* Doug Laney – data complexity *(velocity, volume, variety)*
* Hadoop coined ‘Big data’
* Big data gained recognition around 2010 – data has been growing since
* **Big Data:** dealt with when information required by an organisation cannot be obtained within the necessary timeframes for the organisation to add value to it’s activities.
* Velocity – interpretations needed in real time
* Variety – structured, semi-structured and **unstructured**
* Structured – payroll, finance, ERP (relational database in tabular structure)
* Semi-structured – large volume, small size of individual records and simple structure i.e. intelligent power meter to central system
* Unstructured – many different structures i.e. social media, audio, video, graphics, text, PDF, email, word

Hadoop

* Open-source framework for large scale data processing (managed by Apache)
* Operates over number of nodes – increasing reliability, speed of processing
* Large datasets can be processed on commodity hardware rather than high-end servers
* Doesn’t require upfront investment – splits data processing amongst machines and coordinates actions
* Hadoop commons – libraries and resources used by other modules – supports other modules within Hadoop installation
* **HDFS**- store data across cluster of commodity hardware; allows for fault tolerant, distributed storage across commodity hardware (consists of namenodes and datanodes)
* YARN – resource managing in cluster and job management
* Functionalities of YARN were in Mapreduce but moved to allow it to focus on data processing
* MapReduce – key-value pairs
* Apache pig- creating mapReduce programs
* Apache Hive – builds data warehouses on top of Hadoop – data query, analysis ETL, QL, access to Hbase or HDFS files
* Hbase – distributed non-relational database which stores and processes sparse data. High performance DB on top of HDFS.
* Apache spark- open source analytics framework – faster than mapreduce
* Hadoop can support any type of database - can run OLAP or OLTP on top but wouldn’t do it from scratch. Would go with MongoDB if I wanted to start from scratch.

MongoDB

* A document database that provides high performance, availability and easy scalability
* Features: replication, data across machines, embedded structures, document oriented architecture
* NoSQL open-source database for large data processing tasks; document oriented storage – documents combined into collections
* Documents – records in a relational database
* Collections – tables
* Collection is a group of documents
* Don’t modify underlying design – more flexible than relational databases
* Auto-sharding – storing data across multiple nodes
* Replication – same data, several locations for high protection
* MapReduce – large data sets condensed into aggregated results
* Full index support – supports indexes on any attribute type and number of attributes

Document databases

* Flexible schema – don’t enforce structure before inserting data
* Collections correspond to tables, documents to records and fields to columns
* Collections are only within single database
* Documents – data structures consisting of field value pairs
* Embedded documents store related data in single document structure – better read performance but may impact write performance leading to fragmentation
* Fields – cannot be $. NULL and must be unique

Piracy, Privacy and Ethics

* Big data solutions need to evolve to counter new strategies employed by pirates
* Challenges to privacy: complex environment, solutions outside Information systems, higher risk, reidentification not effective
* Need to anonymise data, bring issues to boards attention, policies cover new data, governance (educate business)
* Consumers need informing when using data, and purpose of data collection communicated
* Big data projects – scary people, IT professionals, C-suite members, domain experts
* Stages – initiation, support, identify resources, develop solution, train, deploy

Enterprise Data Sources

* Different apps supporting business operations
* Integrate and retrieve data for internal and external communication
* Ensures trust and confidence in data assets
* Based on relational database tech – well structured/accessible
* Oracle (HR/Payroll, finance CRM), SAP (HR/Payroll, SAP suite), and Microsoft ( CRM, ERP)
* OBDC, JDBC and flat lines – access mechanisms
* SAP accessed through connector – harder to extract data as it uses a proprietary language (ABAP)

Data warehouses

* Based on relational database technology using ODBC, JDBC or flat files
* Information from number of systems in organisation – convenient location to access multiple datasources
* Lacks detail
* Unstructured data – corporate knowledge outside enterprise systems (email, pdf, word)
* Metadata – data about data

Facebook

* Access data streams – public feed API (limited set of media publishers), keyword insights API (access to analytics managed by facebook), graph API (nodes ie posts, photos and users/ edges ie connections between nodes/ fields ie node attributes).

Data Mining

* Data mining - discovering patterns in large datasets – classification, association and clustering
* Classficiation – common traits extracted based on a set where traits already identified (supervised) – decision trees, rules, neural networks
* Association – dependencies between attributes – may not be part of rule set; coverage and accuracy
* Clustering – common attributes – kmeans
* Data mining tools – weka, KNIME, R
* Weka – written in Java (uni of Waikato) – explorer, CLI, experimenter, knowledge flow (modules)
* KNIME – uni of Konstanz; based on eclipse
* R – statistical computing

Hadoop Installation

* Types of installation – local, pseudo distributed, fully distributed