YORK UNIVERSITÉ UNIVERSITY

ASRL
Adaptive Systems Research Lab

Big Data Platforms

Marin Litoiu
Department of Electrical Engineering and Computer Science
York University
mlitoiu@yorku.ca
http://www.ceraslabs.com

Summary so far: Clouds – Everything as a Service

- **The computing infrastructure is programmable, as a service**
  - Publish, discover, bind
  - Tune, migrate, dispose, etc.
- **The network is programmable**
- **The COST is first class citizen**
- **Cloud is not a centralized data centre**
  - Has “edges” and cores
  - Locality is important for applications and cloud

Discussion….

- **Development in cloud**
- **Migration to cloud**
- **Performance**
- **Cost**
- **Security**
- **Privacy**

Big Data and Cloud

- Ultra Large Applications
- Big Data Analytics
- Cloud File systems
- Hbase and BigTable
- MapReduce (Hadoop and Spark)

Big Data and Ultra Large Applications

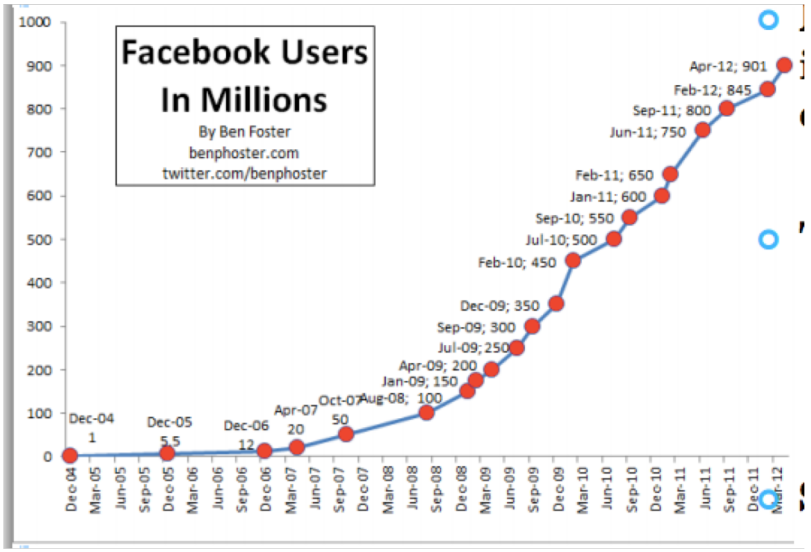
- **Big data**
  - Volume: tera, petabytes
  - Velocity: data comes at high speed
  - Variety: structured and unstructured
  - Traditional RDBMS and infrastructure cannot handle it
- **Google:**
  - from *googol*, 10^{100},emphasizes the scale of Internet
  - launched in 1998, as a search engine
  - 88 billion queries a month (2010)
  - 0.2 second/query
- **Facebook:**
  - Over 1 billion users
  - Scaled very fast

The Scalability Problem

- **Google:**
  - More data
    - Digitized libraries, movies, museums
  - More queries
    - More people and programs use Google applications
  - Better (more accurate) results
  - Ex: to crawl the web and pull 20 billion web pages (or 400 terabytes)
    - 1 computer takes 4 months
    - 1000 computers take less than 3 hours
    - There is more to index, handle queries, etc…
  - Store and process tera, petabytes of data

The Scalability Problem

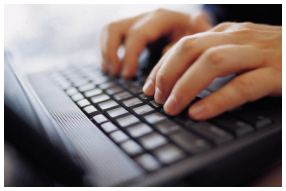
- **Facebook**
  - Billions of users
  - Massive amounts of real time data and processing
  - Scalability issues
  - Economic models based on the data analytics

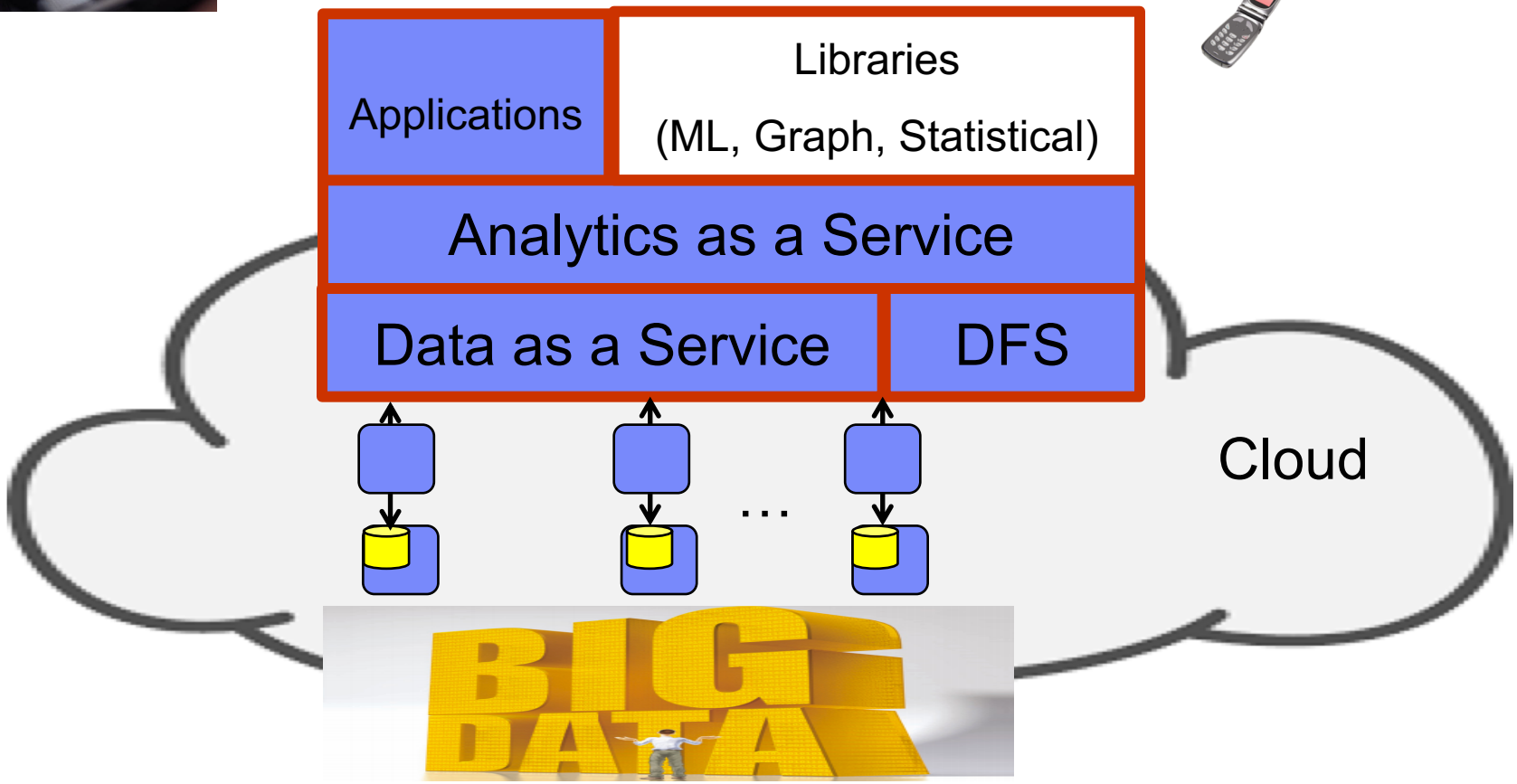


Business Pull: Big Data Analytics

- **Data Analytics is the process of loading, cleaning, modeling, transforming of data in order to gain new knowledge for better decision making**

Big Data Analytics Platform

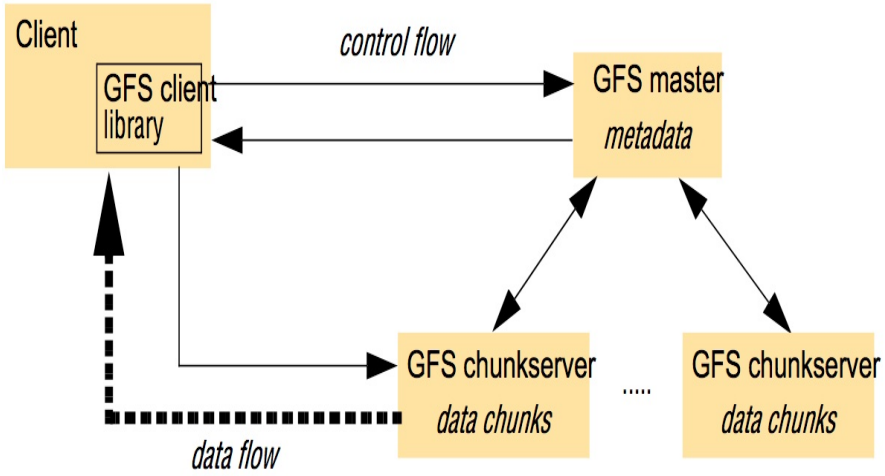




Cloud File Systems (Google and Hadoop File Systems)

https://hadoop.apache.org/

- **A file is divided in many chunks (64GB) each**
- **Chunks resides on different computers/VMs**
- **A Master keeps track of all chunks**
  - As a user you can create, delete, append to a file
- **This file system is optimized for application and data**
  - Large files
  - High scalability
  - Changing is mostly appending
  - Many parallel reads
  - Many parallel writes



Big Data As a Service

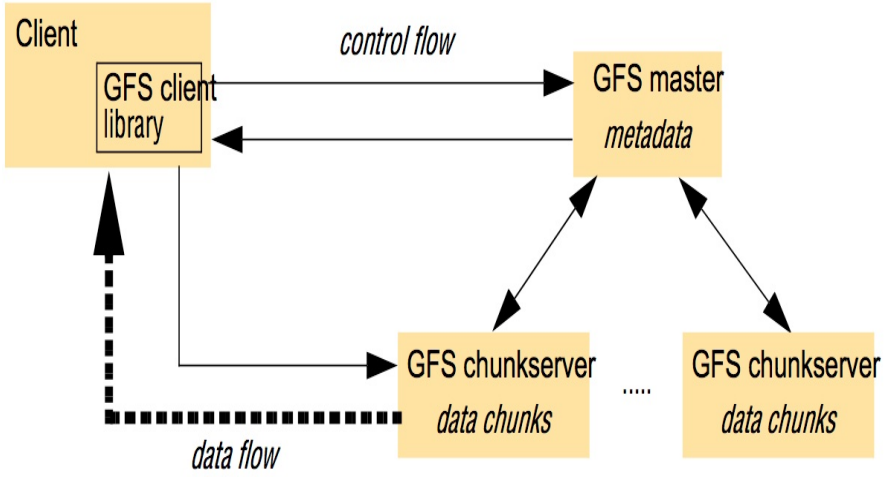
- **"Big" Databases provided on demand**
  - in a customizable format (size, structure)
  - with some quality of service (throughput, latency)
  - eventual consistency
  - highly scalable: A Big Table is split into many tablets residing on many VMs
- **Price per in/out data, storage, CPU, etc…**
- **A small set of operations (put, get, scan…)…NoSQL**
- **Plenty of offerings to choose from: Hbase, Cassandra, MongoDB..**

Which Big Data as a Service to use?

Autoscaling Hadoop File System (your turn)

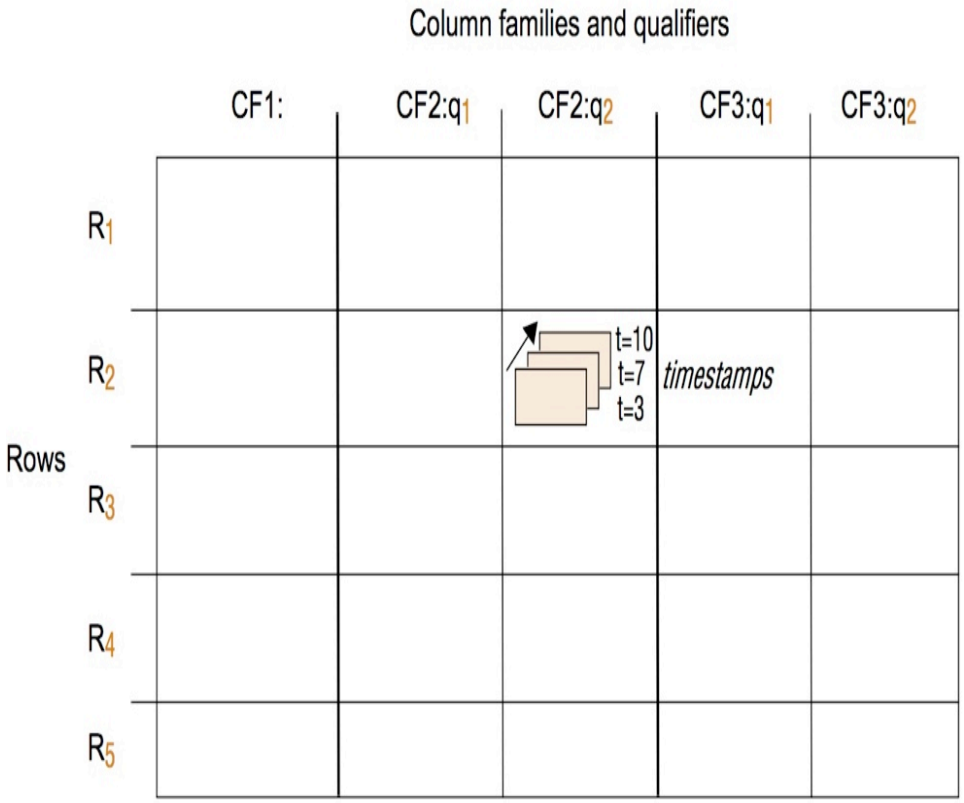
- **Use Cases**

- **Monitor?**
- **Analysis and planning?**
- **Execution?**
- **Challenges?**



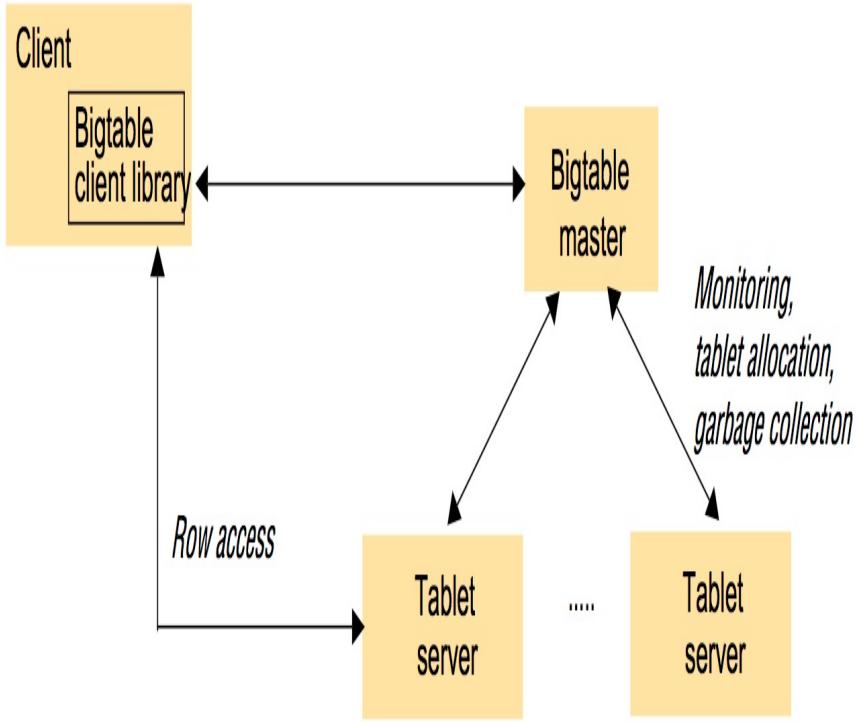
BigTable, Hbase(https://hbase.apache.org/)

- **It is not a relational database but preserves the concept of table**
- **Each row has an associated row key (up to 64KB)**
  - It is common that the row keys are URLs (such as www.bbc.co.uk/sports)
  - Adjacent keys are related (easy to process); to enable that, the domain is in reverse order
    - Ex:R1: uk.co.bbc/sports; R2: uk.co.bbc/news
- **Columns are grouped per families and the names have the form *family:qualifier***
- **A given cell of the table can have many versions, distinguished by the time stamp**
- **To scale up, a big table is split in many smaller tablets (see next slide)**



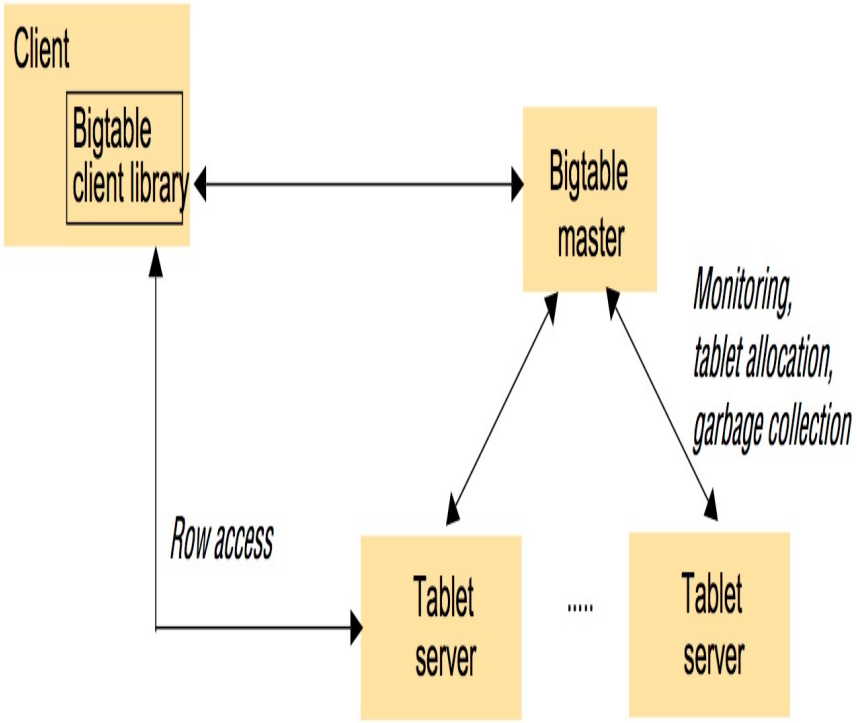
Big Data Database Architecture

- **A table is divided in many Tablets; a tablet is 240GB in size**
- **As the table grows, more tablets are added**
- **Each tablet can reside on a different computer**
- **There is one Master table that keeps track of all tablets**
- **The application can work directly with the tablets**
  - Put(key, data)
  - Get(key)
  - Scan(keyrange)



Autoscaling Hbase (your turn)

- Use Cases
- Monitor?
- Analysis and planning?
- Execution?
- Challenges?



Research Challenges

- **How do we achieve good performance at low cost?**
  - How do we size the clusters (nodes)
  - How do we design the schema
  - How do we scale (adaptively)

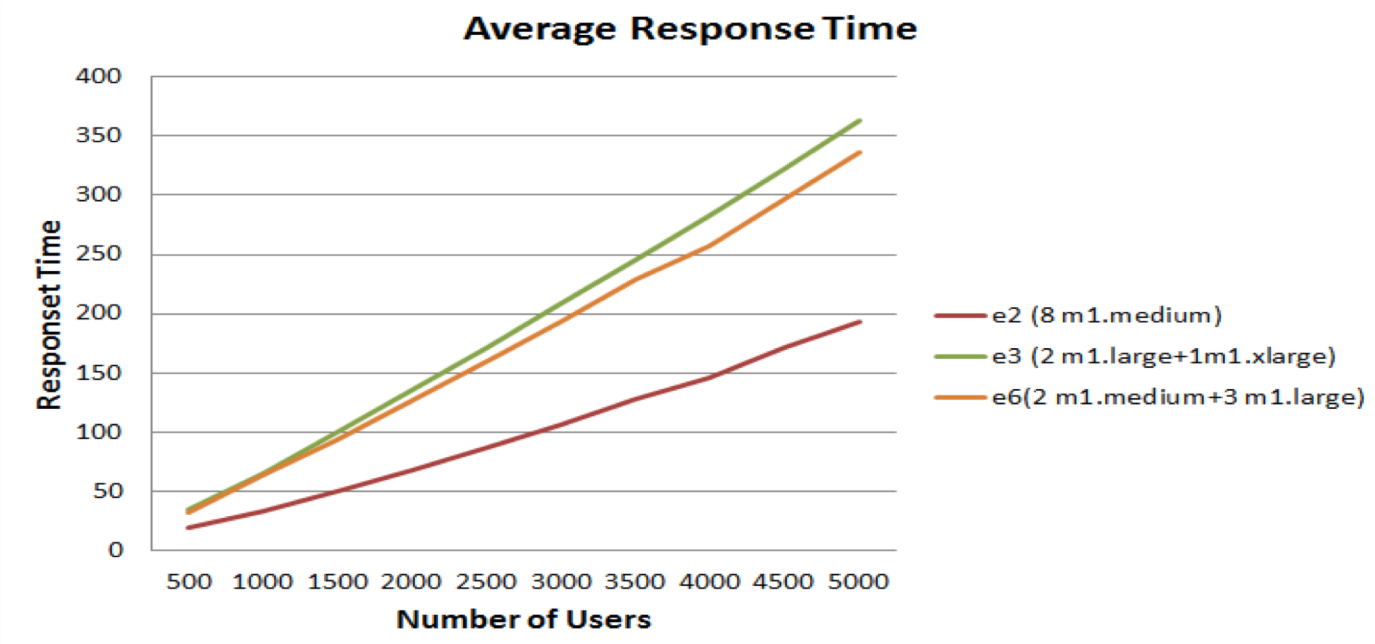
Connected Vehicles and Smart Transportation (CVST) http://cvst.ca

- **Understand and model the traffic in GTA**
- **A Big Data Analytics Platform should support**
  - Users
    - Traveler, Traffic Engineer, Planner, Researcher, etc..
  - Queries
    - Spatiotemporal, Graph, Statistical, etc
  - Workloads
    - Batch, Interactive
  - Data types: structured, semi-structured and unstructured
    - Loop detectors, Traffic cameras, Mobile devices, Social media etc

R1. An Experiment on CVST

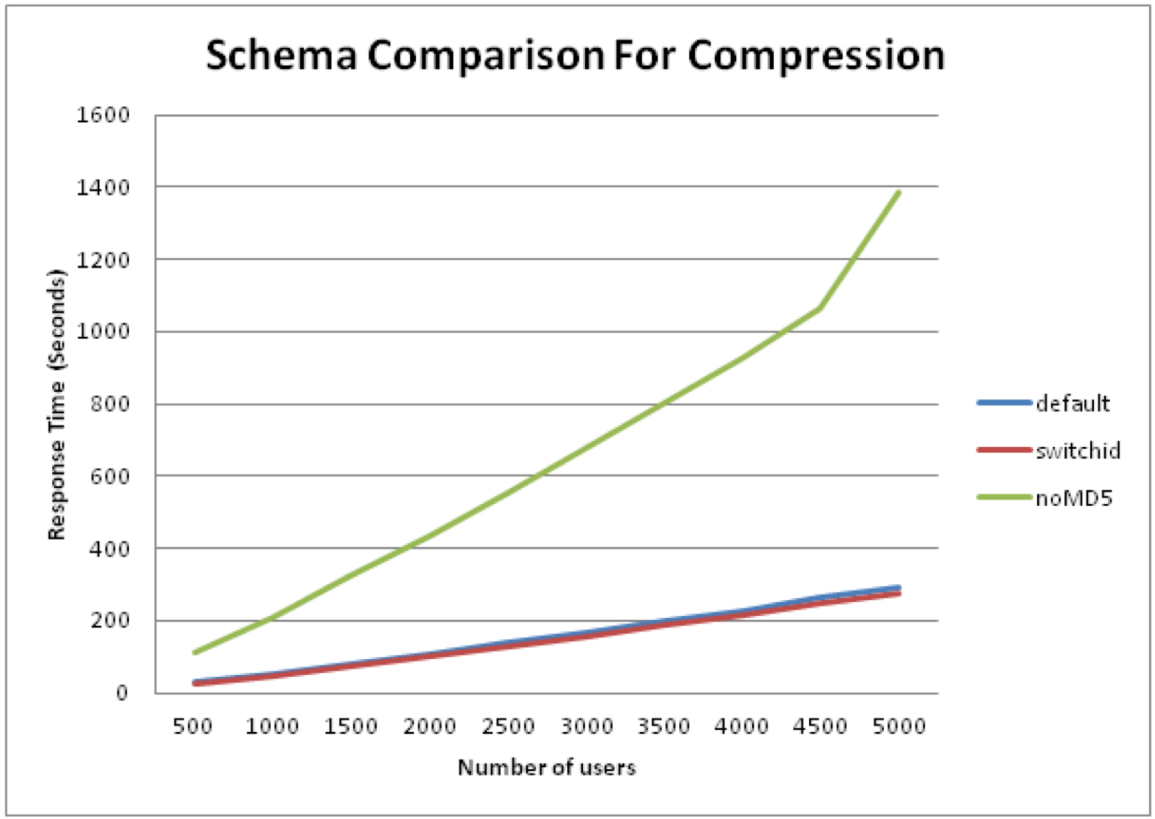
- **Three Months of Real-Time Traffic Data from GTA**
- **Hbase deployed on a reconfigurable cluster**
- **Workload:**
  - Average volume for a random day for a given sensor on GTA highways
  - “Scan” Queries

The Flavor of the VM matters…

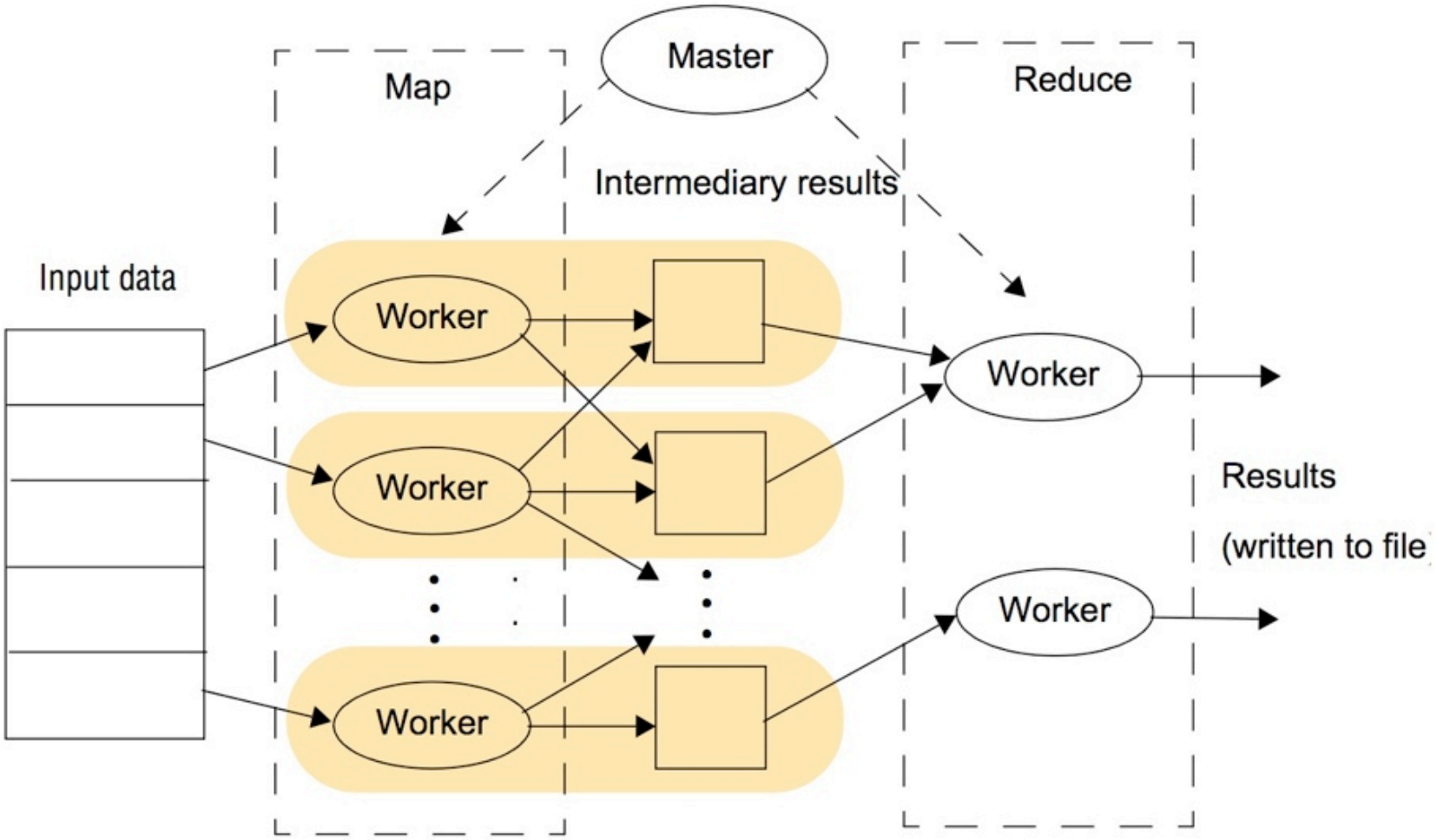


Clusters have the same “price” and capacity
..but different VM flavors
Better to have smaller but more VMs

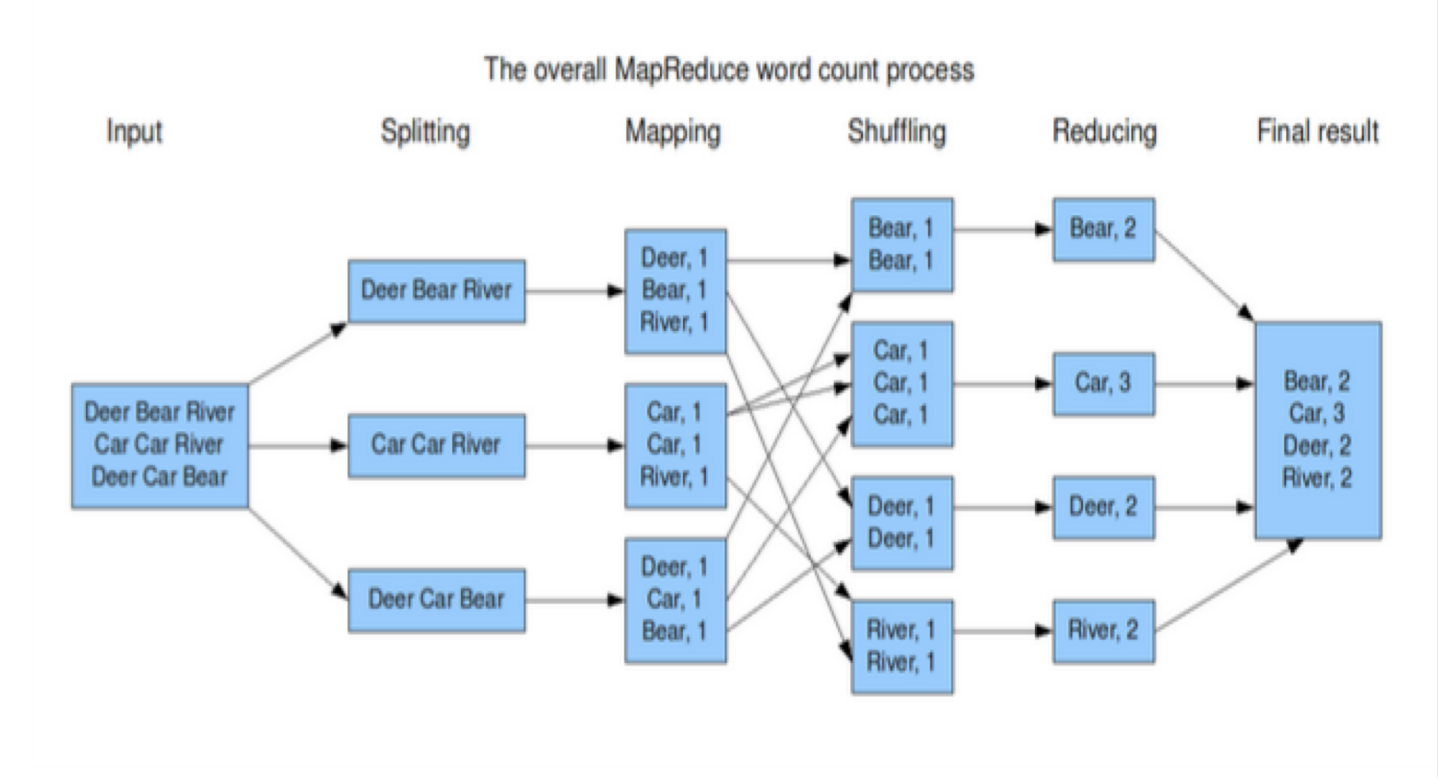
Choosing the Key/Schema



MapReduce: a New Parallel Processing Model
https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html



Example: Word Count



Programming Model in Spark (https://spark.apache.org/docs/latest/quick-start.html)

- **Based on the concept of RDD (Resilient Distributed Dataset)**
- **Split your Big Data in slices, RDD sets, which are going to be processed in parallel**
- **The number of slices is 2-4 times the number of CPUs in your Spark Cluster**
- **Apply Transformations (e.g Map, Shuffle)**
- **Apply Actions (e.g. Reduce)**

Example, Counting Words

```
//splitting data into lines
JavaRDD<String> lines = sc.textFile("hdfs://log.txt");
//Map: separates the words…the result is a list of words
JavaRDD<String> words =
    lines.flatMap(line -> Arrays.asList(line.split(" ")));
//sufle and reduce ( counts the identical words)
JavaPairRDD<String, Integer> counts =
    words.mapToPair(w -> new Tuple2<String, Integer>(w, 1))
        .reduceByKey((x, y) -> x + y);
counts.saveAsTextFile("hdfs://counts.txt");
```

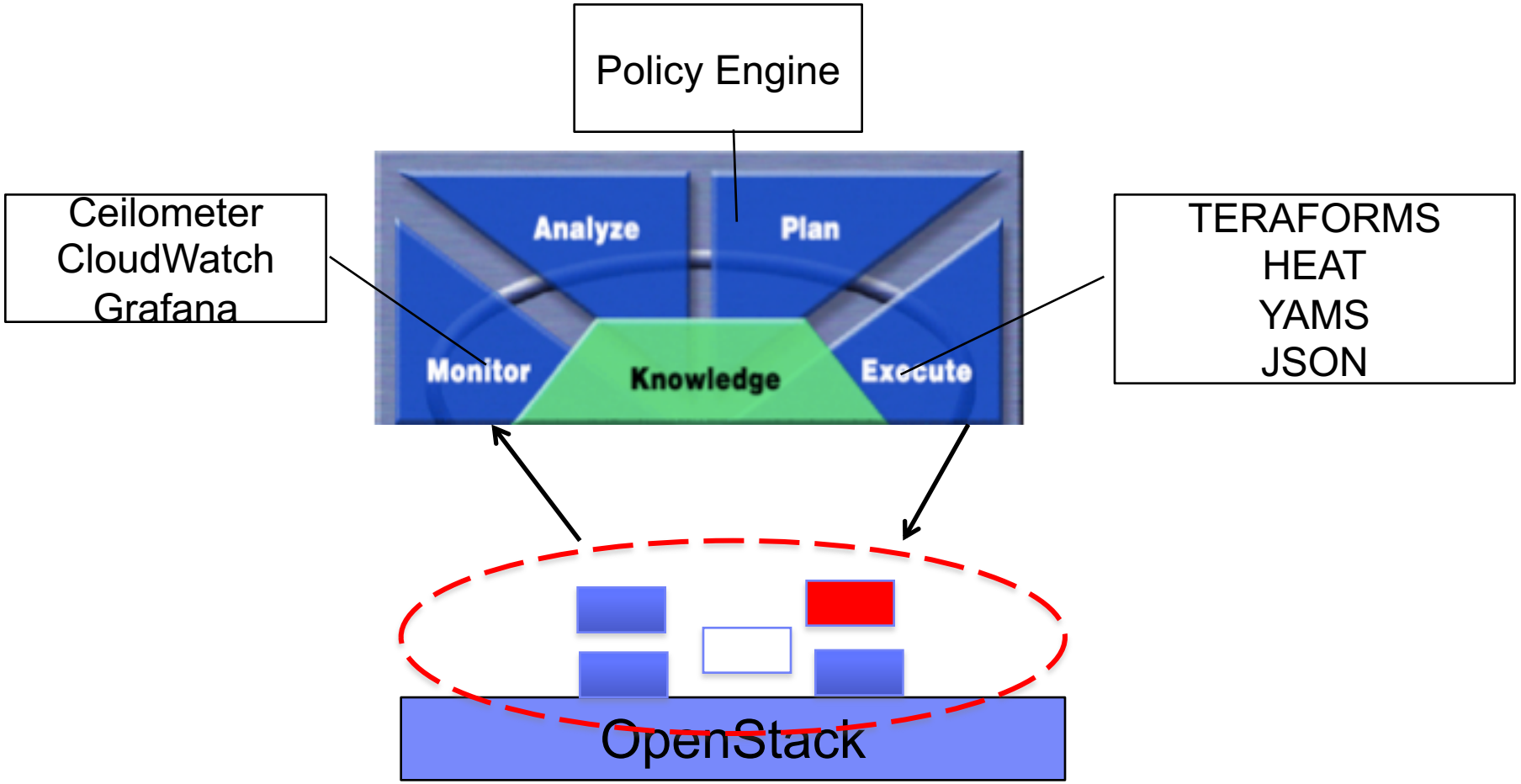
Discussion…

- **Business pulls: what drives big data?**
- **Technology push: what enables big data?**
- **Data lock-in**
- **Cost**
- **Design for Performance and Scalability**
- **Privacy**

Summary

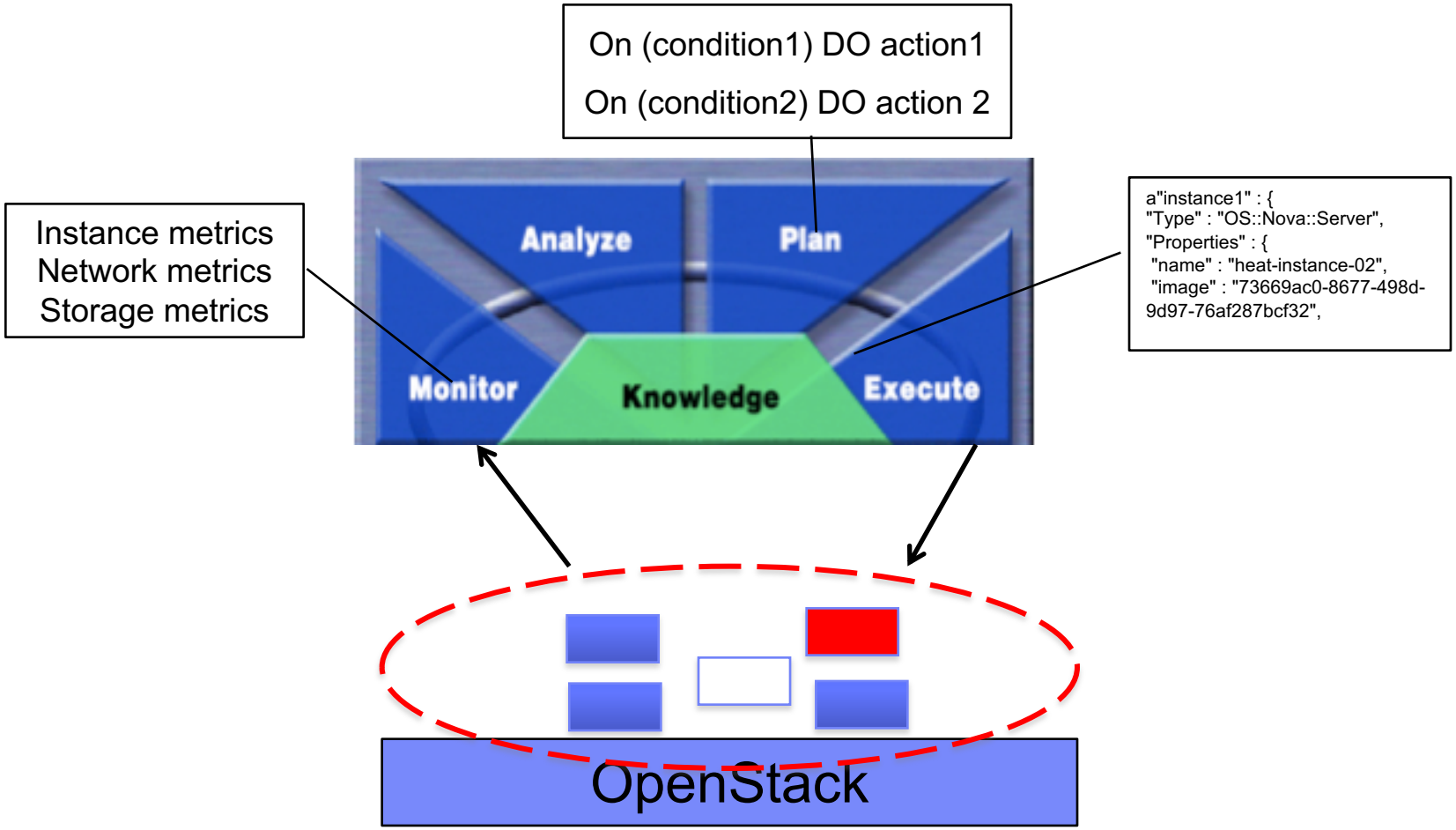
- **Big data is pushed by business analytics and by ultra large applications**
  - Volume, velocity, variety
- **Based on**
  - Cloud File Systems (GFS, HDFS)
  - Big Databases, NoSQL (Hbase, BigTable, Cassandra..)
  - Analytics platforms (Hadoop, Sahara)
  - MapReduce

Adaptive Applications on Clouds



Cloud software provides initial building blocks…not the whole feedback loop

Adaptive Applications in Public Clouds



Your turn

- **Search the web for the definition of those terms, used in cloud computing monitoring and execution**
  - YAML
  - HEAT
  - Terraforms
  - json
  - Ceilometer
  - Cloudwatch
  - Grafana
  - drools

More readings

- **Templates (Planing/Execution)**
  - https://link.springer.com/content/pdf/10.1186%2Fs13677-014-0007-3.pdf