Big Data Analytics (2CS702)

Lecture #1

Introduction to Big Data

Dr Jai Prakash Verma

Associate Professor

Department of Computer Science & Engineering

Institute of Technology, Nirma University, Ahmedabad

In today's discussion…

- Introduction to data
- Current trend
- Data and Big data
- Big data vs. small data
- Tools and techniques

Introduction to data

- Example:

  10, 25, …, Kharagpur, 10CS3002, namo@gov.in

  Anything else?

- Data vs. Information

  100.0, 0.0, 250.0, 150.0, 220.0, 300.0, 110.0

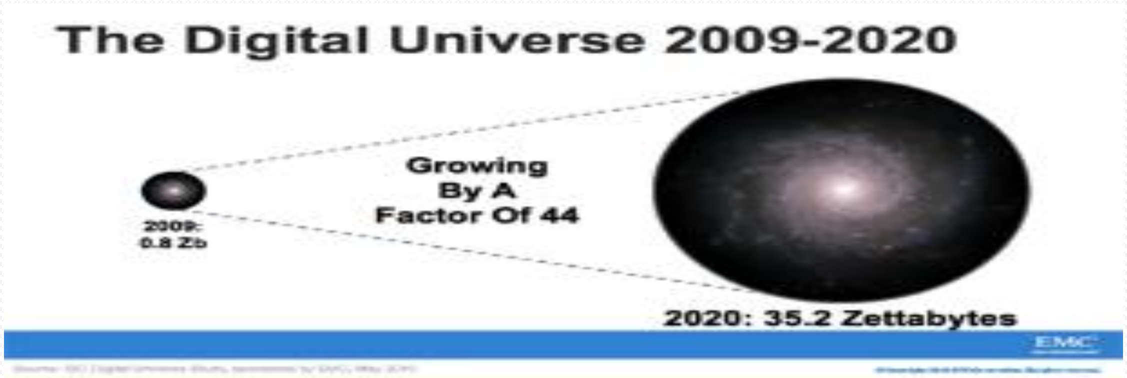
  Is there any information?

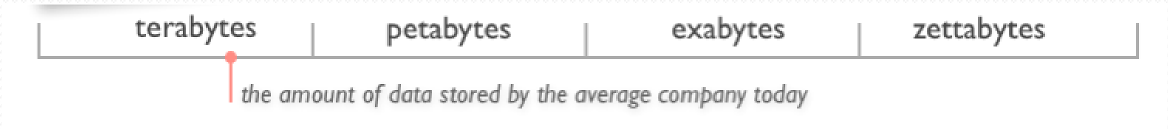
How large your data is?

- What is the maximum file size you have dealt so far?
  - Movies/files/streaming video that you have used?
- What is the maximum download speed you get?
  - To retrieve data stored in distant locations?
- How fast your computation is?
  - How much time to just transfer from you, process and get result?

Memory unit	Size	Binary size
kilobyte (kB/KB)	10^3	2^{10}
megabyte (MB)	10^6	2^{20}
gigabyte (GB)	10^9	2^{30}
terabyte (TB)	10^{12}	2^{40}
petabyte (PB)	10^{15}	2^{50}
exabyte (EB)	10^{18}	2^{60}
zettabyte (ZB)	10^{21}	2^{70}
yottabyte (YB)	10^{24}	2^{80}

Growth of data





Sources of data

- “Every day, we create 2.5 quintillion bytes of data
  - So much that 90% of the data in the world today has been created in the last two years alone.

  - The data come from several sources
    - sensors used to gather climate information
    - posts to social media sites,
    - digital pictures and videos
    - purchase transaction records
    - cell phone GPS signals

etc. ...... to name a few!

Examples

Social media and networks
(All of us are generating data)

Scientific instruments
(Collecting all sorts of data)

Mobile devices
(Tracking all objects all the time)

Sensor technology and networks
(Measuring all kinds of data)

Now data is Big data!

- No single standard definition!

- 'Big-data' is similar to 'Small-data', but bigger

  …but having data bigger consequently requires different approaches

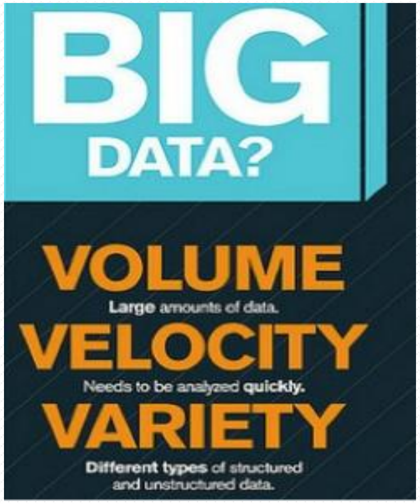
  - techniques, tools and architectures

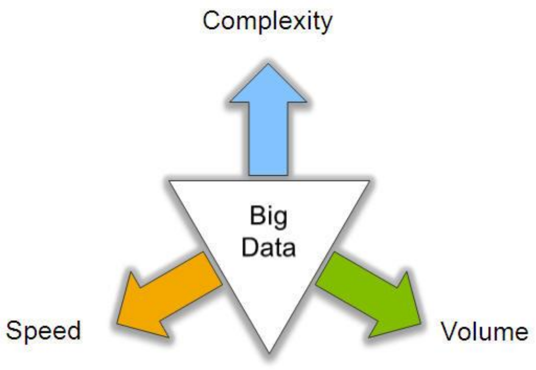
  …to solve: new problems

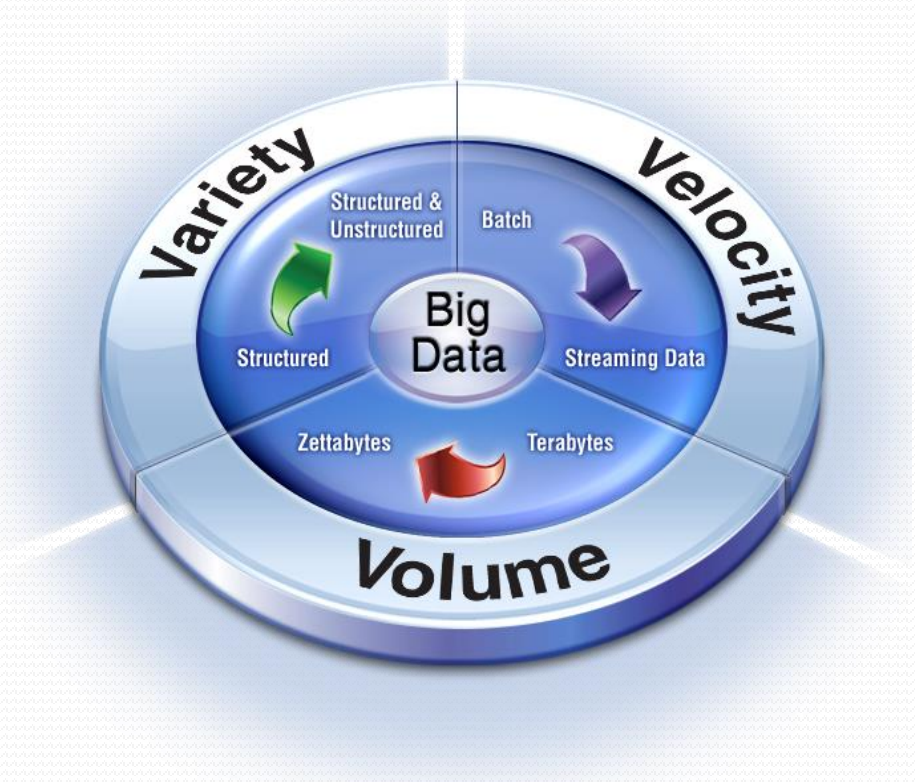
  …and, of course, in a better way

Big data is data whose scale, diversity, and complexity require new architecture, techniques, algorithms, and **analytics** to manage it and extract value and hidden knowledge from it…

Characteristics of Big data: **V3**

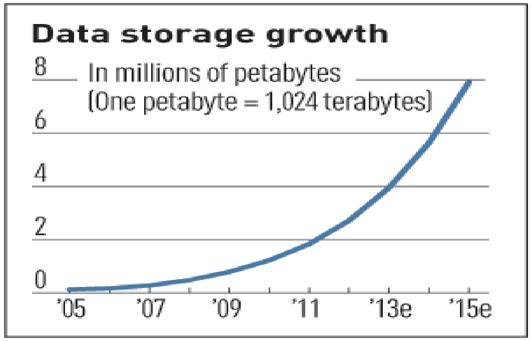


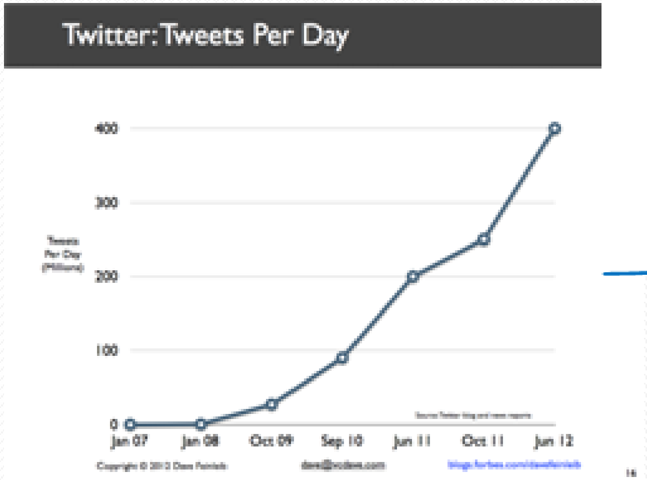




V3 : V for Volume

- Volume of data, which needs to be processed is increasing rapidly
  - More storage capacity
  - More computation
  - More tools and techniques



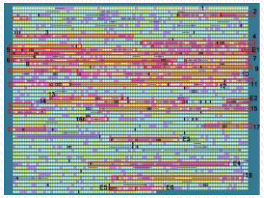


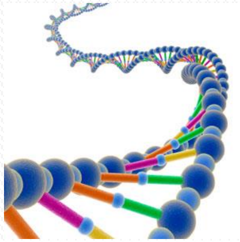
Exponential increase in collected/generated data

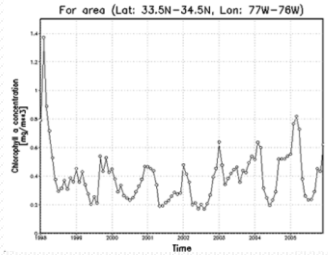
V3: V for Variety

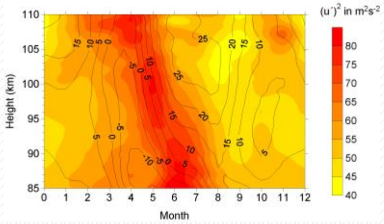
- Various formats, types, and structures
  - Text, numerical, images, audio, video, sequences, time series, social media data, multi-dimensional arrays, etc…
- Static data vs. streaming data
- A single application can be generating/collecting many types of data

To extract knowledge➔ all these types of data need to be linked together





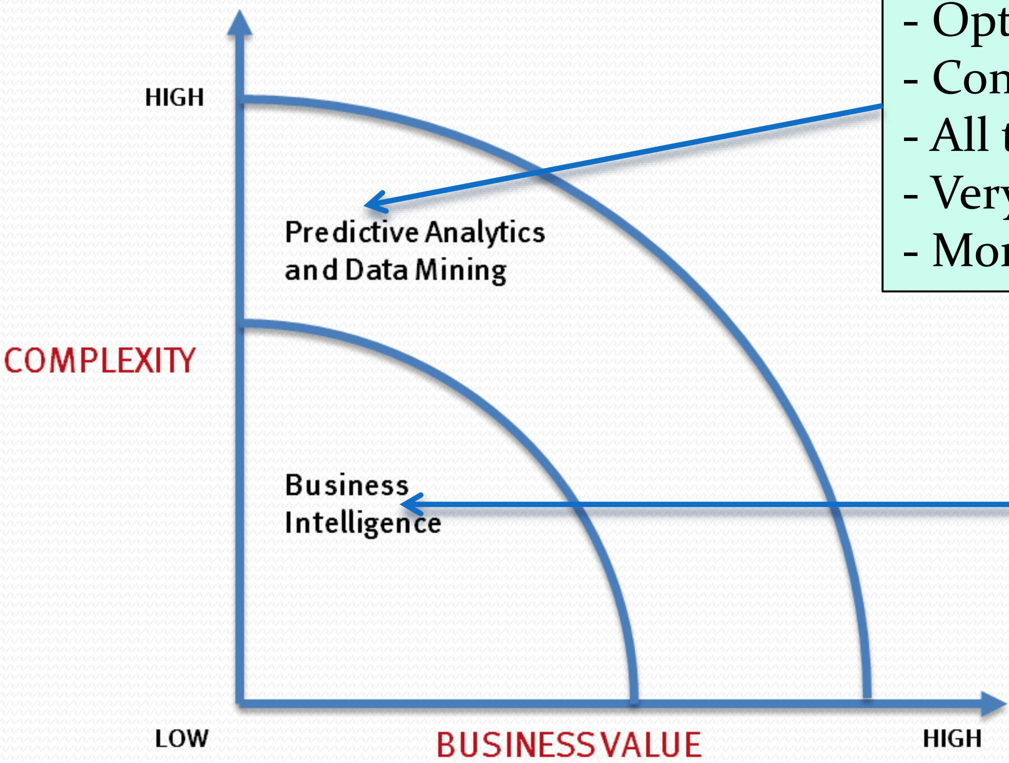




V3: V for Velocity

- Data is being generated fast and need to be processed fast
  - For time-sensitive processes such as catching fraud, big data must be used as it streams into your enterprise in order to maximize its value
  - Scrutinize 5 million trade events created each day to identify potential fraud
  - Analyze 500 million daily call detail records in real-time to predict customer churn faster
- Sometimes, 2 minutes is too late!
  - The latest we have heard is 10 ns (nano seconds) delay is too much

Big data vs. small data



- Optimizations and predictive analytics
- Complex statistical analysis
- All types of data, and many sources
- Very large datasets
- More of a real-time

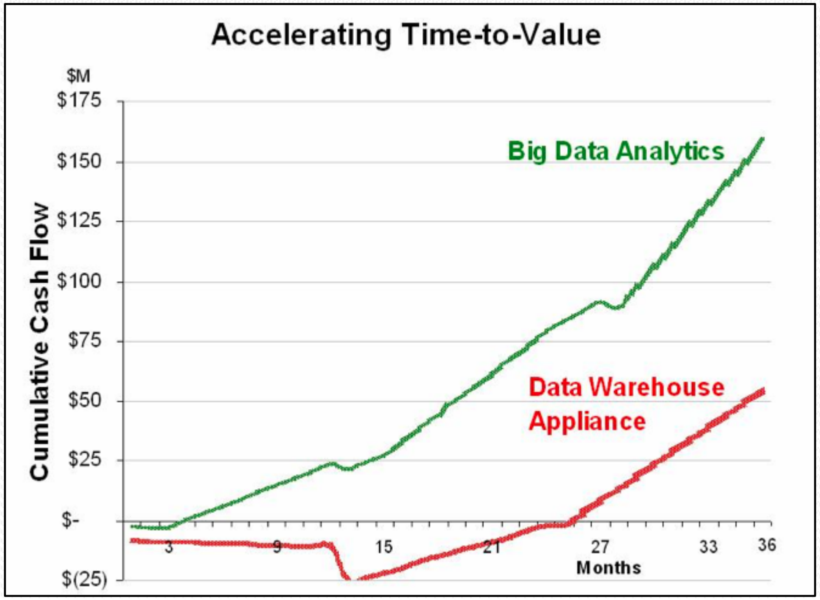
- Ad-hoc querying and reporting
- Data mining techniques
- Structured data, typical sources
- Small to mid-size datasets

Big data vs. small data

- Big data is more real-time in nature than traditional applications

- Big data architecture
  - Traditional architectures are not well-suited for big data applications (e.g. Exa-data, Tera-data)

  - Massively parallel processing, scale out architectures are well-suited for big data applications



Challenges ahead…

- **The Bottleneck is in technology**
  - New architecture, algorithms, techniques are needed
- **Also in technical skills**
  - Experts in using the new technology and dealing with Big data

Who are the major players in the world of Big data?

Big Data Landscape

Vertical Apps
- Predictive Policing
- bloomreach
- MYRRIX

Log Data Apps
- splunk>
- loggly
- sumologic

Ad/Media Apps
- rocketfuel
- collective[i]
- bluefin
- Recorded Future
- LuckySort
- Media Science
- TURN
- DataXu

Data As A Service
- kaggle
- factual.
- knoema
- GNIP
- DATASIFT
- Windows Azure Marketplace
- INRIX
- LexisNexis
- Space Curve
- LOCATE

Business Intelligence
- ORACLE | Hyperion
- SAP
- Business Objects
- RJMetrics
- Microsoft Business Intelligence
- IBM
- COGNOS
- birst
- MicroStrategy
- Autonomy
- QlikView
- bime
- DOMO
- Chart.io
- GoodData

Analytics and Visualization
- tableau
- Palantir
- OPERA
- metaLayer
- METAMARKETS
- datasporacentrifuge
- TERADATA ASTER
- sas
- TIBCO
- KARMASPHERE
- panopticon
- pentaho
- Datameer
- platfora
- ClearStory
- CIRRO
- alteryx
- visual.ly
- AYATA

Analytics Infrastructure
- Hortonworks
- VERTICA
- MAPR
- cloudera
- INFOBRIGHT
- PARACCEL
- EMC²
- GREENPLUM
- NETEZZA
- kognitio
- DATASTAX
- EXASOL
- calpont

Operational Infrastructure
- COUCHBASE
- 10gen
- TERADATA
- HADAPT
- TERRACOTTA
- VoltDB
- MarkLogic
- INFORMATICA

Infrastructure As A Service
- amazon web services
- Windows Azure
- infochimps
- Google BigQuery

Structured Databases
- ORACLE
- MySQL
- SQL Server
- PostgreSQL
- IBM DB2
- SYBASE
- memsql

Technologies
- hadoop
- hadoop MapReduce
- mahout
- APACHE HBASE
- Cassandra

dave@vcdave.com

blogs.forbes.com/davefeinleib

Major players…

- Google
- Hadoop
- MapReduce
- Mahout
- Apache Hbase
- Cassandra

Tools available

- **NoSQL**
  - DatabasesMongoDB, CouchDB, Cassandra, Redis, BigTable, Hbase, Hypertable, Voldemort, Riak, ZooKeeper
- **MapReduce**
  - Hadoop, Hive, Pig, Cascading, Cascalog, mrjob, Caffeine, S4, MapR, Acunu, Flume, Kafka, Azkaban, Oozie, Greenplum
- **Storage**
  - S3, HDFS, GDFS
- **Servers**
  - EC2, Google App Engine, Elastic, Beanstalk, Heroku
- **Processing**
  - R, Yahoo! Pipes, Mechanical Turk, Solr/Lucene, ElasticSearch, Datameer, BigSheets, Tinkerpop

Any question?

Questions of the day…

1. What is the smallest and largest units of measuring size of data?
2. How big a Quintillion measure is?
3. Give the examples of a smallest the largest entities of data.
4. Give FIVE parameters with which data can be categorized as i) simple, ii) Moderately complex and iii) complex?

Questions of the day…

5. What type of data are involved in the following applications?
    1. Weather forecasting
    2. Mobile usage of all customers of a service provider
    3. Anomaly (e.g. fraud) detection in a bank organization
    4. Person categorization, that is, identifying a human
    5. Air traffic control in an airport
    6. Streaming data from all flying aircrafts of Boeing

System/platform level requirements

- How quickly do we need to get the results?
- How big is the data to be processed?
- Does the model building require several iterations or single iteration?
- Will there be a need for more data processing capability in the future?
- Is the rate of data transfer critical for this application?
- Is there a need for handling hardware failures within the application?

Horizontal Scaling

- It involves distributing the workload across many servers which may be even commodity machines.
- It is also known as “scale out”, where multiple independent machines are added together in order to improve the processing capability.
- Typically, multiple instances of the operating system are running on separate machines.

Vertical Scaling

- Vertical Scaling involves installing more processors, more memory and faster hardware, typically, within a single server.
- It is also known as “scale up” and it usually involves a single instance of an operating system.

Table 1 A comparison of advantages and drawbacks of horizontal and vertical scaling

Scaling	Advantages	Drawbacks
Horizontal scaling	➔ Increases performance in small steps as needed	➔ Software has to handle all the data distribution and parallel processing complexities
	➔ Financial investment to upgrade is relatively less	➔ Limited number of software are available that can take advantage of horizontal scaling
	➔ Can scale out the system as much as needed	
Vertical scaling	➔ Most of the software can easily take advantage of vertical scaling	➔ Requires substantial financial investment
	➔ Easy to manage and install hardware within a single machine	➔ System has to be more powerful to handle future workloads and initially the additional performance in not fully utilized
		➔ It is not possible to scale up vertically after a certain limit

Horizontal Scaling Platforms

- Peer-to-Peer Network
- Apache Hadoop
- Apache Spark

Vertical Scaling Platforms

- High performance computing clusters
- Multicore CPU
- Graphics Processing Unit(GPU)
- Field Programmable gate arrays(FPGA)

Peer-to-Peer networks

- involve millions of machines connected in a network
- decentralized and distributed network architecture where the nodes in the networks (known as peers) serve as well as consume resources.
- oldest distributed computing platforms
- Message Passing Interface (MPI) for communication scheme used in such a setup to communicate and exchange the data between peers.
- Each node can store the data instances and the scale out is practically unlimited (can be millions of nodes).

Apache Hadoop

- open source framework for storing and processing large datasets using clusters of commodity hardware.
- Hadoop is designed to scale up to hundreds
- highly fault tolerant
- The Hadoop platform contains the following two important components: (1) HDFS (2) YARN

Apache Spark

- developed by researchers at the University of California at Berkeley. designed to overcome the disk I/O limitations
- ability to perform in-memory computations.
- allows the data to be cached in memory, thus eliminating the
- Hadoop’s disk overhead limitation for iterative tasks.
- supports Java, Scala and Python and for certain tasks
- it is tested to be up to 100× faster than Hadoop MapReduce

HPC clusters

- Known as blades or supercomputers, are machines with thousands of cores.
- They can have a different variety of disk organization, cache, communication mechanism etc.
- powerful hardware which is optimized for speed and throughput.
- They are not as scalable as Hadoop or Spark clusters but they are still capable of processing terabytes of data.

Multicore CPU

- Multicore refers to one machine having dozens of processing cores They usually have shared memory but only one disk.
- the number of cores per chip and the number of operations that a core can perform has increased significantly. Newer breeds of motherboards allow multiple CPUs within a single machine thereby increasing the parallelism.
- Until the last few years, CPUs were mainly responsible for accelerating the algorithms for big data analytics.

GPU

- It is designed to accelerate the creation of images in a frame buffer intended for display output
- GPUs were primarily used for graphical operations such as video and image editing, accelerating graphics-related processing etc. due to their massively parallel architecture, recent developments in GPU hardware and related programming frameworks have given rise to GPGPU
- In addition to the processing cores, GPU has its own high throughput DDR5 memory which is many times faster than a typical DDR3 memory.

FPGA (Field Programmable gate arrays**)**

- highly specialized hardware units for specific applications
- FPGAs can be highly optimized for speed and can be orders of magnitude faster compared to other platforms for certain applications.
- Due to customized hardware, the development cost is typically much higher compared to other platforms.
- On the software side, coding has to be done in HDL with a low-level knowledge of the hardware which increases the algorithm development cost.

Comparison of platforms

System/Platform Level characteristics

- Scalability
- Data I/O performance
- Fault Tolerance

Scalability

Platform	Scalability
Peer-to-Peer	* * * * *
Virtual Clusters(MapReduce/MPI)	* * * * *
Virtual Clusters(Spark)	* * * * *
HPC clusters (MPI/MapReduce)	* * *
Multicore(Multithreading)	* *
GPU(CUDA)	* *
FPGA(HDL)	*

Data I/O Performance

Platform	Data I/O performance
Peer-to-Peer	*
Virtual Clusters(MapReduce/MPI)	* *
Virtual Clusters(Spark)	* * *
HPC clusters (MPI/MapReduce)	* * * *
Multicore(Multithreading)	* * * *
GPU(CUDA)	* * * * *
FPGA(HDL)	* * * * *

Fault Tolerance

Platform	Fault Tolerance
Peer-to-Peer	*
Virtual Clusters(MapReduce/MPI)	* * * * *
Virtual Clusters(Spark)	* * * * *
HPC clusters (MPI/MapReduce)	* * * *
Multicore(Multithreading)	* * * *
GPU(CUDA)	* * * *
FPGA(HDL)	* * * *

Comparison of platforms

Application/Algorithm Level characteristics

- Real time processing
- Data size supported
- Iterative task support

Real Time Processing

Platform	Real Time Processing
Peer-to-Peer	*
Virtual Clusters(MapReduce/MPI)	* *
Virtual Clusters(Spark)	* *
HPC clusters (MPI/MapReduce)	* * *
Multicore(Multithreading)	* * *
GPU(CUDA)	* * * * *
FPGA(HDL)	* * * * *

Data Size supported

Platform	Data Size supported
Peer-to-Peer	* * * * *
Virtual Clusters(MapReduce/MPI)	* * * *
Virtual Clusters(Spark)	* * * *
HPC clusters (MPI/MapReduce)	* * * *
Multicore(Multithreading)	* *
GPU(CUDA)	* *
FPGA(HDL)	* *

Iterative Task Support

Platform	Iterative task support
Peer-to-Peer	* *
Virtual Clusters(MapReduce/MPI)	* *
Virtual Clusters(Spark)	* * *
HPC clusters (MPI/MapReduce)	* * * *
Multicore(Multithreading)	* * * *
GPU(CUDA)	* * * *
FPGA(HDL)	* * * *

- Thank You