**NIST Big Data**

**Definitions and Taxonomies**

**Version 0.9**

**Definitions & Taxonomies Subgroup**

**NIST Big Data Working Group (NBD-WG)**

**February, 2014**

Executive Summary 3

1 Introduction 3

1.1 Objectives 3

1.2 How This Report Was Produced 3

1.3 Structure of This Report 4

2 Big Data Definitions and Taxonomies 4

2.1 Big Data Definition 4

2.2 Data Science Definition 6

2.3 Big Data Taxonomy 9

2.4 Actors 9

2.5 Roles and Activities 9

2.5.1 Roles potentially external to the reference architecture 10

2.5.2 System Orchestrator – provides requirements for the system. 11

2.5.3 System Services 13

3 Big Data Elements 15

3.1 Data Elements 15

3.2 Dataset at Rest 16

3.3 Dataset in Motion 17

3.4 Data Science Flavors 17

3.5 Big Data Analytics 19

3.6 Big Data Metrics 19

3.7 Big Data Security and Protection 19

4 Big Data Templates 19

4.1 Data Processes 20

4.2 Data Process Ordering Changes 20

4.3 Data Analytics Templates 20

# Executive Summary

The technologies for processing data and performing analytics have undergone a revolution within the last few years. New tools and approaches have emerged that have changed the way we orchestrate and resource our data processing systems. Any new technology emerges with a lot of hype, and it can take some time before a clearer picture emerges of what is new and different. This report seeks to clarify the underlying concepts of Big Data and Data Science to enhance communication among producers and consumers of these new technologies by helping us use the same term for the same concept.

# Introduction

## Objectives

The Definitions and Taxonomy subgroup focused on identifying the concepts involved in big data, and defining terms in both the concepts needed to describe this new paradigm, and to define the terms used in the reference architecture.

For *managers* the terms will distinguish the concepts needed to understand this changing field

For *procurement officers* this will provide the framework for discussing organizational needs, and distinguishing among offered approaches

For *marketers* this document will provide the means to promote the characteristics of solutions and innovations

For the *technical* community it will provide a common language to better differentiate the specific offerings

## How This Report Was Produced

“Big Data” and “Data Science” are being used as buzzwords that are composites of many concepts. To better identify those terms, we first addressed the individual concepts needed in this disruptive field. Then we came back to clarify the two over-arching buzzwords to provide clarity on what concepts they encompass.

To keep the topic of data and data systems manageable, we tried to restrict our discussions to what is different now that we have “big data”. We did not want to delve into expansive topics such as data-type or analytics taxonomies. We did, however, include the concepts involved in other methodologies that are needed in order to understand the new big data methodologies.

We further tried to keep all terms independent of a specific tool or implementation to not highlight only specific examples, and to stay general enough for the inevitable changes in the field.

We are aware that there is specific language is fields such as legal, that have implications for certain terms. While we are mindful of this, we were limited in creating this document to using the breadth of knowledge of the subgroup members. We will have to request input from the broader community during the comment period to address any specific domain conflicts the terminology used in this report.

## Structure of This Report

This document seeks to clarify the meanings of the broad terms big data and data science. So the reader can immediately go to this, they are presented first in section 2. The more elemental concepts and terms that shed additional insights are discussed later in section 3. In section 4 we begin to explore some more detailed concepts that will be important as we move into future phases of this project. In this first version, we describe some of the concepts that will be important to begin to determine categories or functional capabilities that represent architecture choices. By understanding the underlying communication and storage patterns, we will begin to provide more clarity on the strengths and weaknesses of different approaches.

Tightly coupled information can be found in a number of additional publications. The NIST Big Data Working Group Taxonomydescription of the more detailed components of described in the NIST Big Data Working Group Reference Architecture.concepts related to security and privacyDescriptions of where big data is going and how to get started to make use of these technologies, the reader is referred to the NIST Big Data Roadmap. to understand how these systems are architected to meet users’ needs, the reader is referred to the NIST Big Data Use Cases and Requirements document.

# Big Data Definitions



## Big Data Definition

Big data is used as a concept that refers to the inability of traditional data architectures to efficiently handle the new data sets. Characteristics that force a new architecture to achieve efficiencies are the dataset-at-rest characteristics ***volume***, and ***variety*** of data from multiple domains or types; and from the data-in-motion characteristics of ***velocity***, or rate of flow, and ***variability***, as the change in velocity. Each of these characteristics result in different architectures or different data lifecycle process orderings to achieve needed efficiencies. A number of other terms (particularly anything that can be expressed using a term starting with the letter ‘V”) are also used, but a number of these refer to the analytics, not to new big data architectures.

The new big data paradigm occurs when the scale of the data at rest or in motion forces the management of the data to be a significant driver in the design of the system architecture. Fundamentally the big data paradigm represents a shift in data system architectures from monolithic systems with vertical scaling (faster processors or disks) into a horizontally scaled system that integrates a loosely coupled set of resources. This shift occurred 20-some years ago in the simulation community when the scientific simulations began using massively parallel processing (MPP) systems. In different combinations of splitting the code and data across independent processors, computational scientists were able to greatly extend their simulation capabilities. This of course introduced a number of complications in such areas as message passing, data movement, latency in the consistency across resources, load balancing, and system inefficiencies while waiting on other resources to complete their tasks. In the same way, the big data paradigm represents this same shift, again using different mechanisms to distribute code and data across loosely-coupled resources to provide the scaling in data handling that is needed to match the scaling in the data.

The **Big Data Paradigm** consists of the distribution of data systems across horizontally-coupled independent resources to achieve the scalability needed for the efficient processing of extensive datasets.

While we certainly expect a continued evolution in the methods to achieve efficient scalability across resources, this paradigm shift (in analogy to the prior shift in the simulation community) is a one-time occurrence; at least until a new paradigm shift occurs beyond this “crowdsourcing” of a processing or data system across multiple horizontally-coupled resources.

***Big Data*** *consists of extensive datasets, primarily in the characteristics of volume, velocity and/or variety, that require a scalable architecture for efficient storage, manipulation, and analysis.*

A difficult question is what makes “Big Data” big, or how large does a dataset have to be in order to be called big data? The answer is an unsatisfying “it depends”. Data is considered “big” if the use of the new scalable architectures provides a business efficiency over other relational data model, in other words the functionality cannot be achieved in a traditional relational database platform.

Big data essentially focuses on the self-referencing viewpoint that data is big because it requires scalable systems to handle it, and architectures with better scaling have come about because of the need to handle big data.

**Big Data Engineering** is the storage and data manipulation technologies that leverage a collection of horizontally coupled resources to achieve a nearly linear scalability in performance.

New engineering techniques in the data layer have been driven by the growing prominence of data types that cannot be handled efficiently in a traditional relational model. The need for scalable access in structured data has led to software built on the name-value pair or big table paradigms. The rise of the importance of document analysis has spawned a document-oriented database paradigm, and the increasing importance of relationship data requirements have led to efficiencies in the use of graph-oriented data storage.

The new non-relational model database paradigms are typically referred to as NoSQL systems, alternately defined as “no SQL” or “not only SQL” (see the concept discussions in section 3). The difficulty in the identification of big data storage paradigms as NoSQL is first that it describes data persistence paradigms with respect to a query language, and second that there is a growing capability in the application of the SQL query language against the new data repository paradigms. While this term will continue to refer to the new data models beyond the relational model, the term itself will hopefully be replaced with a more suitable term, since it is misplaced to name a set of new storage paradigms with respect to a query language that is now being used.

**NoSQL** or **Big Data Models** refers to non-relational logical data models for the storage and manipulation of data across horizontally scaled resources; including techniques for example categorized at a high level as name-value, big table, document or graphical.

The Big Data paradigm has other implications from the technical innovations. The changes are not only in the logical data storage paradigm, but in the parallel distribution of data and code in the physical file system and direct queries against this storage.

The paradigm shift causes changes in the traditional data lifecycle. One description of the end-to-end data lifecycle categorizes the steps as collection, preparation, analysis and action. Different big data use cases can be characterized in terms of the dataset characteristics at-rest or in-motion, and in terms of the time window for the end-to-end data lifecycle. Dataset characteristics change the data lifecycle processes in different ways, for example in the point in the lifecycle at which the data is placed in persistent storage. In a traditional relational model, the data is stored after preparation (for example after the extract-transform-load and cleansing processes). In a high velocity use case, the data is prepared and analyzed for alerting, and only then is the data (or aggregates of the data) given a persistent storage. In a volume use case the data is often stored in the raw state in which it was produced, prior to the application of the preparation processes to cleanse and organize the data. The consequence of persistence of data in its raw state is that a schema or model for the data is only applied when the data is retrieved, known as schema on read.

**Schema-on-read** is the recognition that big data is often stored in a raw form based on its production, with the schema needed for organizing (and often cleansing) the data for analytics being applied as the data is queried from the repository.

A third consequence of big data engineering is often referred to as “*moving the processing to the data, not the data to the processing*”. The implication is that data is too extensive to be queried and moved into another resource for analysis, so the analysis program is instead distributed to the data-holding resources; with only the results being aggregated on a remote resource.

A number of additional uses of the buzzword “Big Data” actually refer to changes in analytics as a consequence of the extensiveness of the datasets, which will be discussed in the next section on Data Science.

At its heart, Big Data refers to the extension of data repositories and processing across horizontally-scaled resources, much in the same way the compute-intensive simulation community embraced massively parallel processing two decades ago. By working out methods for communication among resources, the same scaling is now available to data-intensive applications.

## Data Science Definition

In its purest form, data science is the “Fourth Paradigm” of science; following theory, experiment, and simulation. The fourth paradigm is a term coined by the Late Jim Gray to refer to the conduct of data analysis as an empirical science, learning directly from data itself . Data science as a paradigm would refer to the formulation of a hypothesis, the collection of the data (new or pre-existing) to address the hypothesis, and the analytical confirmation or denial of the hypothesis (or the determination that additional information or study is needed). As in any experimental science, the end result could in fact be that the original hypothesis itself needs to be reformulated. The key concept is that data science is an empirical science, performing the scientific process directly on the data. Note that the hypothesis may be driven by a business need, or can be the restatement of a business need in terms of a technical hypothesis.

***Data Science*** *is extraction of actionable knowledge directly from data through a process of discovery, hypothesis, and hypothesis testing.*

Data Science incorporates principles, techniques and methods from many disciplines and domains including mathematics, computer science (and more specifically machine learning and pattern recognition), statistics, operation research, data systems engineering and visualization. Data Scientists or data science teams solve complex data problems by employing deep expertise in one or more of these disciplines, as well as business strategy and domain knowledge. Personal skills in communication, presentation and inquisitiveness are also very important.

*A* ***Data Scientist*** *is a practitioner who has sufficient knowledge in the overlapping regimes of expertise in business needs, domain knowledge, analytical skills, and programming and systems engineering expertise to manage the end-to-end scientific method process through each stage in the big data lifecycle.*

While this full collection of skills can be present in a single individual, it is also possible that these skills are covered in the members of a team as shown in Figure 1.

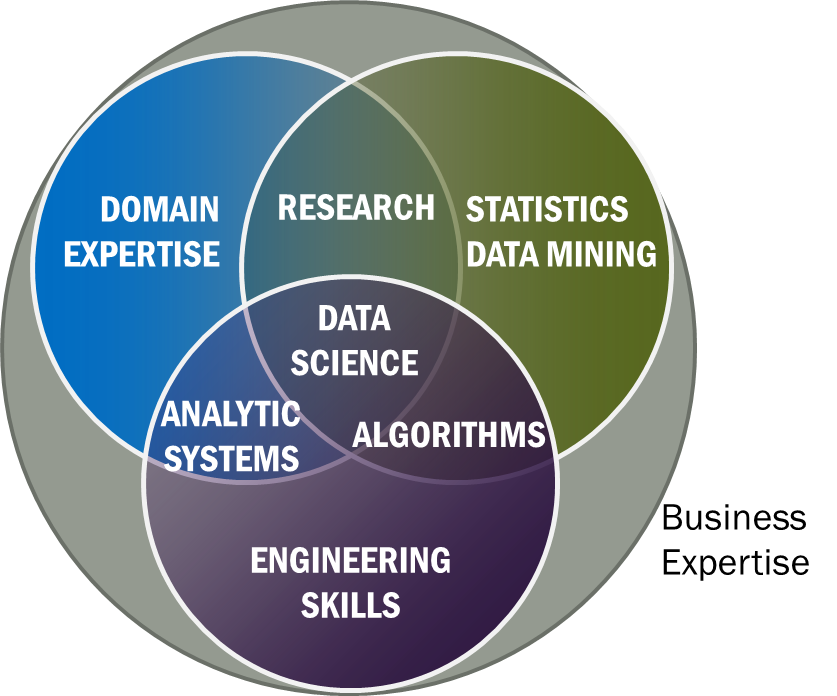


Figure 1: Skills needed in Data Science

As the characteristics of the data and the number of resources continue to scale, then the analysis also begins to change. Data Science is not solely concerned with analytics, but with the end-to-end experimental lifecycle. The implication is that the data scientist must be aware of the sources and provenance of the data, the appropriateness and accuracy of the transforms on the data, the interplay between the transformation algorithms and processes and the data storage mechanisms, etc.. This end-to-end overview role is to ensure that everything is being done right to meaningfully address the hypothesis.

Data Science has also been used as a buzzword to mean additional concepts to those given above.

In big data systems it is often sufficient to identify a **correlation** in order for the business to take action. As a trivial example if it can be determined that using the color blue on a website leads to greater sales over the use of green, then this correlation information can be used to improve the business. The reason for the preference is not needed, it is enough to determine correlation and not causation.

A hotly debated data science concept is also in the use of data **sampling**. A bit of a digression into history is useful here to set the stage. Statistics has a branch of study on the computer design of experiments, determining the necessary and sufficient data that is needed to rigorously determine and outcome, for example in a pharmaceutical clinical trial. When the data mining community began, the emphasis was typically on re-purposed data, meaning you did some data sampling for the data needed to train your models, but the data was collected for some other purpose. The sampling requirements ensured that the analytics were not prone to “over-fitting” (meaning the pattern matched the data sample chosen for training, but did not match well any other sampled data). In the new Big Data Paradigm, it is implied that you no longer have to sample data. This is not true, in that even if you use all the data available it may not address questions of the true “population” of interest; only of those that had behaviors that led them to produce the data. For example, studying twitter data to analyze people’s behaviors doesn’t let you address all people as not everyone uses twitter. Even if you analyze all the data twitter has, it still presents a selection bias across all people for those that use twitter in the first place and thus cannot be used to comment on the full real-world population.

One data science debate is the assertion that **more data beats better algorithms**. The heart of this debate says that a few bad data elements are less likely to influence the analytical results in a large dataset than if errors are present in a small sample of that dataset. If the analytics needs are correlation and not causation, then this assertion is easier to justify. Outside this context, the assertion for a deterministic or causal analysis is not as clear and is debated in each specific circumstance.

Finally, Data Science is also given a number of characteristics, including *veracity* (accuracy of the data) and *value* (to the organization of the results of the analytics). These characteristics and others including all quality control, metadata and data provenance, etc. are already present in any data analytics and are not new to big data.

For descriptive purposes, analytics activities can be broken into different stages including discovery, exploratory analysis, correlation analysis, predictive modeling, machine learning, etc. Again these analytics categories are not specific to big data, but some have gained more visibility due to their greater application in big data analytics.

Data Science is tightly linked to Big Data, and refers to the management and execution of the end-to-end data processes, including the behaviors of the data system as well. As such, Data Science includes all of analytics, but analytics does include all of data science.

## Big Data Taxonomy

The NIST Big Data Working Group Reference Architecture document Provides more detailed information concerning the high level functional architecture in big data systems. In addition, the NIST Big Data Working group has a separate taxonomy document to provide more detailed information on the component elements in the Reference Architecture, and we refer the reader to this document for more in-depth exploration of big data technologies.



# Big Data Elements

The rate of growth of amounts of data generated and stored has been increasing exponentially. 90% of all current data was likely created in the past two years. This data explosion is creating opportunities for new ways of combining and using data to find value. One of the significant shifts is in the amount of unstructured data. Structured data has typically been the focus of most enterprise, and has been handled through use of the relational model. Micro-texts, relationship data, images and videos have seen such an explosion, that the push is to incorporate this data to generate value. The benefit gained from the ability to process large amounts of information, is the main attraction of big data analytics.



## Data Elements

A through description of data elements themselves is beyond the scope of this work. Most of the data in business systems was typically ***structured*** data that could be described efficiently in a ***relational model***. Unstructured data types such as text, image, video, and relationship data have been increasing in both volume and prominence. The need to analyze ***unstructured*** or ***semi-structured*** data has been present for many years, so while important to a discussion of storage and analytics, we will not discuss a data type taxonomy here as it is outside our scope - since it has not changed in the Big Data Paradigm shift.

An additional concept that is again not new in the paradigm shift is the presence of ***complexity*** in the data elements. There are systems where data elements cannot be addressed independently. This is evident in analytics for the human genome, it is the relationship between the elements, their position and proximity to other elements that matters. The term *complexity* refers to this inter-relationship between data elements or data records.

We also note the concept of ***metadata***, or data about data. As we move into an era of open data and linked data, it becomes ever more important to have information about how data was collected, transmitted and processed; to ensure that when it is repurposed from the process from which it was originally collected that it will be used correctly.

A companion concept that is outside the scope of big data is none-the-less worth noting here. ***Semantic data*** essentially refers to the definition description of a data element to ensure it is properly interpreted. There are a number of mechanisms for implementing these unique definitional descriptions which our outside our scope. There are also ***taxonomies*** or ***ontologies*** that represent information about relationships between elements. This concept contributes to the discussion in the next section of the variety of data at rest.

Big Data comes in two distinct forms, at rest and in motion. Both kinds of Big Data typically include both structured and unstructured data. The distinction, as the names imply, is in how the data is handled. Data "at rest" is not permanently at rest – sooner or later it will be used for analysis, visualization and interpretation. This data is typically used in a batch process planned for later time, i.e. the end of the day, month, quarter, etc. Data in motion concerns the data as it is in transit from one position to another. This distinction enables us to clarify some of the characteristics of big data.

## Dataset at Rest

Data at rest is a term that is sometimes used to refer to all [data](http://searchdatamanagement.techtarget.com/definition/data) in computer [storage](http://searchstorage.techtarget.com/definition/storage) while excluding data that is traversing a network or temporarily residing in computer memory to be read or updated. Data at rest can be archival or reference files that are changed rarely or never; data at rest can also be data that is subject to regular but not constant change. Examples include vital corporate files stored on the hard drive of an employee's notebook computer, files on an external backup medium, files on the servers of a storage area network ([SAN](http://searchstorage.techtarget.com/definition/storage-area-network-SAN)), or files on the servers of an offsite backup service provider. While there are monolithic systems that competently process data at rest, there are a number of options to process it by spreading it across a number of less expensive systems. Typical characteristics of data at rest that are significantly different in the era of Big data are the *Volume* and *Variety*.

***Volume*** is the characteristic of data at rest that is most associated with big data. 90% of all data ever [created](http://www-01.ibm.com/software/data/bigdata/), was created in the past 2 years. Estimates show that the [amount](http://www.emc.com/leadership/programs/digital-universe.htm)s of data in the world is doubling every two years. Should this trend continue, by 2020, there will be 50 times the amount of data as there had been in 2011. The sheer volume of the data is colossal - the era of a trillion sensors is upon us. This *volume* presents the most immediate challenge to conventional information technology structures. It has stimulated new ways for scalable storage across a collection of horizontally coupled resources, and a distributed approach to querying, described in Section 2.1 as Big Data Engineering.

Briefly, the traditional relational model has been relaxed for the persistence of newly prominent data types. These logical data models, typically lumped together as ***NoSQL***, can currently be classified at ***Big Table***, ***Name-Value***, ***Document*** and ***Graphical*** models. A discussion of these logical models was not part of the phase one activities that led to this document.

The second characteristic of data at rest is the increasing need to use a ***Variety*** of data, meaning the data represents a number of data domains and a number of data types. Traditionally, a variety of data was handled through transforms or pre-analytics to extract features that would allow integration with other data through a *relational model*. Given the wider range of data formats, structures, timescales and semantics that are desirous to use in analytics, the integration of this data becomes more complex. This challenge arises as data to be integrated could be text from social networks, image data, or a raw feed directly from a sensor source. Big Data Engineering has spawned data storage models that are more efficient for *unstructured* data types than a relational model, causing a derivative issue for the mechanisms to integrate this data. It is possible that the data to be integrated for analytics may be of such volume that it cannot be moved in order to integrate, or it may be that some of the data is not under control of the organization creating the data system. In either case, the variety of big data forces a range of new big data engineering in order to efficiently and automatically integrate data that is stored across multiple repositories and in multiple formats.

There are additional aspects of big data at rest that will not be fully explored in this first phase of the NIST Big Data Working Group process, including the range of persistence mechanisms (flatfiles, RDB, markup, NoSQL models), and the mechanisms for providing the communication among the horizontally coupled resources holding the data in the NoSQL or Big Data logical models. This discussion relates to the ***relaxation of the principles of a relational model***. While very important, this was considered out-of-scope for the phase one activities as it needs much more thought and discussion. Likewise any discussion of the use of ***multiple tiers of storage*** (in-memory, cache, solid state drive, hard drive, network drive, etc.) for storage efficiency is likewise deferred for a later discussion.

## Dataset in Motion

Big Data "in motion" is processed and analyzed on the fly – in real time, or nearly so, therefore it has to be handled in a very different way than traditional stored data. Big Data in motion tends to resemble event processing architectures, and focuses on real-time or operational intelligence applications analyzing and monitoring the current data activity and changes, adding considerable challenge to managing the data both safely and effectively.

The ***Velocity*** is the speed at which the data is created, stored, analyzed and visualized. In the big data era, data is created in real-time or near real-time. With the availability of Internet connected devices, wireless or wired, machines and devices can pass-on their data the moment it is created. Data Flow rates are increasing with enormous speeds and variability, creating new challenges to enable real or near real time data usage. Traditionally this concept has been described as ***streaming data***. As such there are aspects of this that are not new, as companies such as those in telecommunication have been sifting through high volume and velocity data for years. The new horizontal scaling approaches do however add new big data engineering options for efficiently handling this data.

The second concept for data in motion is variability. ***Variability*** can refer to a change in the rate of flow of data. Given many data processes generate a surge in the amount of data arriving in a given amount of time, new techniques are needed for efficiently handling this data. This need is often tied up with the automatic provisioning of additional virtualized resources in a cloud environment. The techniques used here are again outside our scope, and more appropriate to a discussion of operational cloud architectures.

## Data Science

As discussed in Section 2.2, Data Science refers to a shift in analytics to an empirical mindset, considering the data system as the “equipment”. As such Data Science is a superset of analytics, which typically refers to the focus on the processes for the conversion of organized information into actionable knowledge. Data Science is concerned with the end-to-end data lifecycle to achieve actionable results for the enterprise. We can distinguish a number of analytic approaches for the empirical analysis of data to facilitate better communication.

***Blue Sky Data Science*** is a term used to indicate an open-ended discovery process. Hard lessons learned from the Data Mining community suggests that this type of activity be approached with caution. It is far better to follow a rapid hypothesis-exploratory confirmation cycle than to assume you can merely browse data and find actionable insights. It is more efficient to first hypothesize and then browse to get an initial sense if the hypothesis is worth pursuing.

**Basic Data Science** would refer to the more traditional analytics approach, where there is a specific goal in mind. This would more nearly align with traditional statistics and data mining methods.

**Applied Data Science** would then refer to the encoding of the end-to-end data transformations and analytics into a repeatable, operational system.

## Big Data Analytics

The new technologies provide changes to the characteristics of the analytics that are possible, but are not directly ties to completely new types of analytics never done before. However, given the retrieval speeds, analysts are able to interact with their data in ways that were not previously possible. In addition, greater emphasis is being placed on the value of correlation. Most traditional analytics has focused on causation, being able to describe why something is happening. There are cases where it is enough to know the direction of a trend to take action.

Some techniques are being applied differently, to downsize or summarize before you can take information from big data systems and work with it in an analytics application

While the analytics that are being done have otherwise not changed, the analytics tools have to adapt to run against the distributed data repositories. One of the mechanisms for this is a divide and conquer algorithm known as MapReduce.

MapReduce is a method of splitting a query and subsequent analytics task into code that runs on individual data nodes, with a corresponding method of combining the results from each node into a final result of the query and analytics.

While a number of “V”s have been proposed to relate to analytics in the big data realm, primarily these attributes are not different in kind from what they have always been.

***Veracity*** refers to the completeness and accuracy of the data. This relates to the “garbage-in garbage-out” issue that has been with us for a long time. If the analytics are causal, then the quality of every data element is critically important. If the analytics are correlations or trending over massive volume datasets, then individual bad elements will be lost in the overall counts and the trend will still be accurate. In between there are many conversations whether there is a point where more data beats better algorithms.

The ***provenance*** or history of the data is becoming more prominent in big data analytics. As more data is being re-purposed for new types of analytics. As the usage of data persists far beyond the control of the data producers, it becomes ever more critical that metadata about the full creation and processing history is made available along with the data.

## Big Data Metrics

One of our most fundamental questions was actually the first one asked…‘how big does it have to be in order to be Big Data’? The unsatisfactory response is that it depends. The simplistic answer is whenever the data system begins to span horizontally scaled systems, then you have moved into needing big data technology. Another way this is often expressed is that you have a big data problem when the size of the data is itself a significant part of the problem.

In any case this is something that needs more careful thought and delineation, and will be a focus in future phases of the working group.

## Big Data Security and Protection

The security and privacy components and concerns are discussed in the NIST Big Data Working Group Security and Privacy report.

# Big Data Patterns

To provide definitions of the differences in big data technologies, we want to describe different “templates” or “patterns” that describe methods related to the data characteristics described in section 3. These templates describe the technical use cases for the processes in the big data architecture that can be used to categorize and therefore better understand the different use cases presented in the NIST Big Data Use Cases and Reference Architecture document. While these patterns will be explored more fully in later stages of the working group’s discussions, we introduce some of the fundamental concepts here.

## Data Processes

From the data’s perspective, it goes through a number of processes during each of the four stages of a Data Lifecycle

* Collection: results in “raw” data, or data in its original form
* Preparation: is the collection of processes that take raw data and turn it into cleansed, organized information
* Analysis: the techniques that take organized information and produce synthesized knowledge
* Action: the processes that take the synthesized or created knowledge and put them to use in the generation of value for the enterprise.

## Data Process Ordering Changes

In the traditional data warehouse, you collected, prepared, and then stored the data. The data was stored in a way that optimized the intended analytics. Given the different big data characteristics, the ordering of the processes change, in particular with respect to the step after which the data is stored in a persistent repository..

* **Data Warehouse**: Curation=ETL, with storage coming after curation
* **Volume** **application**: the data is stored immediately, before curation. In this case curation occurs on read, and is referred to as “schema on read’
* **Velocity** **application**: do collection and curation and analytics (alerting) on the fly, and possibly some summarization or aggregation prior to storage

Just as simulations split the analytical processing across clusters of processors, here data processes are redesigned to splitting data transformations. Because the data may be too big to move, the transformation code may be sent across the data persistence resources, rather that the data be extracted and brought to the transformation servers

## Transactions

Relational databases have traditionally supported the ACID transaction model. Big Data has introduced BASE transactions. Distributed Big Data is subject to Brewer’s CAP theorem.

### ACID Transactions

Relational and SQL databases typically support online transaction processing using ACID transactions. The ACID acronym stands for:

* Atomic – All of the work in a transaction completes (commit) or none of it completes (rollback)
* Consistent – A transaction transforms the database from one consistent state to another consistent state. Consistency is defined in terms of database constraints.
* Isolated – The results of any changes made during a transaction are not visible until the transaction has committed.
* Durable – The results of a committed transaction survive failures.

The SQL Standard defines four transaction isolation levels in terms of three phenomena that could occur between two concurrent transactions:

* Dirty Reads – T1 reads data modified by T2 but not yet committed
* Unrepeatable reads – T1 rereads data and see effects of dataT2 has modified or deleted AND committed
* Phantom reads – T1 rereads data and sees data T2 has inserted AND committed.

The following chart shows for each theoretical isolation level which phenomena are possible:

|  |  |  |  |
| --- | --- | --- | --- |
| **Isolation Level** | **Dirty Read** | **Unrepeatable Read** | **Phantom Read** |
| Read uncommitted | Yes | Yes | Yes |
| Read committed | No | Yes | Yes |
| Repeatable read | No | No | Yes |
| Serializable | No | No | No |

SQL implementations support transactions and isolation levels using a variety of mechanisms. These mechanisms typically require some amount of overhead. This overhead is often viewed as an impediment to highly scalable databases.

### BASE Transactions

The BASE Acronym is often used to describe the types of transactions typically supported by NoSQL databases. BASE is specifically contrived to be the opposite of ACID:

* Basically Available,
* Soft state,
* Eventually Consistent

While ACID transactions must be consistent at the end of the transaction, BASE transactions allow a database to be in a temporarily inconsistent state that will eventually be resolved.

BASE transactions are the other end of a continuum from ACID transactions where the continuum is in part described by Brewer’s CAP theorem.

### Brewer’s CAP Theorem

Brewer’s CAP Theorem (or conjecture) says that a distributed system can support only two of the following three characteristics:

* Consistency
* Availability
* Partition tolerance

The slides from Brewer’s July 2000 talk do not define these characteristics. Additional definition can be found in Wikipedia articles and “BASE: An Acid Alternative”, Dan Pritchett, ACMQueue, July 28, 2008:

* **Consistency**
  + all nodes see the same data at the same time – Wikipedia
  + client perceives that a set of operations has occurred all at once – Pritchett

This use of “consistency” is similar to the Atomic property in ACID transactions.

* **Availability**
  + Node failures do not prevent survivors from continuing to operate – Wikipedia
  + Every operation must terminate in an intended response – Pritchett
* **Partition Tolerance**
  + The system continues to operate despite arbitrary message loss – Wikipedia
  + Operations will complete, even if individual components are unavailable – Pritchett

Database implementations that support consistency and availability will not be able to tolerate partitioning. Database implementations that support partitioning (sharding) across multiple nodes in a network will be able to support either consistency or availability, but not both.

### Read versus Write Transactions

In many database applications, the ACID transaction characteristics are critical for transactions that write (or modify) one or more rows in one or more tables. For some applications the ACID transaction properties are also critical for transactions that only read data. For these applications, it is critical for the data to effectively remain unchanged across the life of the transactions.

However, many applications can tolerate changes to the underlying data being read during a transaction. A statistical analysis of a billion rows/documents/etc. is unlikely to be significantly affected by the addition or modification of a small percentage of the underlying data. For these applications, it may be reasonable to operate in a read uncommitted mode.

## Data Analytics Time Window

One area that needs more differentiation is the affects of the analytics or data lifecycle time window on the possible architectures. The big data characteristic velocity was discussed earlier, and refers to the rate at which information is flowing into and through the system. An additional time span for consideration is in the speed of interaction between the analytics processes and the person or process responsible for delivering the actionable insight. While the three broadest categories of batch (or offline) processing, online processing, and interactive processing are not new, they are a large factor in the choice of architecture and component tools to be used. Given the greater query and analytic speeds within big data, due to the horizontal scaling, there is an increasing emphasis on the interactive category. Rapid analytics cycles allow an analyst to do exploratory discovery on the data, browsing more of the data space than might otherwise have been examined for the task at hand.

## Storage Medium Changes

Another area to be explored in the patterning approach to explaining different methods and methodologies is the use of a broad range of storage capacities. Expanding the choices for big data include the use of the spectrum from in-memory techniques, to caching techniques, to local disk, to remote disk, to archival storage. As mentioned in Section 3.2, this is an area that needs to be described into the patterns that are used and the benefits of each.

# Conclusion

In this document, we have explored the fundamental concepts needed to understand the new paradigm for data applications, collectively known as Big Data, and the analytic processes collectively known as Data Science. In future working group discussions, greater detail will be given to many of the features of big data systems to describe the patterns that are followed to achieve greater performance and capaiblities.