POZNAN UNIVERSITY OF TECHNOLOGY

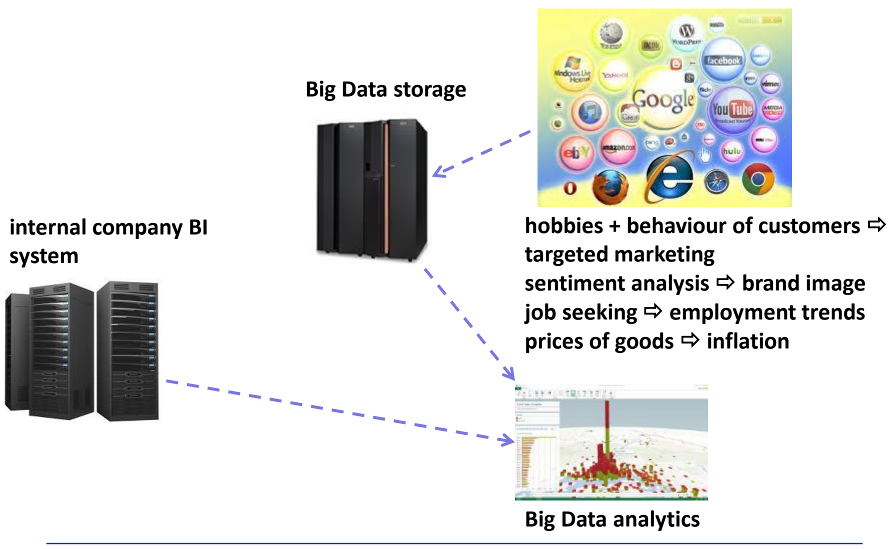
Data Warehouses and Business Intelligence: Big Data

Robert Wrembel
Politechnika Poznańska
Instytut Informatyki
Robert.Wrembel@cs.put.poznan.pl
www.cs.put.poznan.pl/rwrembel

Outline

- **Introduction to Big Data**
- **Big Data Architectures**
- **GFS, HDFS, Hadoop**
- **Some other hot trends**

Big Data



Big Data

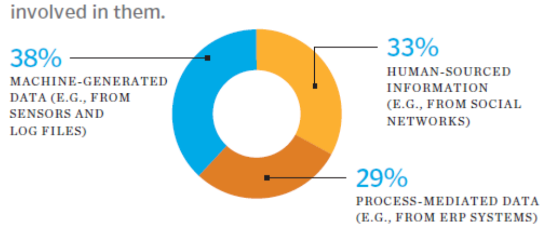
- **Huge Volume**
- **Every minute:**
  - **48 hours of video are uploaded onto Youtube**
  - **204 million e-mail messages are sent**
  - **600 new websites are created**
  - **600000 pieces of content are created**
  - **over 100000 tweets are sent (~ 80GB daily)**
- **Sources:**
  - **social data**
  - **web logs**
  - **machine generated**

Big Data

➲ Sensors

- mechanical installations (refineries, jet engines, crude oil platforms, traffic monitoring, utility installations, irrigation systems)
  - one sensor on a blade of a turbine generates 520GB daily
  - a single jet engine can generate 10TB of data in 30 minutes
- telemedicine
- telecommunication

The percentage of big data projects by the types of data involved in them.



Source: *Operationalizing the Buzz: Big Data 2013*, Enterprise Management Associates Inc. and 9sight Consulting; based on an online survey of 259 business and IT professionals

Big Data

➲ High **Velocity** of

- data volume growth
- uploading the data into an analytical system

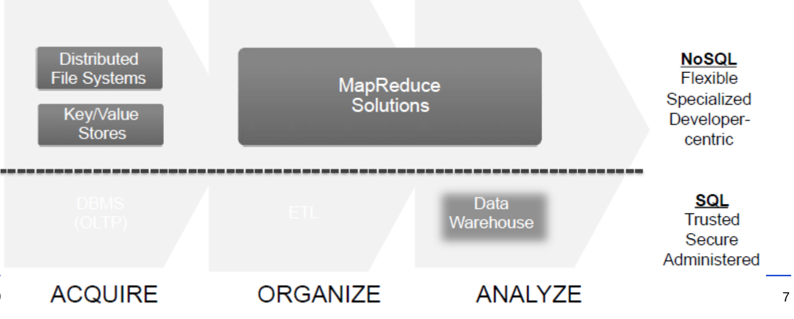
➲ **Variety** (heterogeneity) of data formats

- structured - relational data and multidimensional cube data
- unstructured or semistructured - text data
- semantic Web XML/RDF/OWL data
- geo-related data
- sensor data

➲ **Veracity (Value)** - the quality or reliability of data

Big Data - Problems

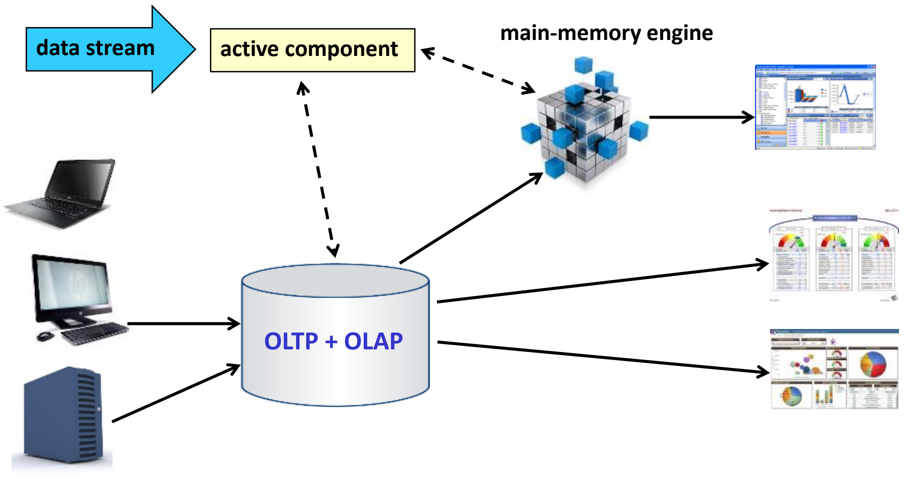
- **Storage**
  - **volume**
  - **fast data access**
  - **fast processing**
- **Real-time analysis**
  - **analyzing fast-arriving streams of data**



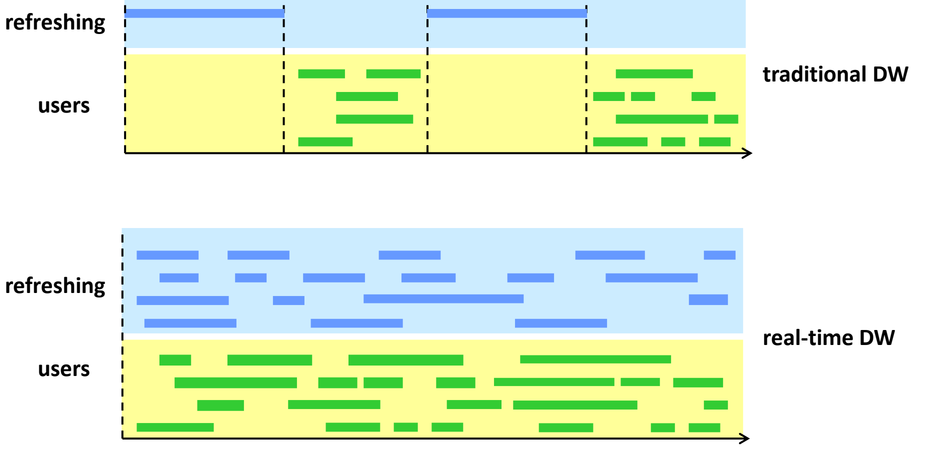
Types of processing

- **Batch processing - standard DW refreshing**
- **Real-time / near real-time data analytics**
  - **answers with the most updated data up to the moment the query was sent**
  - **the analytical results are updated after a query has been executed**
- **Streaming analytics**
  - **a system automatically updates results about the data analysis as new pieces of data flow into the system**
  - **as-it-occurs signals from incoming data without the need to manually query for anything**

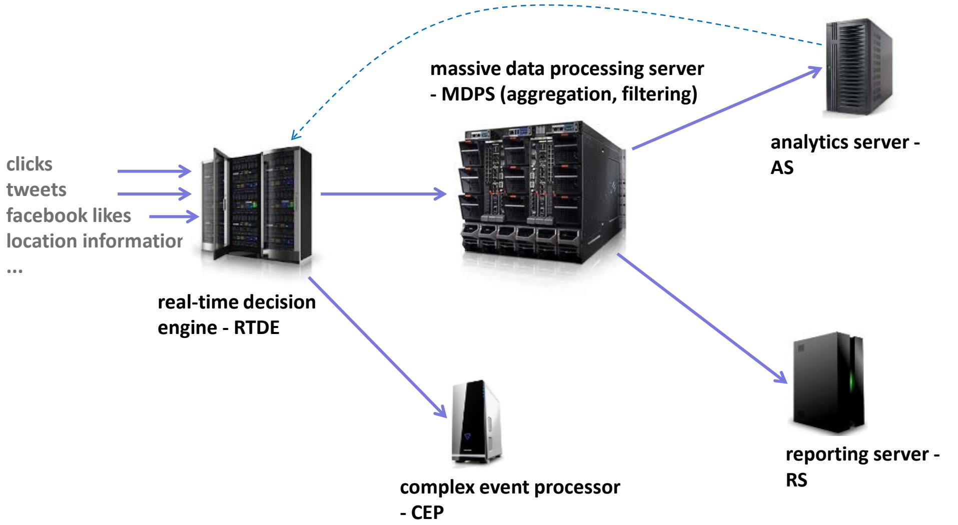
Real-time / Near real-time architecture



Real-time / Near real-time refreshing

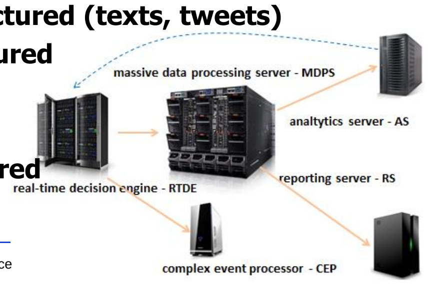


Big Data Architecture



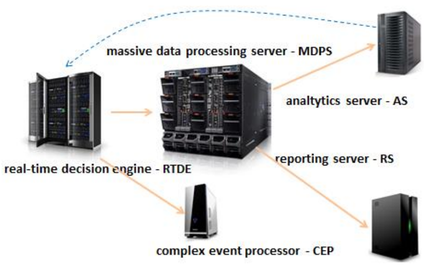
Big Data Architecture

- **Scalability**
  - **RTDE - nb of events handled**
  - **MDPS - volume of data and frequency of data processing**
  - **AS - complexity of computation, frequency of queries**
  - **RS - types of queries, nb of users**
  - **CEP - # events handled**
- **Type of data**
  - **RTDE - unstructured, semistructured (texts, tweets)**
  - **MDPS – semistructured, structured**
  - **AS - structured**
  - **RS - structured**
  - **CEP - unstructured and structured**



Big Data Architecture

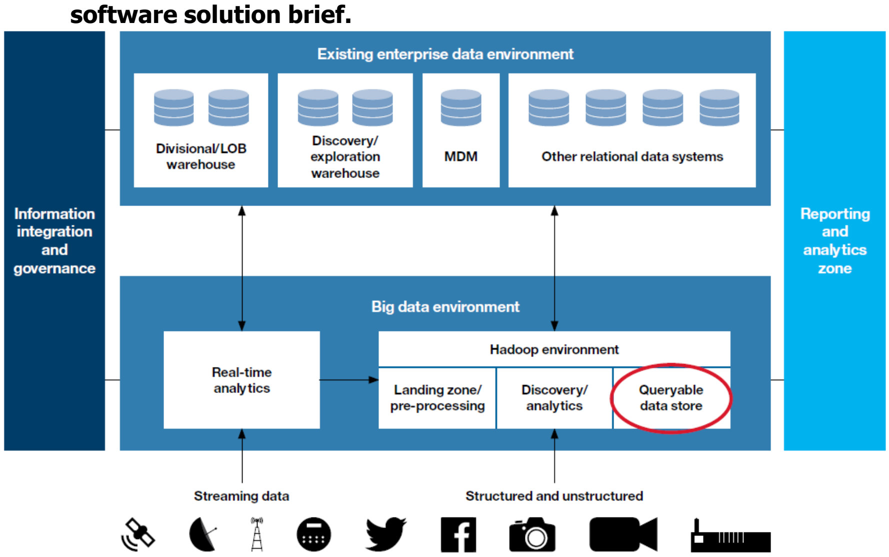
- Workload
  - RTDE - high write throughput
  - MDPS - long-running data processing (I/O and CPU intensive): data transformations, ...
  - AS - compute intensive (I/O and CPU intensive)
  - RS - various types of queries
- Technologies
  - RTDE - key-value, in-memory
  - MDPS - Hadoop
  - AS - in-memory, columnar DBs
  - RS - in-memory, columnar DBs



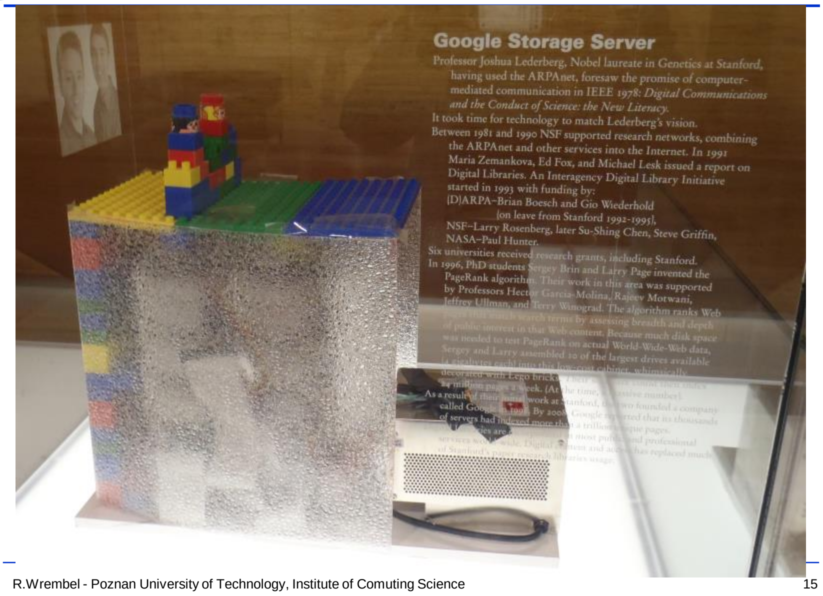
- Conclusion
  - very complex architecture with multiple components
  - the need of integration

IBM Architecture

- Data warehouse augmentation: the queryable data store. IBM software solution brief.

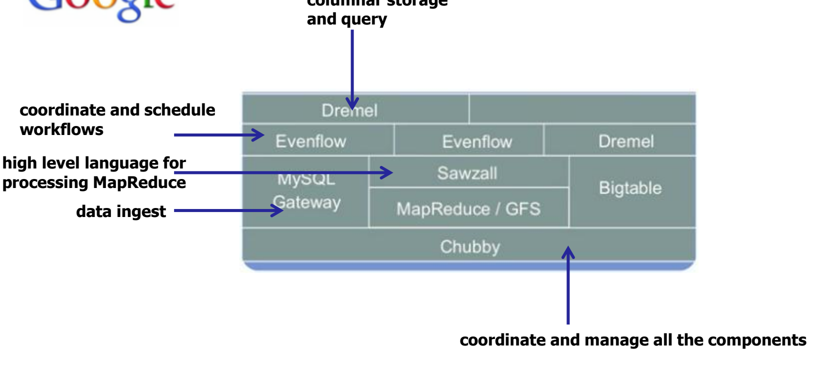


Big Data Architecture



Big Data Architecture

Google



http://www.cloudera.com/content/cloudera/en/resources/library/training/apache-hadoop-ecosystem.html

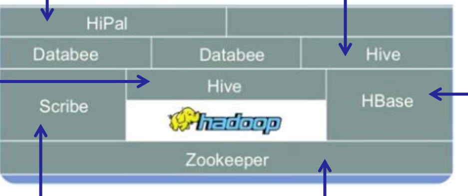
Big Data Architecture

facebook

user interface to Hive

SQL-like language for data analysis
supports selection, join, group by, …

high level language for
processing MapReduce



columnar storage
and query
based on BigTable
manages TB of data

web log loader (log scrapper),
periodical loading into Hadoop
aggregating log entries (for offline
analysis)

coordinate and manage all the components
service discovery, distributed locking, …

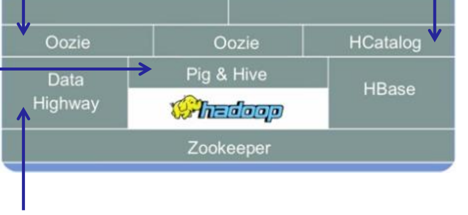
Big Data Architecture

YAHOO!

coordinate and schedule
workflows
schedule and manage Pig, Hive, Java,
HDFS actions

RDB-like interface to data stored
in Hadoop

high level languages for
processing MapReduce



distributed web log loader (log scrapper),
periodical loading into Hadoop (for offline
analysis)

Big Data Architecture

Linkedin

workflow (batch job) scheduler
(e.g., data extraction, loading into Hadoop)

SQL to Hadoop: command line tool for importing any JDBC data source into Hadoop

distributed web log loader and aggregator (in real time)

table-like data storage + in memory caching

managing the services, e.g., detercting addition or removal of Kafka's brokers and consumers, load balancing

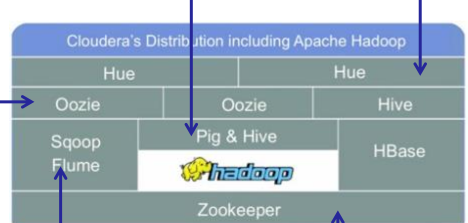
Big Data Architecture

cloudera

UI for Hadoop (e.g., HDFS file browser, MapReduce job designer and browser, query interfaces for Hive, Pig, Impala, Oozie, application for creating workflows, Hadoop API)

high level languages for processing MapReduce

workflow coordination and scheduling



distributed web log loader (log scrapper), periodical loading into Hadoop (for offline analysis)

coordinate and manage all the components

Big Data Architecture

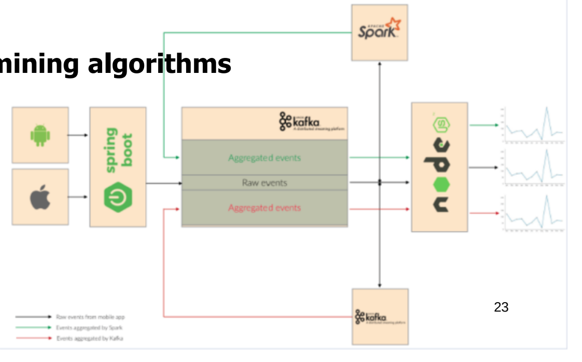
Microsoft **Windows Azure**

Tomasz Kopacz - Microsoft Polska: prezentacja Windows Azure, Politechnika Poznańska, czerwiec 2013

Data Ingest (ETL)

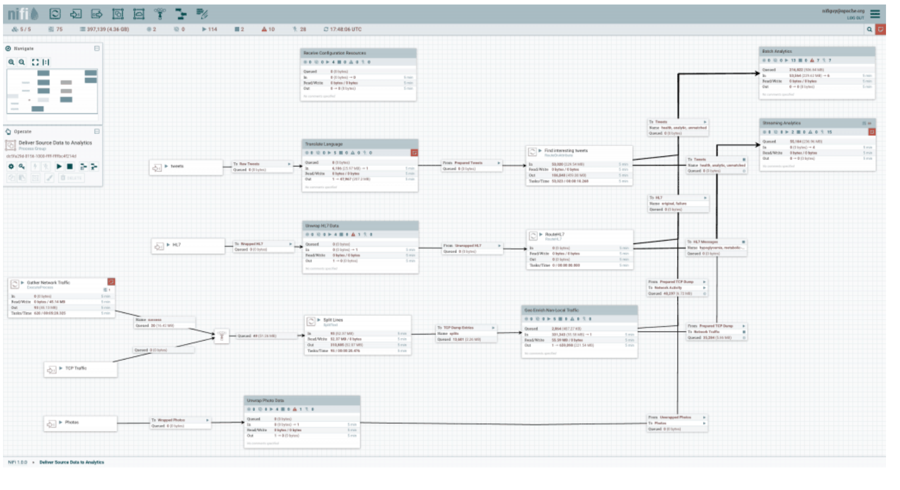
- **Kafka**
- **Storm**
- **Flink**
- **Sqoop**
- **NiFi**

- ➲ **Kafka Streams**
    - ▪ **event by event reading**
    - ▪ **Java**
    - ▪ **aggregation in a sliding window**
    - ▪ **no built-in stream mining algorithms**
- ➲ **Spark Streaming**
    - ▪ **micro batches**
    - ▪ **built-in stream mining algorithms**



NiFi

- ➲ **Purpose: to automate the flow of data between multiple systems → similar to ETL**
- ➲ **Asynchronous: for very high throughput and slow processing buffering may be used**



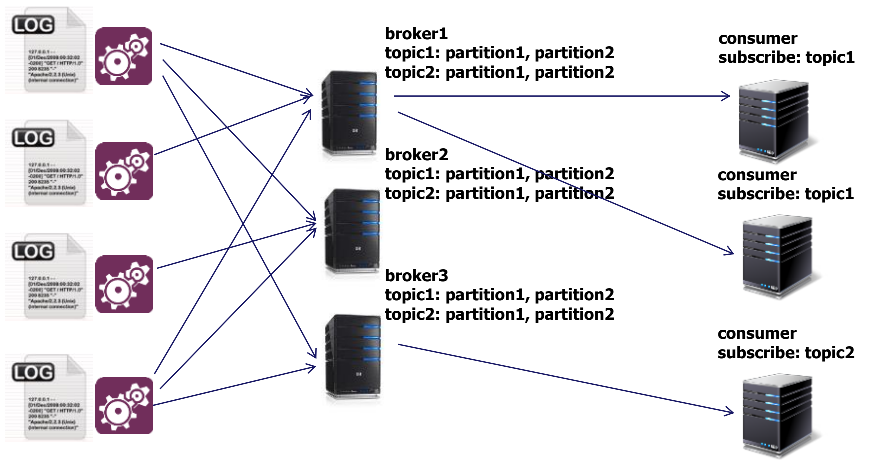
NiFi building blocks

- Processors
  - process data delivered as FlowFiles
- FlowFile
  - represents data moved within NiFi, represented as key-value
- Connection
  - connect processors, serves as a queue - buffering, different processors may read from the queue at differing rates
- Flow Controller
  - acts as a broker facilitating the exchange of FlowFiles between processors
- Process Group
  - is a set of processes and connections, which can receive data via input ports and send data out via output ports

ETL for Big Data - Kafka

- Distributed queuing/messaging
- Handling 1 000 000 000 messages daily
- Used for transferring data from WEB logs in real time
- Terms
  - a topic: stream messages of particular type, divided into partitions
  - a producer: publishes a given topic
  - a consumer: subscribes to one or more topics
  - a broker: stores topics for their distribution to consumers

Kafka



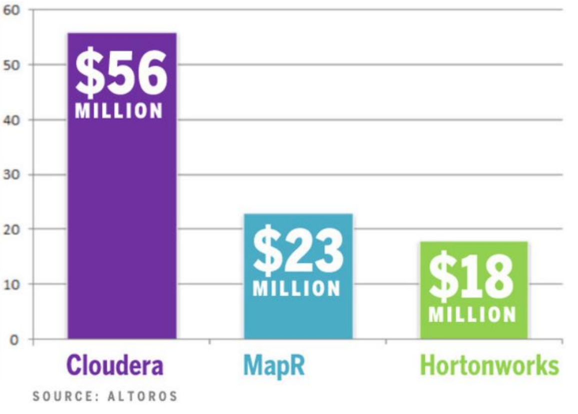
- Multiple brokers (broker cluster) for load balancing

Kafka

- **Consumer maintains the info about read topic's partitions**
- **Broker deletes partitions after a given time period regardless they have been read by a consumer or not ⇨ possible data lost**
- **At-least-once delivery model in the case of consumer failure**
  - **after restart a consumer may re-read the last topic's partition ⇨ duplicates**
- **The order of messages in a partition is preserved within a delivery**
- **The order of inter-partition delivery from different brokers is not preserved (e.g., read partition2 from broker3 then read partition1 from broker2)**

Hadoop Distributions

➲ **Cloudera, MapR, Hortonworks, IBM, Pivotal Software**



Data Stores

➲ **NoSQL**

➲ **Key-value DB**

- **data structure ⇨ collection, represented as a pair: key and value**
- **data have no defined internal structure ⇨ the interpretation of complex values must be made by an application processing the values**
- **operations ⇨ create, read, update (modify), and delete (remove) individual data - CRUD**
- **the operations process only a single data item selected by the value of its key**
- **Voldemort, Riak, Redis, Scalaris, Tokyo Cabinet, MemcacheDB, DynamoDB**

Data Stores

- Column family (column oriented, extensible record, wide column)
  - definition of a data structure includes
    - key definition
    - column definitions
    - column family definitions

	column family CF1			column family CF2	
	Col1	Col2	Col3	Col4	Col5
row key K1	value	value			value
row key K2		value	value		value
row key K3	value	value	value	value	value
row key K4					
row key Kn	value			value	

Data Stores

- column family ⇨ stored separately, common to all data items (~ shared schema)
- column ⇨ stored with a data item, specific for the data item
- CRUD interface
- HBase, HyperTable, Cassandra, BigTable, Accumulo, SimpleDB

Data Stores

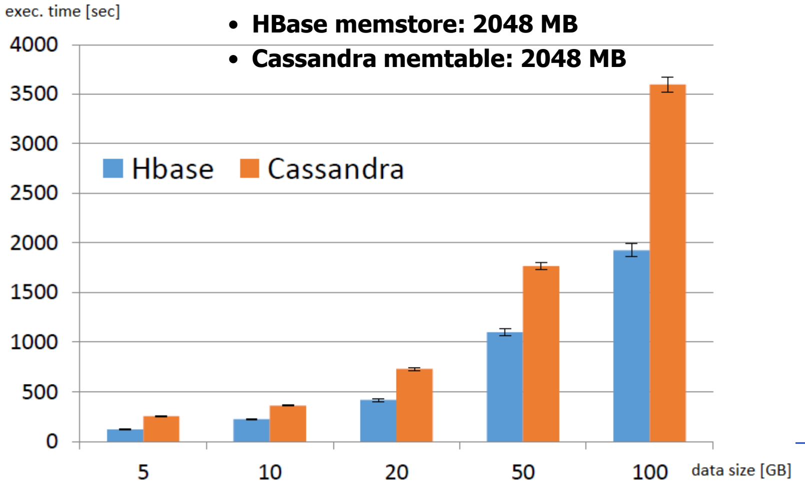
- Document DB
  - typically JSON-based structure of documents
  - SimpleDB, MongoDB, CouchDB, Terrastore, RavenDB, Cloudant
- Graph DB
  - nodes, edges, and properties to represent and store data
  - every node contains a direct pointer to its adjacent element
  - Neo4j, FlockDB, GraphBase, RDF Meronymy SPARQL

Performance evaluation

- A. Rusin, A. Szymczak: master level term project (2015)
- HBase ⇔ Cassandra
  - virtual machines 8 CPUs, 16 GBs RAM, 480 GB HDD
  - Ubuntu (14.04.1 LTS)
  - Cassandra 2.0.14
  - HBase 1.0.0 + Hadoop 2.5.2
  - 2 Cassandra data nodes
  - 2 (HBase RegionServer + Hadoop DataNode) + 1 (HBase MasterServer + Hadoop NameNode)
  - Yahoo Cloud Serving Benchmark with modified workloads

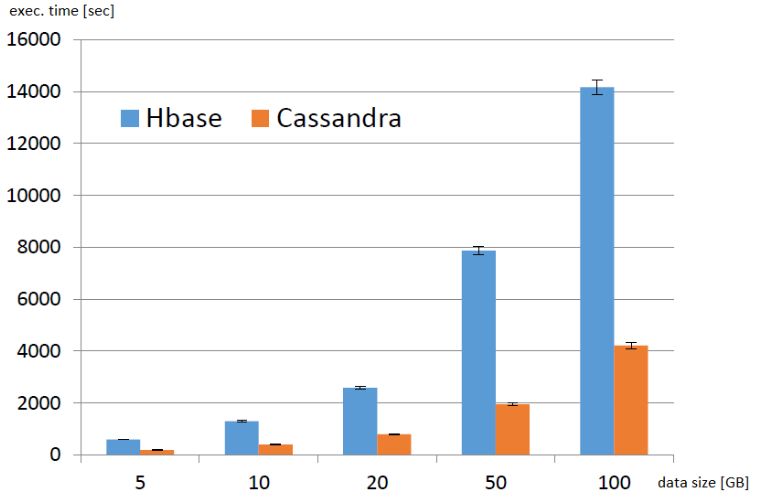
HBase - Cassandra

- Read-only workload
  - # of threads for HBase and Cassandra: 256
  - cache size:
    - HBase memstore: 2048 MB
    - Cassandra memtable: 2048 MB



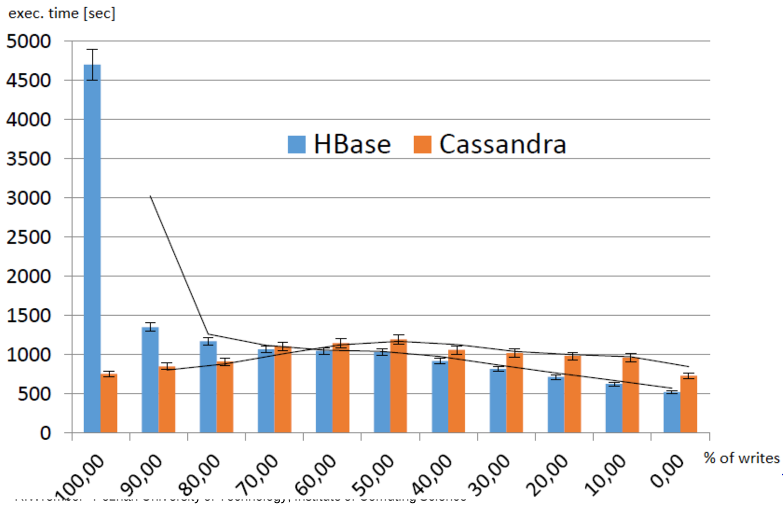
HBase - Cassandra

- Write-only workload



HBase - Cassandra

- Read-write workload
  - data volume: 20 GB



GFS

- **Google implementation of DFS** (cf. The Google File System - whitepaper)
- **Distributed FS**
- **For distributed data intensive applications**
- **Storage for Google data**
- **Installation**
  - **hundreds of TBs of storage, thousands of disks, over a thousand cheep commodity machines**
- **The architecture is failure sensitive ⇨ therefore**
  - **fault tolerance**
  - **error detection**
  - **automatic recovery**
  - **constant monitoring is required**

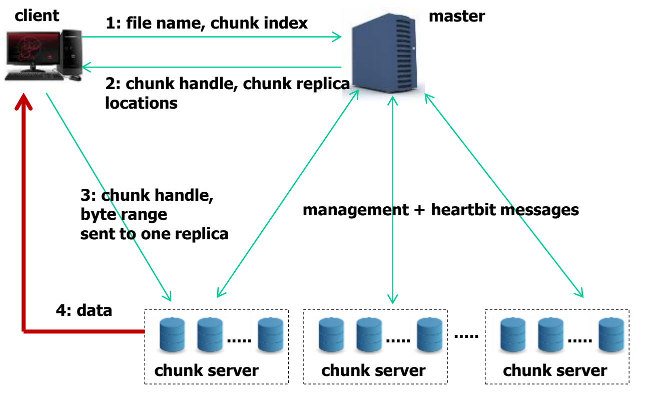
GFS

- ➲ **Typical file size: multiple GB**
- ➲ **Operations on files**
  - **mostly appending new data ⇨ multiple large sequential writes**
  - **no updates of already appended data**
  - **mostly large sequential reads**
  - **small random reads occur rarely**
  - **file size at least 100MB**
  - **millions of files**

GFS

- ➲ **Files are organized hierarchically in directories**
- ➲ **Files are identified by their pathnames**
- ➲ **Operations on files: create, delete, open, close, read, write, snapshot (creates a copy of a file or a directory tree), record append (appends data to the same file concurrently by multiple clients)**
- ➲ **GFS cluster includes**
  - **single master**
  - **multiple chunk servers**

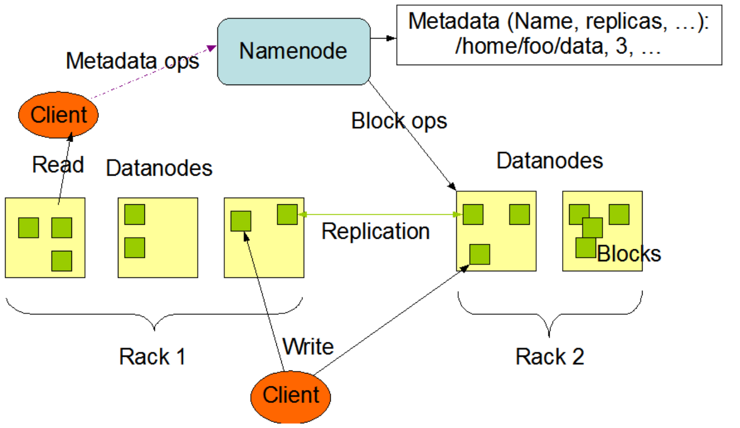
GFS



S. Ghemawat, H. Gobioff, S-T. Leung. The Google File System. http://research.google.com/archive/gfs.html

HDFS

- **Apache implementation of DFS**
  http://hadoop.apache.org/docs/stable/hdfs_design.html



Storage

- ➲ **Distributed file systems**
  - **Amazon Simple Storage Service (S3)**
  - **Gluster**
- ➲ **Storage formats**
  - **Apache Avro for storing serialized data in JSON for Hadoop**
  - **Apache Parquet - column oriented data store for Hadoop**

Example

- ➲ **In 2010 Facebook stored over 30PB in Hadoop**
- ➲ **Assuming:**
  - **30,000 1TB drives for storage**
  - **typical drive has a mean time between failure of 300,000 hours**
  - **2.4 disk drive fails daily**

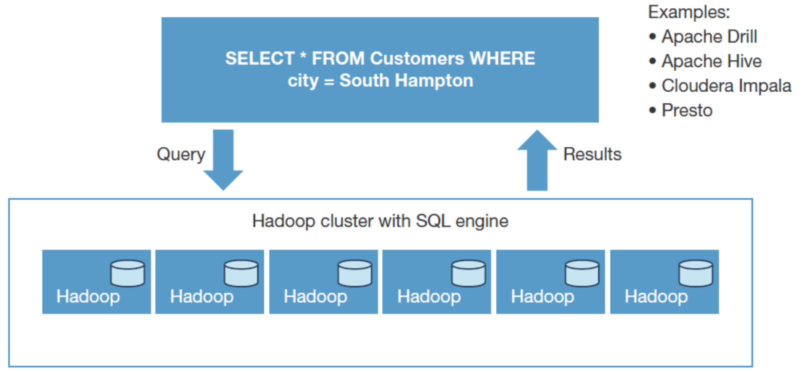
Integration with Hadoop

- IBM BigInsights ⇨ Cloudera distribution + IBM custom version of Hadoop called GPFS
- Oracle BigData ⇨ appliance based on Cloudera for storing unstructured content
- Informatica HParser ⇨ to launch Informatica process in a MapReduce mode, distributed on the Hadoop servers
- Microsoft ⇨ dedicated Hadoop version supported by Apache for Microsoft Windows and for Azure
- EMC Greenplum, HP Vertica, Teradata Aster Data, SAP Sybase IQ ⇨ provide connectors directly to HDFS

Integration with Hadoop

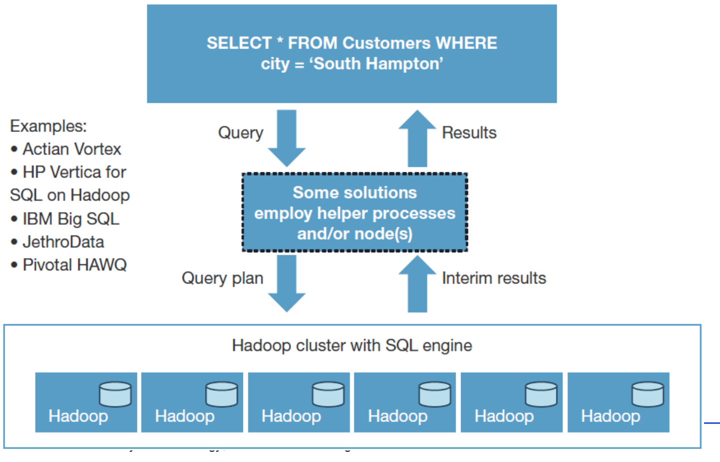
- M.Gualtieri, B. Hopkins: SQL-For-Hadoop: 14 Capable Solutions Reviewed. Forrester, 2015

- **Pure SQL for Hadoop**



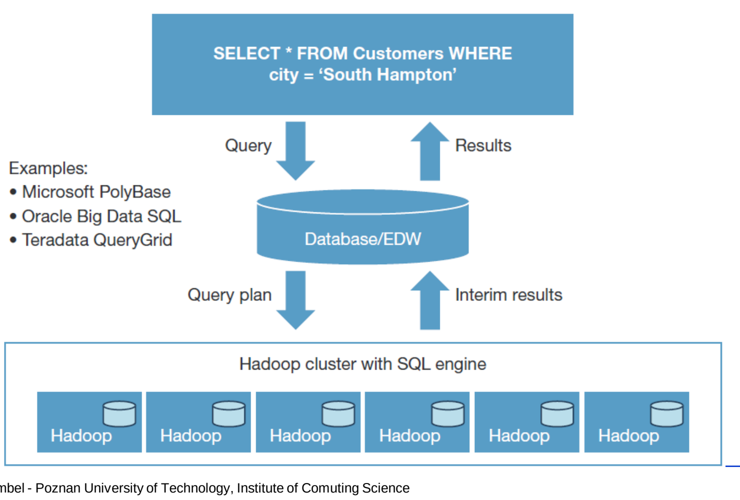
Integration with Hadoop

- **Boosted SQL for Hadoop**
- Typically include: query parser and optimizer
- Require more strucutred data to exploit the power of SQL



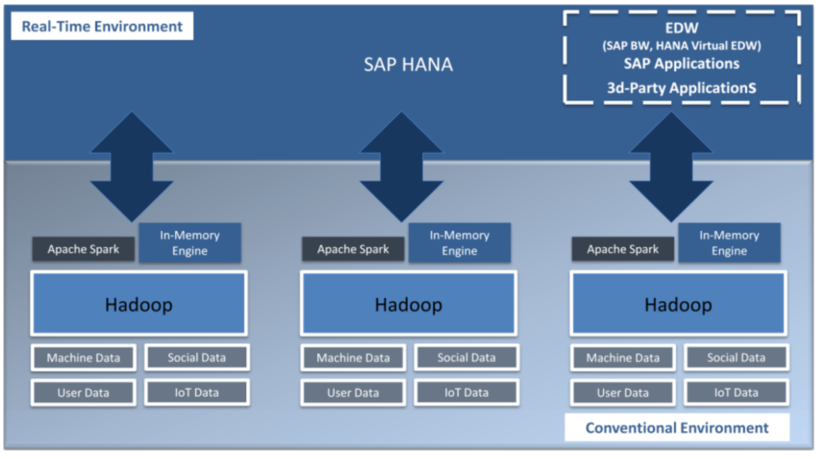
Integration with Hadoop

- **Database + Hadoop**
- Hadoop files accessed via external tables from a DB



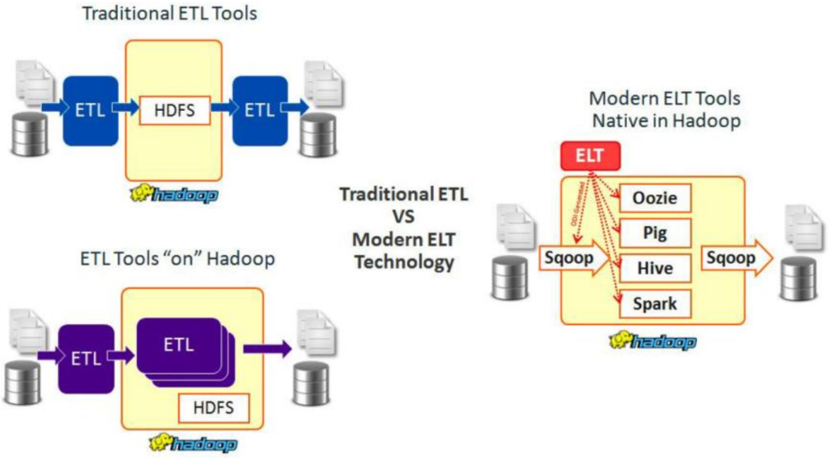
Integration with Hadoop

- SAP Vora: HANA + Spark + Hadoop



- The Contextual Data Lake. By SAP, available at: https://tdwi.org/whitepapers/2015/10/the-contextual-data-lake.aspx

ETL



- The Five Most Common Big Data Integration Mistakes To Avoid. ORACLE white paper, 2015

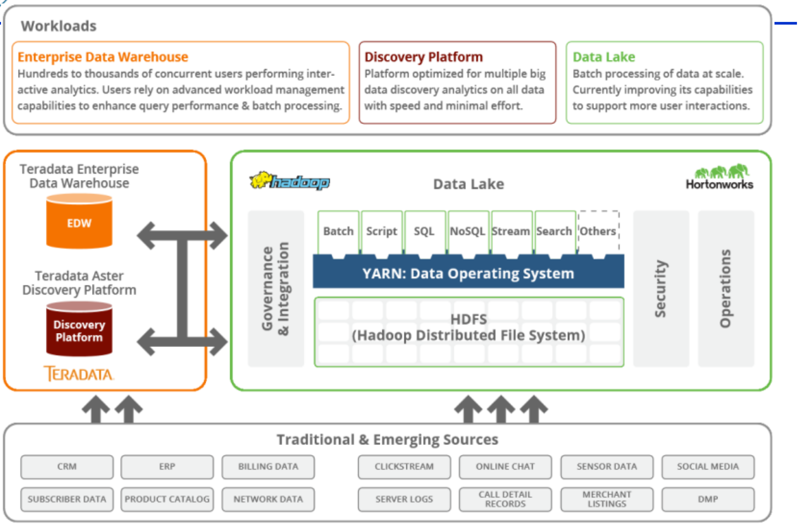
Hadoop-based DWs

- Impala, Stinger, Apache Drill, Phoenix, Shark, Hadapt
- Teradata SQL-H, EMC HAWQ, IBM BigSQL

Data Lake

- A repository that stores a vast amount of **raw data** in its **native format** until it is needed
- Each data element in a lake is assigned a unique identifier and tagged with a set of metadata
- Often implemented based on **Hadoop**
- No schema on write schemas of data are not defined (considered) while writing to a data lake
- The schema is obtained when data are queried → **schema on read**

Data Lake Architecture



- Image taken from: Putting the Data Lake to Work - a Guide to Best Practices. CITO Research, April 2014

ETL for Big Data

streams:
simple records
(sequences, time series)
images
sounds
videos
texts
(unstruct., semi-struct.)
periodically uploaded

E(T)L

data lake

Data Lake

- ➲ No schema on write
- ➲ Schema on read
  - the need to understand the content ⇨ metadata
- ➲ Data lake content
  - relational tables
  - WEB tables
  - XML
  - texts
  - images, sounds, videos
  - graphs
  - ... any existing format

Data Lake

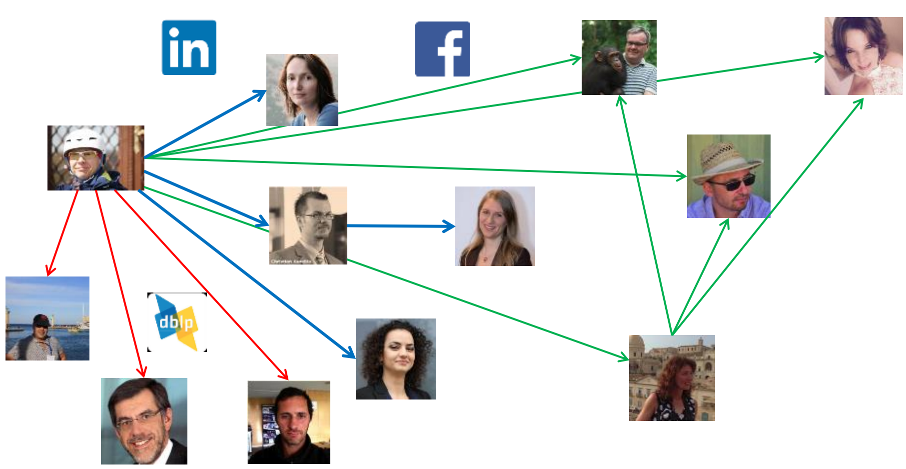
- ➲ Querying a data lake
  - a query language and query engine capable of expressing and processing a query, possibly expressed in a natural language
  - finding relevant data sources for a query
    - relevant "schema"/structure
    - relevant content
    - correlating multiple data sources of the same semantics
    - selecting the most reliable data sources
  - finding the relevant data sources quickly

Data Lake

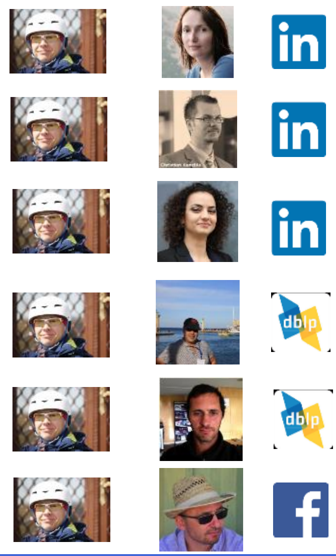
- **Querying a data lake**
  - **efficiently retrieving subsets of data for a query**
    - **data of high quality**
  - **transforming data on the fly (during a query execution) into a common format**
  - **integrating data on the fly**
  - **choosing appropriate ways of visualizing the results**
  - **scalability ⇨ performance**

Novel search methods

- **Find colleagues of Robert Wrembel**



Novel search methods



busniess

busniess

busniess

research

research

social

Novel search methods

- Correlating and combining multiple data sources of different formats
- Information about which DSs were used to answer a query
- Information about the quality of the used DSs

find **colleagues** of Robert Wrembel
context business, research, social
output graph | table

Collaborative BI

- ➲ **Annotating results of analyses**
- ➲ **Searching for the results of previous analyses**

Metadata

- ➲ **Extensive usage of metadata**
  - **schema/structure**
    - **semantics of properties**
  - **content**
    - **semantics of values**
  - **transformation rules**
  - **visualization**
  - **performance**
- ➲ **Data annotation during E(T)L**
- ➲ **Data profiling in a data lake**
- ➲ **Incremental maintenance of metadata**
- ➲ **Metadata standard?**
  - **CWM for relational systems**
  - **? for data lakes**

RDBMS vs. NoSQL: the Future?

- **TechTarget: Relational database management system guide: RDBMS still on top**
  - **http://searchdatamanagement.techtarget.com/essentialguide/Relational-database-management-system-guide-RDBMS-still-on-top**
- **"While NoSQL databases are getting a lot of attention, relational database management systems remain the technology of choice for most applications„**
- **S. Ghandeharizadeh: SQL, NoSQL, and Next Generation Data Stores. Keynote talk at DEXA 2015**
  - **RDBMS will be important components of IT infrastructures**

RDBMS vs. NoSQL: the Future?

- **R. Zicari: Big Data Management at American Express. Interview with Sastry Durvasula and Kevin Murray. ODBMS Industry Watch. Trends and Information on Big Data, New Data Management Technologies, and Innovation. Oct, 2014, available at: http://www.odbms.org/blog/2014/10/big-data-management-american-express-interview-sastry-durvasula-kevin-murray/**
  - **"The Hadoop platform indeed provides the ability to efficiently process large-scale data at a price point we haven't been able to justify with traditional technology. That said, not every technology process requires Hadoop; therefore, we have to be smart about which processes we deploy on Hadoop and which are a better fit for traditional technology (for example, RDBMS)."–Kevin Murray.**

Enterprise Data Warehouse

- The Contextual Data Lake. By SAP, available at: https://tdwi.org/whitepapers/2015/10/the-contextual-data-lake.aspx
- "... companies will retain an EDW as part of their overall data architecture ..."

RDBMS

- **Conceptual and logical modeling methodologies and tools**
- **Rich SQL functionality**
- **Query optimization**
- **Concurrency control**
- **Data integrity management**
- **Backup and recovery**
- **Performance optimization**
  - **buffers' tuning**
  - **storage tuning**
  - **advanced indexing**
  - **in-memory processing**
- **Application development tools**

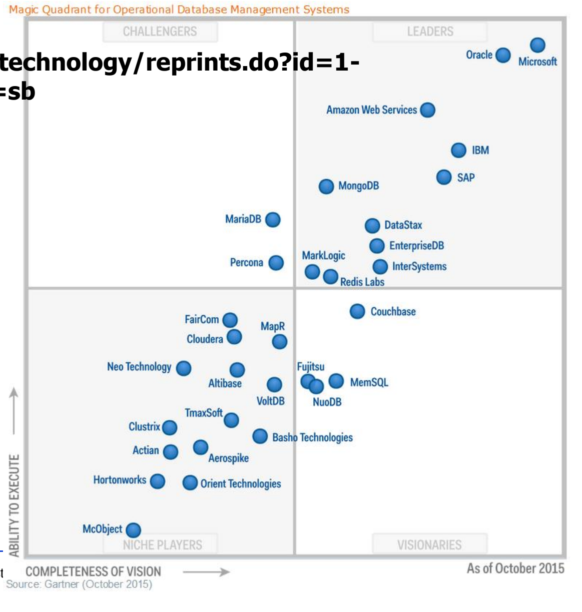
NoSQL

- ➲ Flexible "schema" ⇨ suitable for unstructured data
- ➲ Massively parallel processing
- ➲ Cheap hardware + open source software

- ➲ Choosing the right NoSQL database for the job: a quality attribute evaluation. Journal of Big Data; http://www.journalofbigdata.com/content/2/1/18

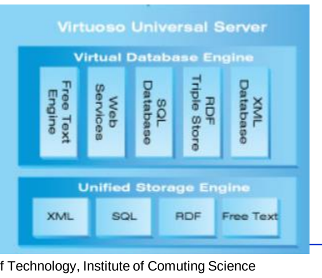
Gartner Report

- ➲ http://www.gartner.com/technology/reprints.do?id=1-2PMFPEN&ct=151013&st=sb



Some other trends

- Apache Derby: Java-based ANSI SQL database
- Splice Machine
  - Derby (redesigned query optimizer to support parallel processing) on HBase (parallel processing) + Hadoop (parallel storage and processing)
- Apache Phoenix
  - relational-like DB on HBase
  - SQL interface
- Virtuoso



Some other trends

- Web Table: https://research.google.com/tables?hl=en
- Google Knowledge Graph

Trends cont.

- ➲ **Analyzing Twitter posts**
  - **Google flu trend maps**
    - **http://www.slate.com/blogs/moneybox/2014/04/04/twitter_economics_university_of_michigan_stanford_researchers_are_using.html**
    - **"Google tracks flu activity around the world by monitoring how many people are searching flu-related terms in different areas, and with what frequency. "We have found a close relationship between how many people search for flu-related topics and how many people actually have flu symptoms""**
  - **tweets on unemployment well correlate with real governmental data**
    - **http://www.washingtonpost.com/blogs/wonkblog/wp/2014/05/30/the-weird-google-searches-of-the-unemployed-and-what-they-say-about-the-economy/**

Trends cont.

- ➲ **Big Data integration and cleaning ⇨ to get correct data**
- ➲ **Text analytics**
  - **summarization**
  - **sentiment**
- ➲ **Tracking the evolution of entities in the Internet over time**
- ➲ **ACM SIGMOD Blog**
  - **http://wp.sigmod.org/**

Trends cont.

- Top 8 Big Data Trends for 2016
  http://www.tableau.com/sites/default/files/media/top8bigdatatrends2016_final_1.pdf

1. NoSQL ↗
2. Apache Spark - more efficient than Hadoop, the largest big data open source project
3. Applying Hadoop to production
4. Hadoop becomes a standard component of Big Data architectures
5. Fast data exploration capabilities and seeing Big Data as OLAP cubes
6. Self-service data preparation tools
7. Massively Parallel Processing Data Warehouse (in a cloud)
8. IoT ⇨ PB of data in a Cloud

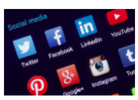
Landscape: past

- Data models
  - relational
  - object-oriented
  - semi-structured
  - ...
- Data formats
  - numbers, dates, strings
  - ...
- Veolocity
  - OLTP systems

Landscape: today







- Data models
  - relational
  - graphs
  - NoSQL
  - semi-structured
  - unstructured
  - ...
- Data formats
  - numbers, dates, strings
  - HTML, XML, JSON
  - time series and sequences
  - texts
  - multimedia
  - ...
- Veolocity
  - frequently changing (e.g., Facebook)
  - constantly changing (streams)

Needs: today

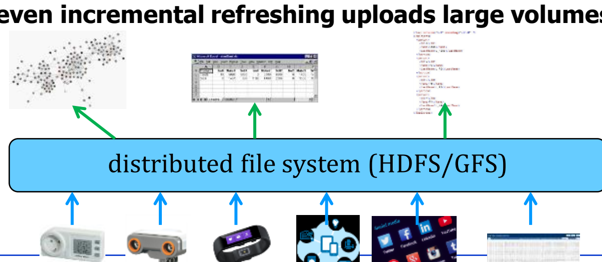
- Storing efficiently (fast writes, compression)
- Retrieving efficiently (fast scans, fast search)
- Integrating for analysis

Data integration: past

- Virtual integration
  - federated
  - mediated
- Physical integration
  - ETL + data warehouse
- Common integration data model
  - relational
  - object-oriented

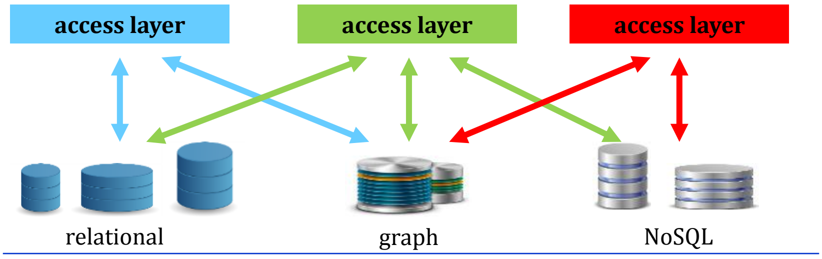
Big Data integration

- Physical integration → data lake
  - large repository of heterogeneous data (in multiple data models/formats)
  - no schema on write - schema on read
  - typically based on a distributed file system
  - need for refreshing
    - how to detect changes?
    - new algorithms for incremental refreshing?
    - even incremental refreshing uploads large volumes of data



Big Data integration

- Logical integration → analogy to mediated/federated architectures
- Polystore
  - J. Duggan, A.J. Elmore, M. Stonebreaker, et. al.: The BigDAWG Polystore System. SIGMOD Record, Vol. 44, No. 2, 2015
  - federation of islands of information
  - island of information: collection of storage engines using the same data model (query language)



Big Data integration challenges (1)

1. How to (semi)-automatically discover data sources?
   - DS structure discovery
   - DS content understanding
2. How to dynamically plug-in a DS into a federation?
3. How to construct an integrated conceptual model?
4. What integration data model to use?
5. Global query processing?
   - parsing, decomposing, translating into native dialects, and routing
6. Global query optimization?

Big Data integration challenges (2)

7. **How to integrate (transform, clean, deduplicate, integrate) on the fly data returned by local queries?**
8. **Performance optimization**
   - **caching some results**
     - **what to cache?**
     - **how to store (RAM only vs. disk)?**
   - **how to manage the cache (removing/adding data)?**
9. **New ways of querying**
   - **fusion tables**

Big Data integration challenges (3)

10. **User interface and visualization**
    - **one wants to work graphs**
    - **another wants to work with tables**
    - **multiple (different) schemas needed for mutliple users → multiple query languages?**
11. **Conceptual modeling for data warehouses**
    - **facts and dimensions in XML, Graph, NoSQL → already ongoing research**

Programming Languages

- Top Languages for analytics, data mining, data science
- Sept 2013, source: Data Science Central
- http://www.datasciencecentral.com/profiles/blogs/top-languages-for-analytics-data-mining-data-science
- The most popular languages continue to be
  - R (61%)
  - Python (39%)
  - SQL (37%)
  - SAS (20%)

Programming Languages

- Growth from 2012 to 2013
  - Pig Latin/Hive/other Hadoop-based languages ⬀ 19%
  - R ⬀ 16%
  - SQL ⬀ 14% (the result of increasing number of SQL interfaces to Hadoop and other Big Data systems?)
- Decline from 2012 to 2013
  - Lisp/Clojure ⬂ 77%
  - Perl ⬂ 50%
  - Ruby ⬂ 41%
  - C/C++ ⬂ 35%
  - Unix shell/awk/sed ⬂ 25%
  - Java ⬂ 22%

Top 10 Data Science Skills

➲ **Data Science Report. 2016, Crowd Flower**

Skills	Job skill appears in	% of jobs with skill
SQL	1987	56%
Hadoop	1713	49%
Python	1367	39%
Java	1287	36%
R	1120	32%
Hive	1099	31%
Mapreduce	768	22%
NoSQL	657	18%
Pig	561	16%
SAS	560	16%