The industry has two more definitions for big data.

- Big data is a collection of data sets so large and complex that it becomes difficult to process using on-hand database management tools or traditional data processing applications.
- Big data is the data whose scale, diversity, and complexity require new architecture, techniques, algorithms, and analytics to manage it and extract value and hidden knowledge from it.

Some of the major challenges that big data analytics program are facing today include the following:

1. **Uncertainty of Data Management Landscape:** Because big data is continuously expanding, there are new companies and technologies that are being developed every day. A big challenge for companies is to find out which technology works bests for them without the introduction of new risks and problems.
2. **The Big Data Talent Gap:** While Big Data is a growing field, there are very few experts available in this field. This is because Big data is a complex field and people who understand the complexity and intricate nature of this field are far few and between. Another major challenge in the field is the talent gap that exists in the industry
3. **Getting data into the big data platform:** Data is increasing every single day. This means that companies have to tackle limitless amount of data on a regular basis. The scale and variety of data that is available today can overwhelm any data practitioner and that is why it is important to make data accessibility simple and convenient for brand managers and owners.
4. **Need for synchronization across data sources:** As data sets become more diverse, there is a need to incorporate them into an analytical platform. If this is ignored, it can create gaps and lead to wrong insights and messages.
5. **Getting important insights through the use of Big data analytics:** It is important that companies gain proper insights from big data analytics and it is important that the correct department has access to this information. A major challenge in the big data analytics is bridging this gap in an effective fashion.
6. **Privacy and Security**
7. **Data access and Sharing of information**
8. **Challenges in Analytics**
9. **Technical Challenges**
    1. **Fault Tolerance**
    2. **Scalability**
    3. **Quality of Data**
    4. **Heterogeneous data**

Traditional Data Systems

Traditional data systems, such as relational databases and data warehouses, have been the primary way businesses and organizations have stored and analyzed their data for the past 30 to 40 years. Although other data stores and technologies exist, the major percentage of business data can be found in these traditional systems. Traditional systems are designed from the ground up to work with data that has primarily been structured data. Characteristics of structured data include the following:

- Clearly defined fields organized in records. Records are usually stored in tables. Fields have names, and relationships are defined between different fields.
- Schema-on-write that requires data be validated against a schema before it can be written to disk. A significant amount of requirements analysis, design, and effort up front can be involved in putting the data in clearly defined structured formats. This can increase the time before business value can be realized from the data.
- A design to get data from the disk and load the data into memory to be processed by applications. This is an extremely inefficient architecture when processing large volumes of data this way. The data is extremely large and the programs are small. The big component must move to the small component for processing.
- The use of Structured Query Language (SQL) for managing and accessing the data.
- Relational and warehouse database systems that often read data in 8k or 16k block sizes. These block sizes load data into memory, and then the data are processed by applications. When processing large volumes of data, reading the data in these block sizes is extremely inefficient.
- Organizations today contain large volumes of information that is not actionable or being leveraged for the information it contains.
- An order management system is designed to take orders. A web application is designed for operational efficiency. A customer system is designed to manage information on customers. Data from these systems usually reside in separate data silos. However, bringing this information together and correlating with other data can help establish detailed patterns on customers.
- In a number of traditional siloed environments data scientists can spend 80% of their time looking for the right data and 20% of the time doing analytics. A data-driven environment must have data scientists spending a lot more time doing analytics.

Every year organizations need to store more and more detailed information for longer periods of time. Increased regulation in areas such as health and finance are significantly increasing storage volumes. Expensive shared storage systems often store this data because of the critical nature of the information. Shared storage arrays provide features such as striping (for performance) and mirroring (for availability). Managing the volume and cost of this data growth within these traditional systems is usually a stress point for IT organizations. Examples of data often stored in structured form include Enterprise Resource Planning (ERP), Customer Resource Management (CRM), financial, retail, and customer information.

Atomicity, Consistency, Isolation, Durability (ACID) compliant systems and the strategy around them are still important for running the business. A number of these systems were built over the years and support business decisions that run an organization today. Relational databases and data warehouses can store petabytes (PB) of information. However, these systems were not

designed from the ground up to address a number of today's data challenges. The cost, required speed, and complexity of using these traditional systems to address these new data challenges would be extremely high.

Why Traditional Systems Have Difficulty with Big Data

The reason traditional systems have a problem with big data is that they were not designed for it.

- **Problem—Schema-On-Write:** Traditional systems are schema-on-write. Schema-on-write requires the data to be validated when it is written. This means that a lot of work must be done before new data sources can be analyzed. Here is an example: Suppose a company wants to start analyzing a new source of data from unstructured or semi-structured sources. A company will usually spend months (3–6 months) designing schemas and so on to store the data in a data warehouse. That is 3 to 6 months that the company cannot use the data to make business decisions. Then when the data warehouse design is completed 6 months later, often the data has changed again. If you look at data structures from social media, they change on a regular basis. The schema-on-write environment is too slow and rigid to deal with the dynamics of semi-structured and unstructured data environments that are changing over a period of time. The other problem with unstructured data is that traditional systems usually use Large Object Byte (LOB) types to handle unstructured data, which is often very inconvenient and difficult to work with.
- **Solution—Schema-On-Read:** Hadoop systems are schema-on-read, which means any data can be written to the storage system immediately. Data are not validated until they are read. This enables Hadoop systems to load any type of data and begin analyzing it quickly. Hadoop systems have extremely short business latency compared to traditional systems. Traditional systems require schema-on-write, which was designed more than 50 years ago. A lot of companies need real-time processing of data and customer models generated in hours or days versus weeks or months. The Internet of Things (IoT) is accelerating the data streams coming from different types of devices and physical objects, and digital personalization is accelerating the need to be able to make real-time decisions. Schema-on-read gives Hadoop a tremendous advantage over traditional systems in an area that matters most, that of being able to analyze the data faster to make business decisions. When working with complex data structures that are semi-structured or unstructured, schema-on-read enables data to be accessed much faster than schema-on-write systems.
- **Problem—Cost of Storage:** Traditional systems use shared storage. As organizations start to ingest larger volumes of data, shared storage is cost prohibitive.
- **Solution—Local Storage:** Hadoop can use the Hadoop Distributed File System (HDFS), a distributed file system that leverages local disks on commodity servers. Shared storage is about $1.20/GB, whereas local storage is about $.04/GB. Hadoop's HDFS creates three replicas by default for high availability. So at 12 cents per GB, it is still a fraction of the cost of traditional shared storage.
- **Problem—Cost of Proprietary Hardware:** Large proprietary hardware solutions can be cost prohibitive when deployed to process extremely large volumes of data. Organizations are spending millions of dollars in hardware and software licensing costs

while supporting large data environments. Organizations are often growing their hardware in million dollar increments to handle the increasing data. New technology in traditional vendor systems that can grow to petabyte scale and good performance are extremely expensive.

- **Solution—Commodity Hardware:** It is possible to build a high-performance super-computer environment using Hadoop. One customer was looking at a proprietary hardware vendor for a solution. The hardware vendor's solution was $1.2 million in hardware costs and $3 million in software licensing. The Hadoop solution for the same processing power was $400,000 for hardware, the software was free, and the support costs were included. Because data volumes would be constantly increasing, the proprietary solution would have grown in $500k and $1 million dollar increments, whereas the Hadoop solution would grow in $10,000 and $100,000 increments.
- **Problem—Complexity:** When you look at any traditional proprietary solution, it is full of extremely complex silos of system administrators, DBAs, application server teams, storage teams, and network teams. Often there is one DBA for every 40 to 50 database servers. Anyone running traditional systems knows that complex systems fail in complex ways.
- **Solution—Simplicity:** Because Hadoop uses commodity hardware and follows the "shared-nothing" architecture, it is a platform that one person can understand very easily. Numerous organizations running Hadoop have one administrator for every 1,000 data nodes. With commodity hardware, one person can understand the entire technology stack.
- **Problem—Causation:** Because data is so expensive to store in traditional systems, data is filtered and aggregated, and large volumes are thrown out because of the cost of storage. Minimizing the data to be analyzed reduces the accuracy and confidence of the results. Not only are accuracy and confidence to the resulting data affected, but it also limits an organization's ability to identify business opportunities. Atomic data can yield more insights into the data than aggregated data.
- **Solution—Correlation:** Because of the relatively low cost of storage of Hadoop, the detailed records are stored in Hadoop's storage system HDFS. Traditional data can then be analyzed with nontraditional data in Hadoop to find correlation points that can provide much higher accuracy of data analysis. We are moving to a world of correlation because the accuracy and confidence of the results are factors higher than traditional systems. Organizations are seeing big data as transformational. Companies building predictive models for their customers would spend weeks or months building new profiles. Now these same companies are building new profiles and models in a few days. One company would have a data load take 20 hours to complete, which is not ideal. They went to Hadoop and the time for the data load went from 20 hours to 3 hours.
- **Problem—Bringing Data to the Programs:** In relational databases and data warehouses, data are loaded from shared storage elsewhere in the datacenter. The data must go over wires and through switches that have bandwidth limitations before programs can process the data. For many types of analytics that process 10s, 100s, and 1000s of terabytes, the capability of the computational side to process data greatly exceeds the storage bandwidth available.
- **Solution—Bringing Programs to the Data:** With Hadoop, the programs are moved to where the data is. Hadoop data is spread across all the disks on the local servers that make up the Hadoop cluster, often in 64MB or 128MB block increments. Individual

programs, one for every block, runs in parallel (up to the number of available map slots, more on this later) across the cluster, delivering a very high level of parallelization and Input/Output Operations per Second (IOPS). This means Hadoop systems can process extremely large volumes of data much faster than traditional systems and at a fraction of the cost because of the architecture model. Moving the programs (small component) to the data (large component) is an architecture that supports the extremely fast processing of large volumes of data.

Successfully leveraging big data is transforming how organizations are analyzing data and making business decisions. The "value" of the results of big data has most companies racing to build Hadoop solutions to do data analysis. Often, customers bring in consulting firms and want to "out Hadoop" their competitors. Hadoop is not just a transformation technology; it has become the strategic difference between success and failure in today's modern analytics world.

Big Data	**Normal or Conventional Data**
Huge data sets.	Data set size in control.
Unstructured data such as text, video, and audio.	Normally structured data such as numbers and categories, but it can take other forms as well.
Hard-to-perform queries and analysis.	Relatively easy-to-perform queries and analysis.
Needs a new methodology for analysis.	Data analysis can be achieved by using conventional methods.
Need tools such as Hadoop, Hive, Hbase, Pig, Sqoop, and so on.	Tools such as SQL, SAS, R, and Excel alone may be sufficient.
Raw transactional data.	The aggregated or sampled or filtered data.
Used for reporting, basic analysis, and text mining. Advanced analytics is only in a starting stage in big data.	Used for reporting, advanced analysis, and predictive modeling.
Big data analysis needs both programming skills (such as Java) and analytical skills to perform analysis.	Analytical skills are sufficient for conventional data; advanced analysis tools don't require expert programming skills.
Petabytes/exabytes of data. Millions/billions of accounts. Billions/trillions of transactions.	Megabytes/gigabytes of data. Thousands/millions of accounts. Millions of transactions.
Generated by big financial institutions, Facebook, Google, Amazon, eBay, Walmart, and so on.	Generated by small enterprises and small banks.