated tools. This may include further munging, data visualization, data aggregation, training a statistical model, as well as many other potential uses. Data munging as a process typically follows a set of general steps which begin with extracting the data in a raw form from the data source, “munging” the raw data using algorithms (e.g. sorting) or parsing the data into predefined data structures, and finally depositing the resulting content into a data sink for storage and future use.” It elaborates on the importance of data wrangling for big data as follows “Given the rapid growth of the internet such techniques (of data wrangling) will become increasingly important in the organization of the growing amounts of data available.” Data wrangling is one form of analysis that is performed in the data lake. This process extracts from multiple raw data sources and transforms them into structures that are appropriate for loading onto the data product and data R&D. In general, it cleanses, transforms, summarizes, and organizes the raw data before loading. This process is performed for non-transactional big data besides transactional data. It loads the data R&D for Weblog, social data, machine, sensor, and other forms of big data. The data lake or data platform increases the value density by virtue of the cleansing and transformation process. According to eBay, a

leading-edge big data user, “cutting the data the right way is key to good science and one of the biggest tasks in this effort is data cleaning” [5].

Some of the insights gained here from analysis are stored locally in the data platform and others are loaded onto the data product or data R&D platforms. In order to perform analysis, some form of descriptive metadata about the raw data should also be available in the data platform. Metadata describes the objects in the various systems and applications from where data is collected, much like a card index in a library. Metadata is needed for analyzing the raw data.

Hadoop is the platform often associated with the data lake or data platform. Data is stored in clusters of low-cost storage. Hadoop uses a distributed file system for storage and is called the Hadoop distributed file system (HDFS). The storage format is key-value pairs. It stores data redundantly and is fault-tolerant. Data is processed in clusters of low-cost nodes using MapReduce which is a highly scalable processing paradigm. Hadoop provides horizontal scaling for many kinds of compute-intensive operations. Therefore, many kinds of analytics are possible in data platform.

The performance metric for the data platform is TB per sec of data capture capacity and dollars per TB of data storage capacity.

5.2 Data Product

Insight gained from analytics is productized for strategic and tactical decision making. An enterprise data warehouse (EDW) is the platform often associated with data product. Reports, analytical dashboards, and operational dashboards with key performance indicators are produced at the warehouse. These provide insight into what happened, why it happened, and what is likely to happen.

Data arrives at the warehouse from the data platform categorized and in a form to fit the relational model structure. This structure presupposes how the data is going to be analyzed. This is unlike how data arrives at the data platform.

The data storage structure has a bearing on the insight derivable from data (see section on analytic agility). Data product provides insight on relational forms of storage. Insight comes from traversing the data in complex and innovative ways. The extent to which such traversal is possible is the extent to which new insight can be gained from the data. Such insights include predictive analytics. For instance, it can determine whether a customer has a propensity toward fraud or whether a transaction is in reality a fraudulent transaction. Such analyses require among other things an understanding of the card holders' previous transactions, an understanding of the transactions that typically end up as fraudulent transaction and the conditions under which they become fraudulent. All this goes in deciding whether to approve the current card transaction. Similarly, predicting the success of a marketing campaign is possible depending on the analysis of previous campaigns for related products during related times of year. Similar examples exist in other

domains. Innovativeness of analysis drives the amount of insight that can be derived from data.

Performance, efficient execution of queries, and availability are key requirements of data product platform. ACID property is necessary for such data since this is data repository of record.

Throughput and response time are the metrics and are measured in queries per second, dollars per analytic, and seconds per query.

5.3 Data R&D

Data R&D provides insight on big data. Insight also means knowing what questions are worth answering and what patterns are worth discerning. Such questions and patterns are typically new and hitherto unknown. Knowing such questions and patterns allows for their application at a later time including in the data product portion. Therefore, data R&D is what creates intelligence through discovery for the organization on new subject matters. It also refines and adds to previous insight. For example, it is a new discovery to determine for the first time that card transactions can at times be fraudulent. A refinement of this discovery is to recognize the kinds of behavior that indicate fraud. In other words having the intelligence to ask the question “can transaction be fraudulent?” is a big learning. Determining the patterns that lead to fraud is another form of learning.

Data R&D is the primary site where deep analytic occurs on big data. Model building is a key part of deep analytics. Knowledge or insight is gained by understanding the structure that exists in the data. This requires analysis of different forms of data from different sources using multiple techniques. This portion has the ability to iteratively build models by accessing and combining information from different data sources using multiple techniques (refer Sect. 7). It also constantly refines such models as new data is received.

There are multiple techniques and technologies in the R&D space. The ability to perform analytic on any type of data is a key requirement here.

The metric for this functional aspect of data is dollars per analytic performed and dollars per new pattern detected.

Analysis of big data goes through these stages—acquire, prepare, analyze, train, validate, visualize, and operationalize. The data platform acquires data and prepares it. The analytic part in R&D recognizes patterns in big data. Many big data analytics use correlation-based analytic techniques. A correlation among large volume of data can suggest a cause–effect relationship among the data elements. Analyzing large volumes of data gives confidence that such patterns have a high probability of repeating. This is the validate part which ensures that the patterns recognized are legitimate for all different scenarios. The train part is to automate the pattern recognition process. The operationalize part is to put structure around the data so they can be ready for high-volume analytics in the product platform.

6 Deriving Value from Data

Deriving value from data is not about big data alone. It is about capturing, transforming, and dredging all data for insight and applying the new insight to everyday analytics.

Big data analysis has been about analyzing social data, Web stream data, and text from call centers and other such big data sources. Typical approaches build independent special systems to analyze such data and develop special algorithms that look at each of these data in isolation. But analyzing these data alone is not where the real value lies. Building systems that go after each type of data or analyzing each type of data in isolation is of limited value. An integrated approach is required for realizing the full potential of data.

Analyzing without a global model or context does not lead to learning. There is definitely some learning that exists in such analysis however. Social Web data or textual entries from a call center contain extremely valuable information. It is interesting to analyze a Web clickstream and learn that customers leave the Web site before they complete the transaction. It is interesting to learn that certain pages of a Web site have errors or that certain pages are more frequently visited than others. It is interesting to learn that a series of clicks in a specific order is predictive of customer churn. It is interesting to learn how to improve user experiences with a Web site. It is interesting to learn in depth about the behavior of a Web site. It is interesting to learn from graph analysis of the social Web the influencers and followers. There is value in all this. However, this value is far less than the value that is derived when it is combined with other data and information from other sources from the rest of the organization. This kind of Metcalf's law applies to data—connecting data sources and analyzing them produces far more value than analyzing each of the data sources alone. Combining the behavior of a customer on the Web site with what is already known about the customer from his transactions and from other interactions provides more insight about the customer. For instance, a call center conversation shows a trend in how people talk to each other but combining information about the customer on the call and information about the call center representative on the call (with the customer) adds context about both parties and shows what kind of representative has what kind of effect on what kind of customer. This is more valuable insight than how people talk to each other. Similarly combining product experiences expressed on the Web with product details, suppliers' information, and repair history gives a better understanding of the views expressed in social media. Analyzing social media such as Twitter and Facebook can reveal an overall sentiment trend. For instance, it can tell how people feel about a product or company. It can tell about whether people are complaining about a product. Such information in isolation is interesting but lightweight in value. Linking the active people on the social media with the customer base of the company is more valuable. It combines customer behavior with customer transactions. The sentiments become yet another attribute of a known customer and can

be analyzed holistically with other data. With a holistic view, social Web chatter becomes tailored and more actionable.

Adding global context to any isolated analysis increases the value of such analysis significantly. This is true for transaction analysis. This is true for raw data transformation and cleansing. This is true for big data analysis. For instance, analyzing sensor data with context from patient information and drug intake information makes the sensed data more meaningful.

Therefore, deriving value from big data is not about analyzing only big data sources. It is about reasoning with all data including transactions, interactions, and observations. It is myopic to build an environment for analyzing big data that treats big data sources as individual silos. A total analytic environment that combines all data is the best way to maximize value from big data. This is one of the key requirements for our big data architecture—the ability to reason with all data.

The three functional components we discussed earlier—Data lake, data R&D, and data product—must all be combined for any analysis to be complete. Therefore, reasoning with all data is possible only if the three aspects of data are integrated. We call this integrated analytics.

6.1 The Three Functional Components at Work

The discovery process puts all these three functional components of data-driven decision making together through integrated analytics.

Sales listing applications, barter listing applications, or auction item listing applications are examples of integrated analytics. Such applications determine, through analysis, that a number of different attributes of an item listed for sale impact a buyers' purchasing decision. The goodness metric on each of these attributes affect the order of how the sale items are listed in a search result. For instance if multiple parties list similar products, the search engine would return and list the results based on ranking of these attributes. It is advantageous for better buyer response to have items listed in the beginning of the list rather than lower down in the list. Therefore, sellers prefer their items in the beginning of the list.

There are three functions that are necessary in this application. One, determining the list of attributes that users consider important. This means finding out what attributes impact customer choices. This is a deep analytic. Two, rating the quality of each attribute for each item being listed. Attributes may be objective such as price and age. Attributes may be subjective such as product photograph, product packaging, and product color. Analysis is required to rate the quality of each attribute. Such analysis is harder for subjective attributes. Subjective items such as photograph quality and packaging quality should be analyzed on the basis of background, clarity, sharpness, contrast, color, and perhaps many others. Data platform performs these kinds of analytic using MapReduce. Three, combining all

these to determine the listing order. The data product considers the rating of each of the attribute in order to determine the listing order of each item for sale when a search is performed.

7 Data R&D—The Fertile Ground for Innovation

Data R&D is the challenging part. The other two parts are fairly straightforward. Hadoop clusters meet the requirement of the data platform. Hadoop clusters are scalable and are the preferred solution for the data platform. They can capture all forms of data from an array of sources and store them with redundancy and high availability. They can also do the forms of analytics mentioned earlier using MapReduce. An EDW meets the requirement of data product. Parallel RDBMS can meet the complexity and performance requirements.

The challenge is the middle part or data R&D. Neither MapReduce alone nor relational technology alone is appropriate for this phase. RDBMS requires the data to fit into well-defined data models. Hadoop clusters require super human programmers to program new applications.

Figure 4 shows this challenge.

Data R&D is the focus of a number of innovations and new start-ups. Technologies such as graph engine, text analytics, natural language processing, machine learning, neural networks, and new sensor models are being explored for big data. There are four broad types of efforts currently underway:

1. Bottom-up approaches. These provide relational database-like aspects on top of Hadoop. They add SQL or SQL-like features to Hadoop and HDFS. Some leading Hadoop vendors and Hadoop open source groups fall under this

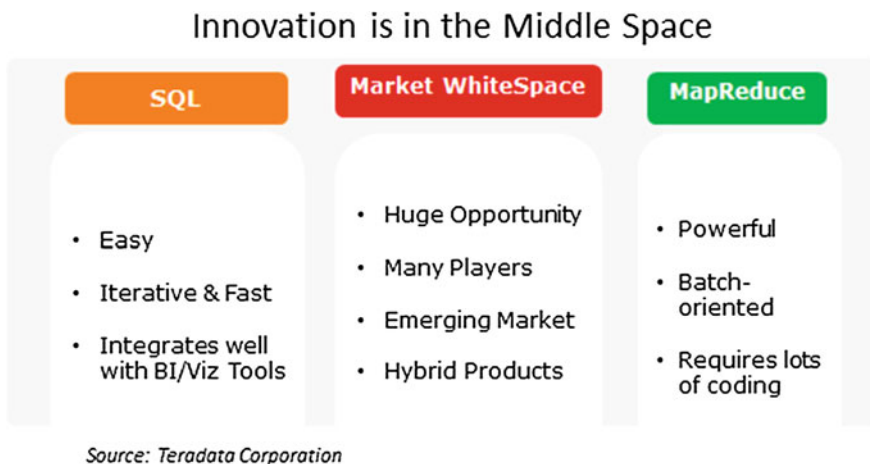


Fig. 4 Innovation rich R&D environment

category. Examples include Apache Hive [6], Cloudera® Impala [7], Hortonworks® Stinger [8], Apache HBase [9] among others. The last one is not relational, but the idea is to provide classical database-like features and consistency property for sparse data.

2. Top-down approaches. This approach starts with a mature SQL engine and integrates other capabilities as callable facilities. SQL is used as a kind of scripting-like facility in order to orchestrate access to other stores. They embed other technologies such as natural language processing functions, pathing functions, and others as SQL callable functions using existing frameworks such as UDFs or through proprietary frameworks. For some technologies, it creates applicable storage natively. Many leading database companies are doing this approach. Besides call-outs, some support other storage and processing technologies natively.
3. Application-specific approach. These are specific solutions targeted toward a specific application. They create targeted technologies and solutions specific to a problem. These may be on top of Hadoop in some cases. Splunk® [10] is one successful example of this approach. It is a specific solution for collecting, analyzing, and visualizing machine-generated data. It provides a unified way to organize and extract real-time insights related to performance, security, and privacy from massive amounts of machine data generated across diverse sources.
4. Technology-specific approach. This is similar to the above-mentioned third category but is technology-specific rather than application-specific. These also target specific problems. Graph engines, path analytics, machine learning, and natural language processing engines are some examples. There are platforms available for this approach from open source and industry.

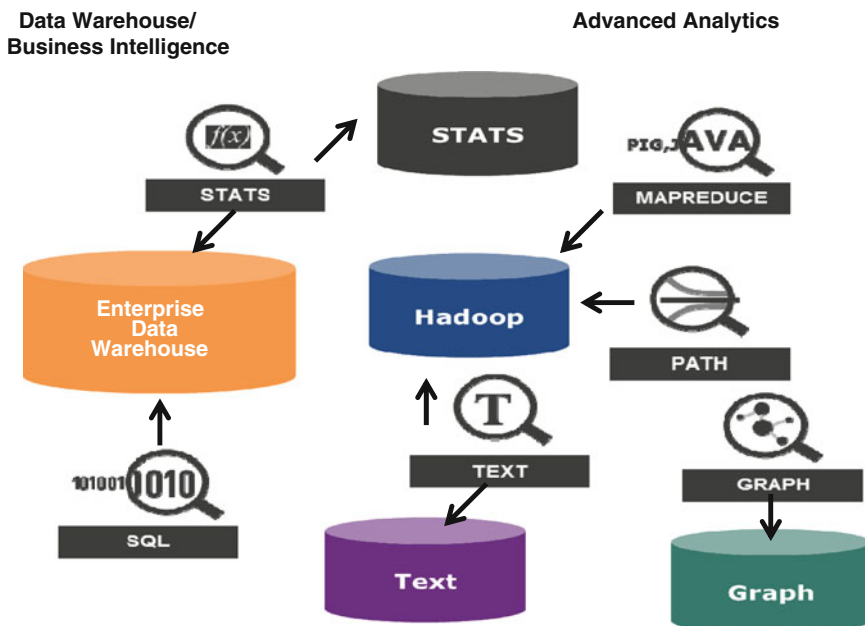
Specific solutions such as above-mentioned items 3 and 4 are appropriate as stand-alone solutions. They often target a specific problem. They are the best of breed in their area. Splunk® [10] is very successful in analyzing machine logs. SPARQLVerse [11] is good as a traversal engine for graphs representations. Depending on the need, some products and solutions are ideal for a problem but their broad applicability is limited. Therefore, we recommend such solutions only if the problem is confined to what the solution can address.

The choice for data R&D architecture is a choice between the first two approaches. Adding SQL maturity to HDFS seems like a significant effort. This approach assumes all future technology evolutions for the different forms of big data analytics use HDFS.

The SQL foundation approach seems broader in that it attempts to go beyond SQL and integrate MapReduce, HDFS, and other technologies. This is important when big data space and the technologies and algorithms that operate on big data are evolving.

The following picture shows the chaotic evolution for knowledge discovery.

Figure 5 shows stand-alone solutions that target specific problems. This is acceptable if the problem is limited to the application of one such solution.



Proliferation of advanced analytics environments has resulted in fragmented data, higher costs, expensive skills, longer time to insight

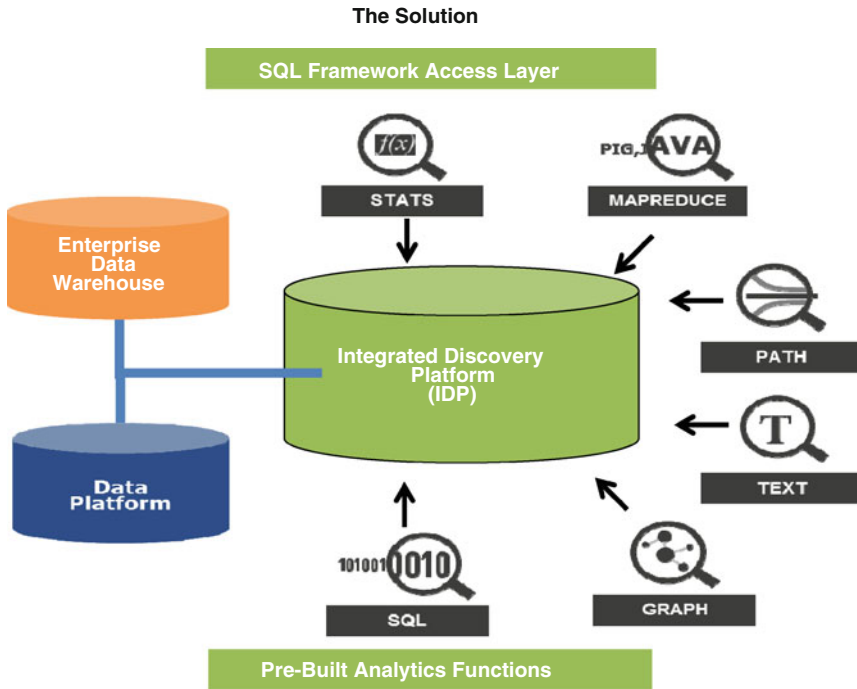
Source: Teradata Corporation

Fig. 5 Bunch of stand-alone solutions is a problem

However, creating insight is an iterative approach where multiple techniques must be applied one after the other. For instance sessionization, followed by text analytics, followed by sentiment analysis, followed by pathing must be applied in order to understand customer behavior across multiple channels. Each technology application refines the input data set and the refined set is fed to another technique. This process continues until insight is gained or structure is found in the data. Such refinement may include reaching out to raw data storage and to SQL stores besides applying multiple big data technologies.

Applying each of the techniques as free-standing techniques repeatedly on an iteratively refined data set requires highly skilled programmers. It is also human intensive when more than a few techniques have to be applied on the same problem. For example, repeated application requires data files to be produced and formatted before each new technique is applied. It is not a scalable solution. By definition, insight creation is a trial-and-error process. The number of application of such trials reduces when each is hard to apply. An integrated mechanism is therefore needed.

Figure 6 shows such an integrated approach. The discovery platform integrates other technologies.



Integrated discovery analytics provides deeper insight, integrated access, ease of use, lower costs, and better insight

Source: Teradata Corporation

Fig. 6 Integrated discovery platform

The discovery platform starts with a mature SQL engine which acts as a discovery hub operating on mixed data—big data, HDFS data, relational data, and other stores. It has different forms of local store such as a graph store besides SQL store. It uses the SQL language as a kind of scripting facility to access relational store and other stores through UDF and proprietary interfaces. The SQL engine provides the usual SQL platform capabilities of scalability and availability. The platform can also execute MapReduce functions. The discovery platform may natively implement big data analytic techniques such as time series analysis and graph engine. The platform may also natively support other data stores such as HDFS and graph storage. It may also access stand-alone techniques such as machine learning to return results that are refined by that technique.

The industry can encourage this form of discovery through standardization efforts for interaction across platforms for data and function shipping. Currently, these are mostly ad hoc and extensions are specific to the vendor.

8 Building the Data Architecture

Reasoning with all data is possible only if the three functional aspects of data are interconnected.

Earlier, we specified three functional aspects of data. These aspects evolved data from raw to a product by undergoing R&D. These aspects were mapped to a platform:

1. Data lake or data platform. Raw data lands here and is stored. This platform has the capacity and cost profile to capture large volumes of data from varied sources without any loss or time lag. This is typically a Hadoop cluster.
2. Data product. BI, analytical dashboards, and relational forms of predictive analytics happen here. This platform supports high reuse and high concurrency of data access. It supports mixed workloads of varying complexity with large number of users. This is typically a parallel EDW platform.
3. Data R&D. Deep analytics from big data happens here. This platform determines new patterns and new insights from varied sets of data through iterative data mining and application of multiple other big data analytic techniques. The system combines a number of different technologies for discovering insight. This is the integrated discovery platform with tentacles that reach into other stand-alone technology engines besides implementing multiple technologies and multiple data stores natively.

Data lake is satisfactory for stand-alone applications. For example, a ride-sharing company can create a stand-alone booking application in the data lake. The booking application connects passengers to drivers of vehicles for hire. Customers use mobile apps to book a ride, cancel a ride, or check their reservation status and other similar operations. Such applications have a high number of users and a high number of passenger interactions. Such applications can be implemented as a stand-alone entity in the data lake using Hadoop. The application is scalable in terms of storage capacity and processing capacity.

An integrated analytic is, however, required if we want to go beyond stand-alone applications. Such analytic is more valuable. For example, an integrated analytic that reaches elsewhere for data is required if the ride-sharing company wants to consider as part of the booking application other information such as vehicle maintenance records, customer billing records, and driver safety records. These kinds of information usually come from other places in the organizations chiefly from the EDW. Integrated analytics require combining data lake and data product. If in addition the analytic wants to understand driver routes and driving patterns and to perform graph analytics on driver movements and location of booking data, R&D becomes a part of this integration with data lake and data product.

This requires an architecture that integrates all three platforms as shown in Fig. 7.

The focus of this architecture is the data layer and everything else is pushed off to the side. A complete architecture needs all the access tools, data acquisition tools, and all visualization tools. It requires multiple mechanisms to transport and load

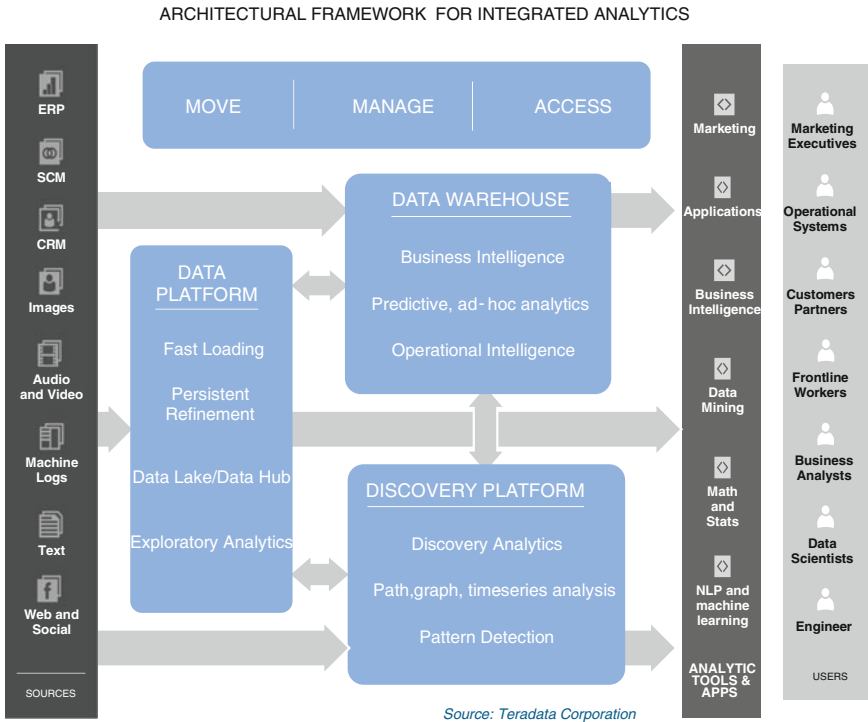


Fig. 7 Integrated analytics architecture

data in parallel. It requires all the client tools that interact with each of these three platforms. Some of these facets of the architecture are merely shown as small boxes but are not elaborated or explained in this picture. Each of these is an entire technology and outside the data architecture portion of this chapter.

Data connections between the three data storage platforms (lines and arrows in this chart) are more important than the data stores. Data stores are an acceptance that there must be different kinds of platforms with different kinds of capabilities and different kinds of attributes to build an overall successful analytic solution. These were elaborated in the previous sections of this chapter. These evolve over time. The arrows show the need for their interaction. As observed earlier, the industry is doing very little by way of standardization in these interaction areas. It is left to individual vendors to evolve their own solution.

Unlike transactional data, the value of big data is often unknown when it initially arrives or is accepted for storage. It is not possible to know all the patterns the data exhibits unless we know in advance all the questions that are worth answering from that data. New questions may come at any time in the future. Therefore, organizing and storing the organized data is not very useful for such analytics. Organizing it upfront loses some of the values that are in the original data even though such value may be unknown at the time of storage.

8.1 Data Platform or Data Lake

The data platform is the data landing zone. All new forms of raw data arrive at this platform for storage and processing. The goal is get the data in, especially new types of data, quickly, and directly as possible; land it in the cheapest storage; and have a platform to efficiently filter, preprocess, and organize the landed data for easier analysis. The primary point of the data platform is to deal with low value density data and increase the value density.

Hadoop is the platform of choice here. It has a high degree of parallelism and low cost of horizontal scaling. MapReduce is agile and flexible. It can be programmed to perform all forms of data transformations. In addition, an increasing number of tools are available to work on data in Hadoop for improving data quality and value density. These range from traditional extract–transform–load (ETL) tools expanding their footprints to new data wrangling tools that are developed specifically for this space.

An example analytic that was covered earlier was to score and rate subjective attributes such as photographs and packaging for their quality for items listed for sale.

Organizing and transforming are forms of analytics that improve value density. Organizing means for instance adding cross-referencing links between store ID and customer ID. Sessionizing is an example of organizing data such as Web records in order to infer that all the Web records are part of a particular conversation with a user. This is a highly resource-intensive and complex operation especially since volumes are high and session logs interleave many different users.

Sessionization is a common form of organization for customer-related interactions. A customer may interact from multiple touch points. Making a chronological order of those interactions organizes the data for analysis. Figure 8 shows customer interactions across multiple channels.

Another example of organizing is text analytics where say four encoded attributes from call center interaction text are extracted and forwarded to the customer record. For example, this customer complained about this problem about this product on this day could be the four attributes. The general goal is to structure the data or add a bit of structure to the data for ease of understanding.

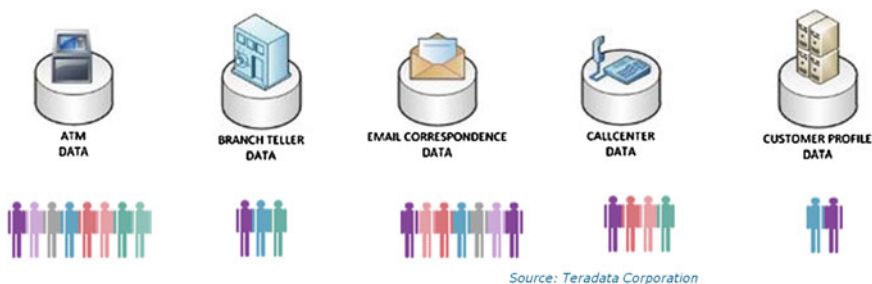


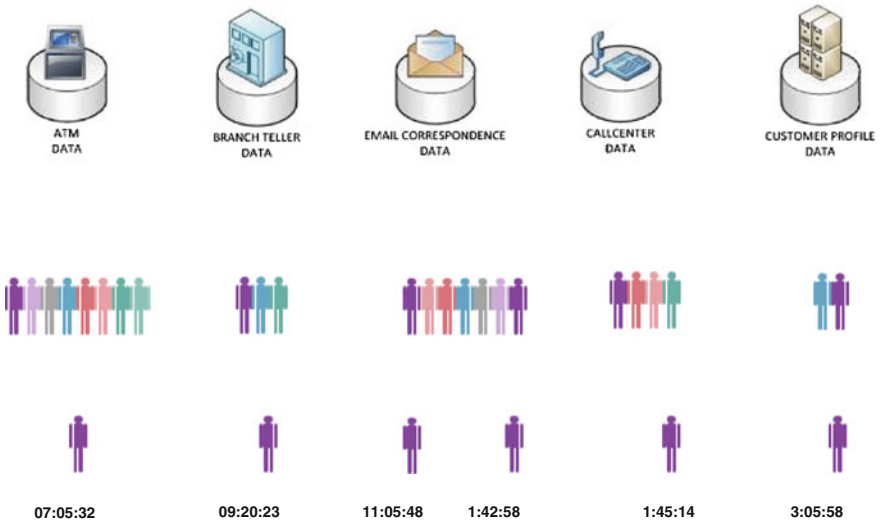
Fig. 8 Sessionization—interaction across different touch points

Figure 8 shows the sessionization process. It shows the importance of this process for all social interaction-related analytics.

Figure 8 shows four example channels of customer interactions—ATM, Teller, e-mail and written correspondences, and call center. It also shows customer profile data.

Many customers arrive at each of these interaction touch points. Many may visit the same touch point more than once at different times. This chaos has to be sorted. The touch point each customer visits is important. The order in which they visit each touch point is important.

Figure 9 shows the result of the sessionization process. Wikipedia [12] defines sessionization and elaborates it this way: “Sessionization is a common analytic operation in big data analysis and is used to measure user behavior. Behavioral analytics focuses on how and why users of eCommerce platforms, online games and web applications behave by grouping certain events into sessions. A sessionization operation identifies users’ web browsing sessions by storing recorded events and grouping them from each user, based on the time-intervals between each and every event. Conceptually, if two events from the same user are made too far apart in time, they will be treated as coming from two browsing sessions. A session can be as short as a few seconds or as long as several hours.” This Wikipedia definition should help in understanding Fig. 9. Basically, sessionization extracts the series of customer interactions and their order. A customer may come from a single session or from multiple sessions. If the interactions are too far apart in time, they may be considered different sessions. They may come from different IP. Therefore, studying the session ID, the time sequence, and time gap are



Source: Teradata Corporation

Fig. 9 Sessionized result

all important to determine and stitch together the customer interactions in time order. The customer profile and transaction data are integrated with the sessionized data. The integration of all this data along with path analysis determines the customer interactions that led to the final outcome. Combined with the profitability of the customer, necessary action can be taken. This example shows multiple technologies being combined: sessionization, pathing, and relational databases.

ETL is another form of organization that is also performed on the data platform. ETL applies to both big data and transactional data. ETL on big data includes operations on raw data such as structured and semi-structured transformations, sessionization, removing XML tags, and extracting key words.

Archival is another key operation performed at the data platform. The data landing zone has different retention periods compared to older architectures. In the past, the landing zone was a very transient place where structure was added through ETL processes and then the landed data was destroyed. Only the structured data was retained for regulatory and other archival purposes. The landing zone in this architecture is where raw data comes in and value is extracted as needed. The raw data is archived.

There are privacy, security, and governance issues related to big data in general and the data platform in particular. Access to raw data cannot be universal. Only the right people should have access. Also data life cycle management issues about when to destroy data and the considerations for doing that are important. There are also questions about how to differentiate between insights derived versus third-party data and how to protect each of these. Some of these issues are covered in other places in this book. Some are outside the scope of this chapter. They are mentioned here to indicate these have to be considered in the overall architecture discussions.

8.2 Data Product

Parallel RDBMSs are the platform of choice for the data product platform. BI, dashboards, operational analytics, and predictive analytics are all performed here. Such analytics are based upon relational forms of data storage. Profitability analysis is an example of the kind of predictive analytics this platform performs. The profitability of a customer is based on revenues from the customer versus costs associated with maintaining the relationships with the customer in a specific period. Future profitability is predicted by associating customer characteristics and transaction patterns over a period with other known customers' and their characteristics. Profitability is also predicted during customer acquisition. Questions such as whether the customer will be profitable and is he worth acquiring are answered here. These are not covered further in this chapter.

As mentioned earlier, big data analytics is a combination of analytics using MapReduce on flat files and key-value stores, SQL on relational stores, and other forms of analytics on other forms of storage. Different analytics are combined for discovery in the discovery or R&D platform. Varieties of different technologies are

involved in this area. Innovation is also constantly changing this space. The discovery process is different from the discovery or R&D platform. If the problem is SQL tractable and the data is already in the warehouse, then discovery is done in the EDW. If a particular algorithm is appropriate on Hadoop and HDFS, then discovery is done in the Hadoop platform. The discovery process is done in the discovery platform when big data and SQL have to be combined with SQL, MapReduce, and others technologies such as graph engines, neural nets, and machine learning algorithms.

8.3 Data R&D or Data Discovery Platform

The discovery or R&D platform is one of the key pieces of the architecture for a learning organization. New insights gained here are used by other components to drive day to day decision making. It has the capabilities and technologies to derive new insights from big data by combining data from other repositories. It has the capabilities to apply multiple techniques to refine big data.

Health care is an example of such capabilities. Big data analytics are applied successfully in different care areas. One example application identifies common paths and patterns that lead to expensive surgery. Another application reduces physician offices' manual efforts for avoiding fraud and wasteful activities. These applications combine different kinds of analytics. They pool data from all three platforms. Data sources include physician notes, patient medical records, drug information, and billing records. Analytic techniques applied include fuzzy matching, text analytics, OCR processing, relational processing, and path analytics.

In industries such as insurance, sessionation and pathing techniques are combined to understand paths and patterns that lead to a policy purchase from a Web site or analyze customer driving behaviors for risk analysis and premium pricing.

In aircraft industry, graph analytics, sessionization, and pathing are combined to analyze sensor data along with aircraft maintenance records to predict part failure and improve safety of aircrafts.

As we saw earlier, interactions are much higher in volume than transactions. It is possible to predict the value of a customer from his transactions. It is possible to understand and predict behavior of a customer from his interactions. Predicting behavior is more valuable insight. Data analytic techniques on different forms of user interactions make such predictions possible. There are different techniques to predict behavior.

Clustering is an analytic technique to predict behavior. Collaborative filtering and affinity analysis techniques fall under this category. The idea is that if a person A has tastes in one area similar to a person B, he is likely to have similar tastes in another area. Market basket analysis and recommender systems are examples of this form of analytics. "People who bought this also bought this other item" and "You may be interested in viewing this profile" are some examples of recommendations based on behavior. Combining a customers' historical market basket with the

shopping patterns of “similar” buyers can provide personalized recommendations to the customer. There are also time-ordered collaborative filters which consider the order in which items are likely to be put into the basket. These kinds of analytics combine time series analysis and affinity analysis. There are other forms of cluster analysis such as k -means which clusters data such as customers into a specified number of groupings, and canopy which partitions data into overlapping subsets. Each of these techniques requires a chapter of its own and is therefore outside the scope of this chapter.

Statistical analysis is another class of big data analytics. Analytics such as correlation [13] which determines the strength of relationships between different variables, regression which is used to forecast and predict the future based on past observations, classification which identifies the set of population to which a new observation belongs such as patients based on biological readings, and many more are part of this technique.

Text analytics are another class of big data analytics. Sentiment analysis [14], which classifies user statements as positive or negative, and text categorization, which is used to label content such as e-mail as spam, and many more are part of this technique.

Graphs are a technique for modeling relations in any field [15, 16]. In social graphs, they are used to measure a persons’ prestige and his ability to influence others. In government, they are used to identify threats through (a) detection of non-obvious patterns of relationships and (b) group communications in e-mail, text, and telephone records. In health care, it is used to detect drug efficacy and outcome analysis. Graph analytics are flexible. It is possible to add to an existing graph incrementally. For instance, it is possible to add new data sources and new relationship to an existing graph in order to support new lines of inquiry. Graph analytics are highly mathematics oriented. Some common graph analysis functions include centrality, pathing, community detection, direction of relationships, and strength of relationships. These are also outside the scope of this chapter. The reader is left to deduce what these analyses imply from the name of the technique. They are mentioned here to convey the kinds of deep analytics on big data that occur at the discovery platform.

9 Putting It All Together

This section shows a use-case that implements our data architecture. It is a big data analytic for preventive aircraft maintenance and for providing part safety warnings to airlines.

Figure 10 shows integrated analytic with all three platforms. The arrows in this picture are data flow and are not indicative of process or control flow.

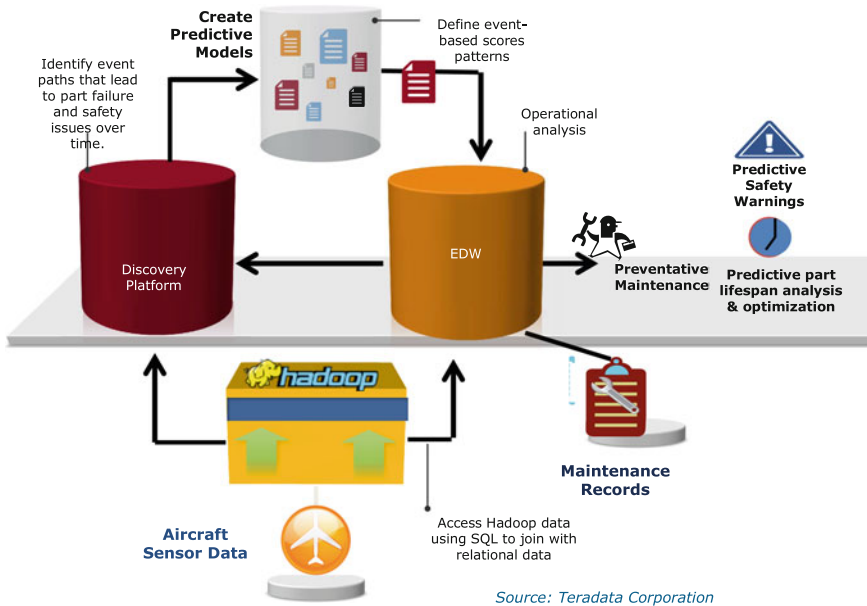


Fig. 10 Data refinery—refining raw data to actionable knowledge

The overall flow is as follows. Sensor data arrives at the data platform. This platform reaches out to the data product and data R&D platforms. Sessionized data is sent to the data R&D platform. The R&D platform models the part and reaches out to the data product and data platforms for information. The product platform performs BI analytics and sends out warnings. It reaches out to the data R&D and data platforms for supporting information.

Each system reaches out to the other for data that is needed in order to make sense of its patterns.

Sensor data are received and analyzed in the data platform (Data lake). However, if such analysis is done in isolation, then it is unwise to execute on it. Sensor data information itself is insufficient. The sensor readings must be organized. It must be sessionized for a time sequence of readings. The discovery platform puts the readings in the context of a part. The part is modeled in order to understand its behavior. This requires knowledge of the part and other sensor readings. These different readings are organized in order to understand part behavior. Some of this is done where the data lands. Most of it is done in the discovery platform (data R&D). Path analysis and statistical analysis are some of the analytical capabilities used to model and identify the pattern of failures of the part.

It is not enough to determine the current state of the part in the aircraft. Information is also needed about when the part was last maintained, the history of the part's behavior on this aircraft, history of the part's behavior in other aircrafts (i.e., the global database), the serial number of the part on this aircraft, and a host of other information. In addition, part advisories from the manufacturer are also needed from documents and other manufacturers release materials. Natural language analytic [17], text analytic [18, 19], and SQL are some of the other techniques used for analysis. The end result of this knowledge refinery is the issuance of a preventive maintenance direction and safety warning for failing parts in order to keep the plane flying safely.

Figure 10 is like a data refinery. Raw data enters the data platform. Interesting stuff is extracted from it and passed for discovery. Data R&D models with enterprise data, transaction data, and dimensional data from the operations side. It is passed to data product which produces the refined output and issues the warnings. Learning from this refinery is used to enhance the product platform for future automated and large-scale application of knowledge.

10 Architecture Futures

We see the future of integrated analytic evolving to satisfy two goals:

- (a) The integrated analytic refinery shown in the previous section should be projected as a single system for interaction and control. This means that the data lake, data product, and data R&D are treated as a single integrated entity for control, load, and querying.
- (b) The maturity, robustness, and predictability capabilities of EDW should be made available on the integrated entity.

These two goals translate into the following enhancements:

1. A global front end for analytic that optimizes user queries and interactions across all three systems.
2. A data loader that determines where objects should reside. Such decisions are based among other things on different properties of data such as data temperature, frequency of data usage, and frequency of analytic types.
3. A work load manager that guarantees metric of analytic refinery as a whole. Metric such as dollars per analytic performed, throughput, and response time is managed for delivery by the work load manager.

Acknowledgments The author wishes to acknowledge Todd Walter, Teradata Distinguished Fellow, for his many discussions and ideas on this topic. Todd has been a reservoir of valuable information and has been willing to freely share it.

The views expressed in this chapter are the authors' own and do not reflect Teradata corporations' views or ideas.

Questions

1. What is the primary intent of big data?
2. What forms of data constitutes the big data?
3. What are the 4Vs of big data?
4. Can a data warehouse be considered a solution for big data system?
5. Can Hadoop be considered a solution for big data system?
6. Define big data.
7. What caused the large data volume increases? In what order did the volume increases occur?
8. What forms of data are called interaction data?
9. What forms of data are called observation data?
10. How do the 3Vs relate to each other?
11. What are the data characteristics that must be considered for a successful big data architecture? Explain each of them.
12. List the different forms of analytic and write briefly about each of them.
13. What are the three functional phases of data for a big data architecture? Give a brief description of each of them.
14. What are the performance metric of each functional phase of data in a big data architecture? Give a brief analysis of each of this metric and why they make sense.
15. What are the aspects of the big data architecture. Discuss each of them.
16. Discuss some forms of data R&D analytic techniques.
17. Discuss an application of big data architecture.
18. Describe the kind of functions that are performed in each of the three data stores of the architecture with examples.

References

1. <http://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf>
2. <http://anlenterprises.com/2012/10/30/ibms-4th-v-for-big-data-veracity>
3. <http://www.infogroup.com/resources/blog/4-major-types-of-analytics>
4. http://en.wikipedia.org/wiki/Data_wrangling
5. <http://petersposting.blogspot.in/2013/06/how-ebay-uses-data-and-analytics-to-get.html>
6. <https://hive.apache.org/>
7. <http://www.cloudera.com/content/cloudera/en/products-and-services/cdh/impala.html>
8. <http://hortonworks.com/labs/stinger/>
9. <http://hbase.apache.org/>
10. http://www.splunk.com/en_us/products/splunk-enterprise.html
11. <http://sparqlcity.com/documentation/>
12. <http://en.wikipedia.org/wiki/Sessionization>
13. Liu, B., Wu, L., Dong, Q., Zhou, Y.: Large-scale heterogeneous program retrieval through frequent pattern discovery and feature correlation analysis. In: IEEE International Congress on Big Data (BigData Congress), 2014, pp. 780–781, 27 June–2 July 2014

14. Park, S., Lee, W., Moon, I.C.: Efficient extraction of domain specific sentiment lexicon with active learning. *Pattern Recogn. Lett.* **56**, 38–44 (2015). ISSN:0167-8655, <http://dx.doi.org/10.1016/j.patrec.2015.01.004>
15. Chui, C.K., Filbir, F., Mhaskar, H.N.: Representation of functions on big data: graphs and trees. *Appl. Comput. Harmonic Anal.* Available online 1 July 2014, ISSN:1063-5203, <http://dx.doi.org/10.1016/j.acha.2014.06.006>
16. Nisar, M.U., Fard, A., Miller, J.A.: Techniques for graph analytics on big data. In: *BigData Congress*, pp. 255–262 (2013)
17. Ediger, D., Appling, S., Briscoe, E., McColl, R., Poovey, J.: Real-time streaming intelligence: integrating graph and NLP analytics. In: *High Performance Extreme Computing Conference (HPEC)*, IEEE, pp. 1, 6, 9–11, Sept. 2014
18. Atasu, K.: Resource-efficient regular expression matching architecture for text analytics. In: *IEEE 25th International Conference on Application-specific Systems, Architectures and Processors (ASAP)*, pp. 1, 8, 18–20, June 2014
19. Denecke, K., Kowalkiewicz, M.: A service-oriented architecture for text analytics enabled business applications. In: *European Conference on Web Services (ECOWS, 2010)*, pp. 205–212. doi:[10.1109/ECOWS.2010.27](https://doi.org/10.1109/ECOWS.2010.27)

Big Data

A Primer

Mohanty, H.; Bhuyan, P.; Chenthati, D. (Eds.)

2015, XIV, 184 p. 48 illus., 42 illus. in color., Hardcover

ISBN: 978-81-322-2493-8