Big Data

A general approach to process multimedia datasets

David Mera

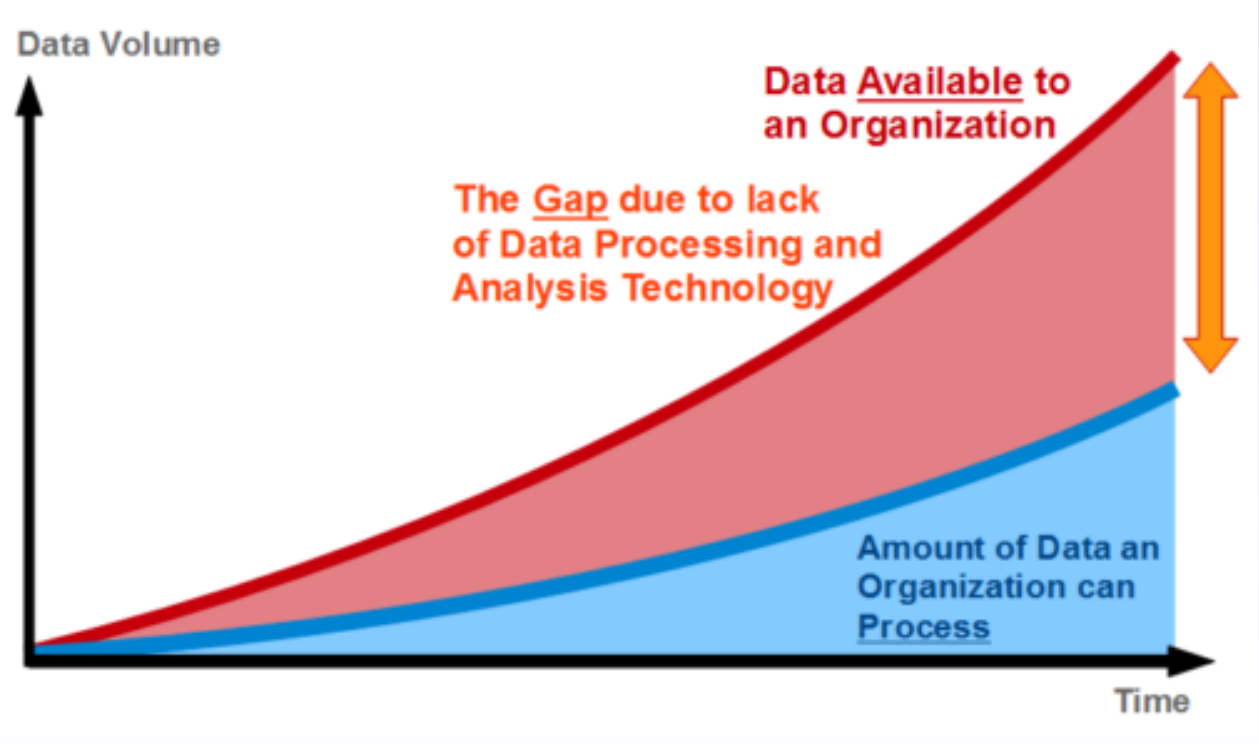
Masaryk University
Brno, Czech Republic

24/02/2014

Table of Contents

Table of Contents

- Organizations have potential access to huge datasets of heterogeneous data.
- Stored data are usually not structured.
- Data should be processed to uncover useful information.



Batch data

- Static snapshot of a dataset
- Batch computation has a ‘start’ and an ‘end’
- Fast datasets processing

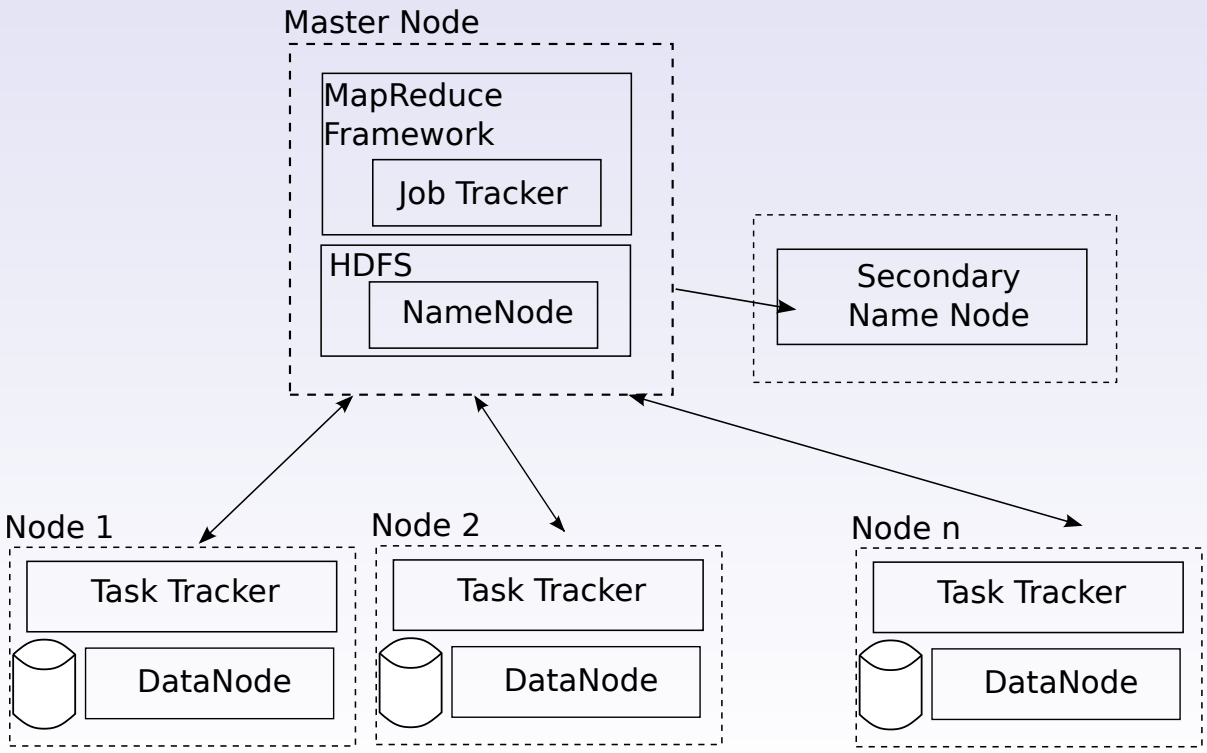
Stream data

- Stream of events that flows into the system at a given data rate over which we have no control
- Stream computation ‘never’ ends
- The processing system must keep up with the event rate or degrade gracefully
- Near-real time answers

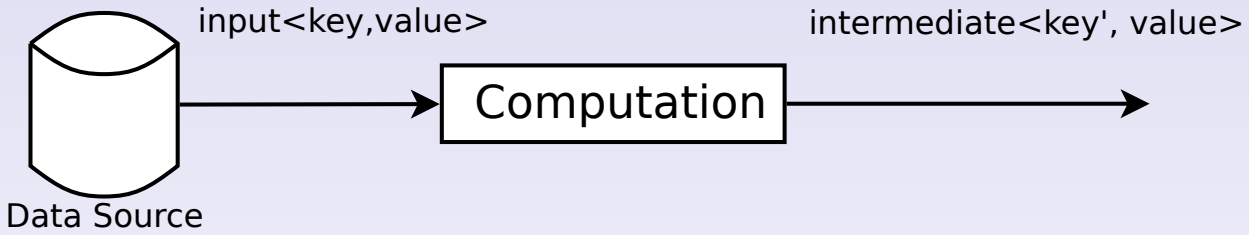
Table of Contents

MapReduce

- MapReduce is a framework for paralleling the processing of massive datasets.
- The Hadoop implementation is highly optimized for batch processing
- Hadoop attempts to run Map and Reduce tasks at the machines were data being processed are located



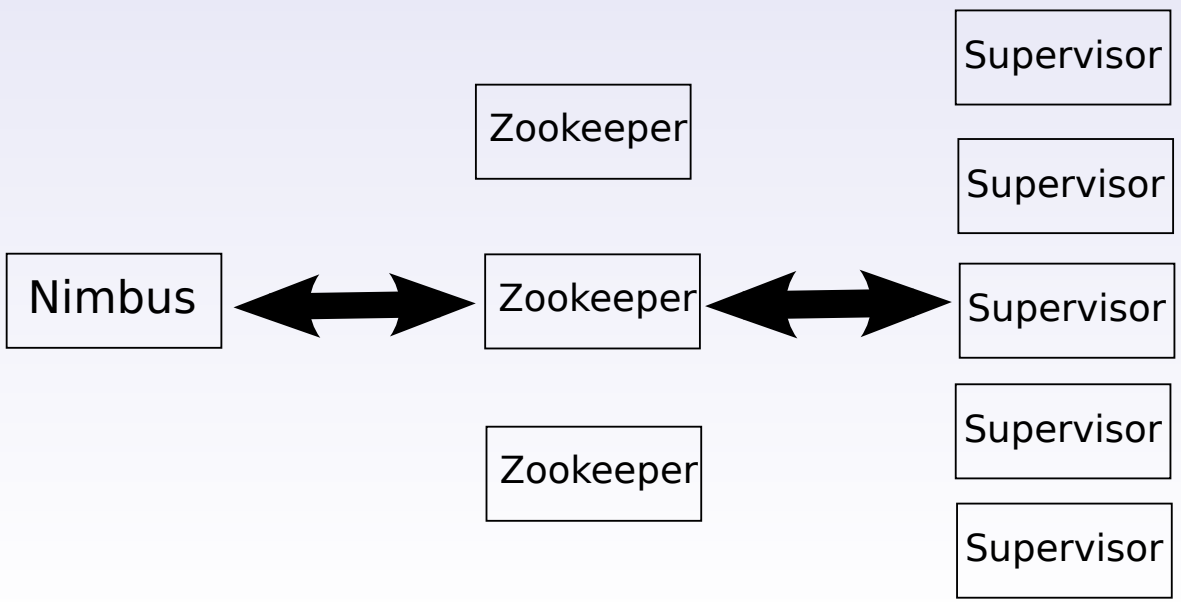
- MapReduce Job
  - Map function (mandatory)



  - Reduce function (optional)

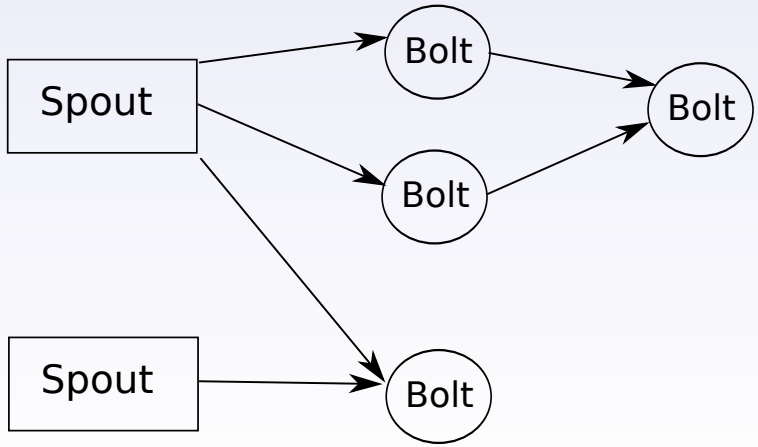
Storm
Distributed and fault-tolerant realtime computation

- Storm cluster
  - Master node
    - The Nimbus daemon is responsible for distributing code around the cluster, assigning tasks to machines, and monitoring for failures
  - Worker nodes
    - The Supervisor daemon listens for work assigned to its machine and starts and stops worker processes as necessary based on what Nimbus has assigned to it.
  - Communication - Zookeeper



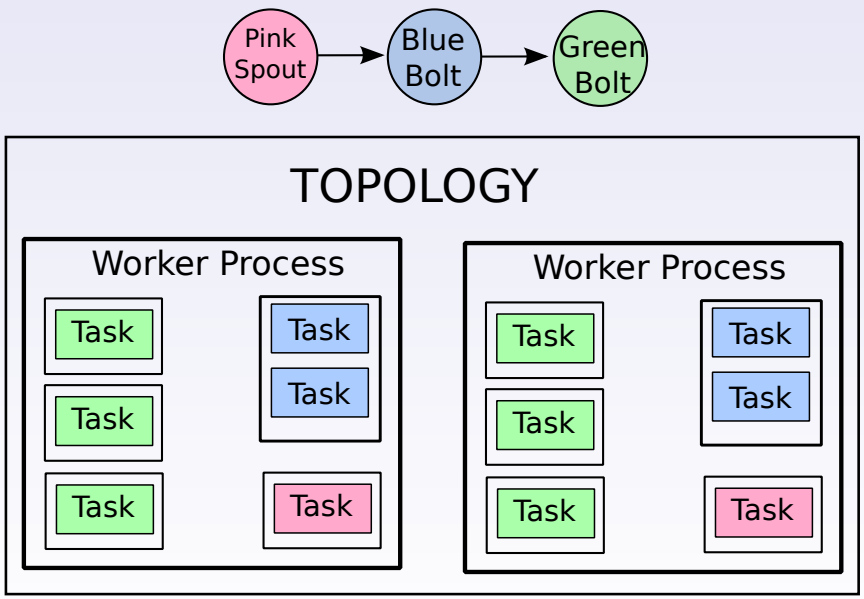
Storm
Components

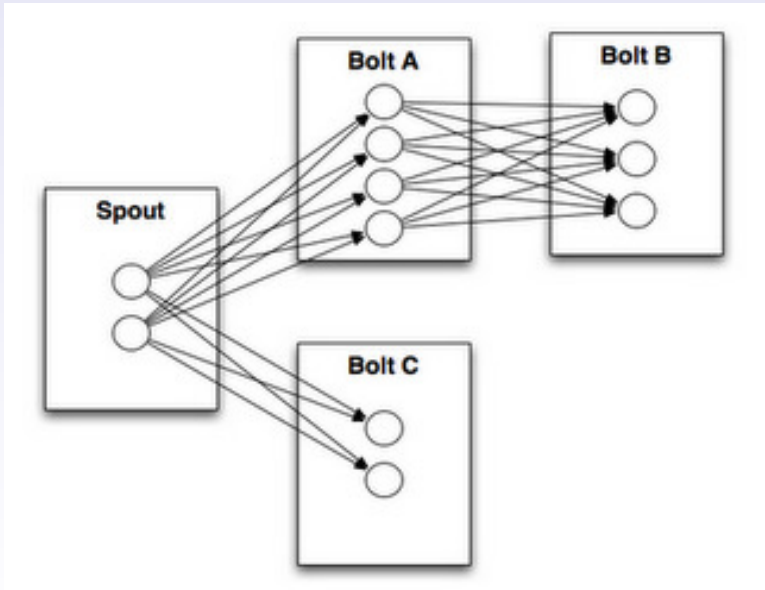
- Storm runs topologies
  - Graph of computation
  - Each node in a topology contains processing logic
- Stream
  - Unbounded sequence of tuples
- Spout
  - It reads input data from an external source and emits them as a stream
  - It is capable of replaying a tuple
- Bolt
  - Input streams –> some processing –> new streams.



Storm
Parallelism of a Storm topology

- Topologies execute across worker processes (JVM)
- Tasks are spread evenly across all the workers
- The parallelism for each node is defined by the user
- User can also specify tasks for each node
- Stream grouping - How a stream should be partitioned
  - i.e.Shuffle grouping
- Scalability in processing time



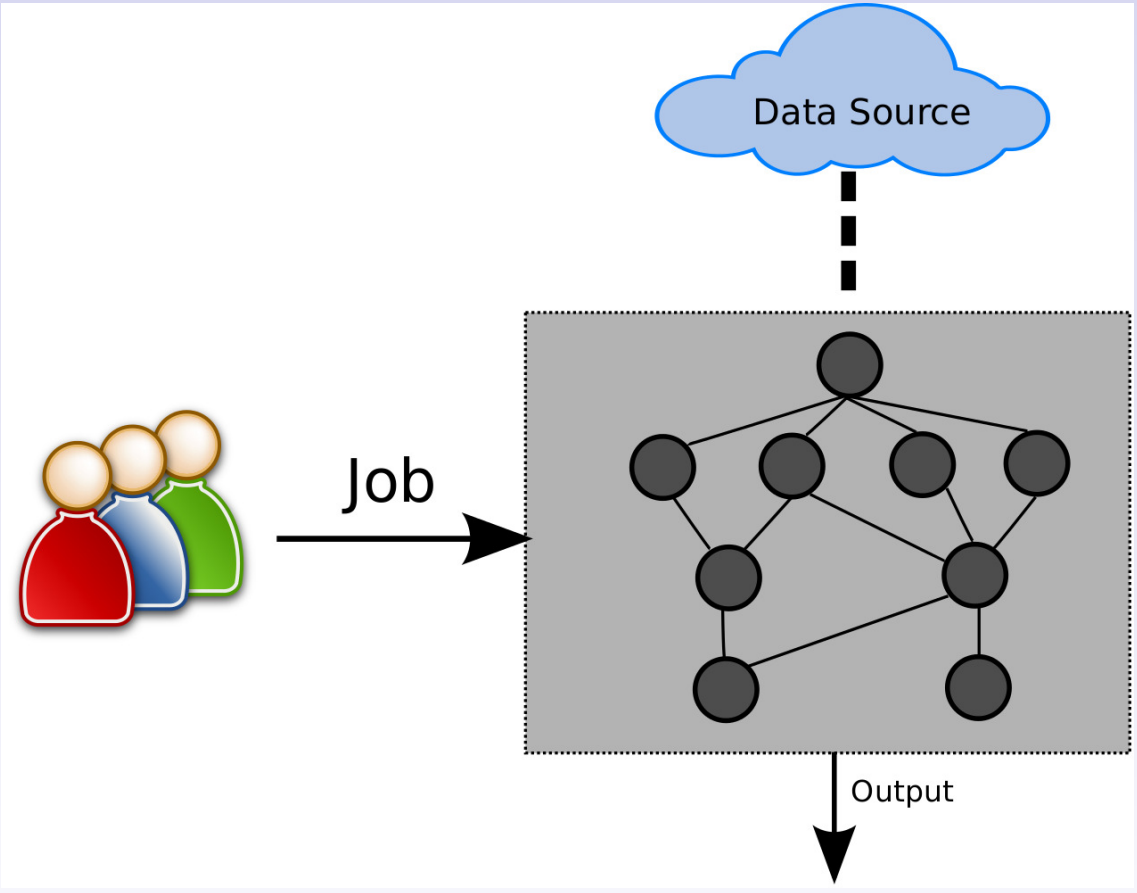


Previous conclusions

“Attempting to build a general-purpose platform for both batch and stream computing would result in a highly complex system that may end up not being optimal for either task”

Table of Contents

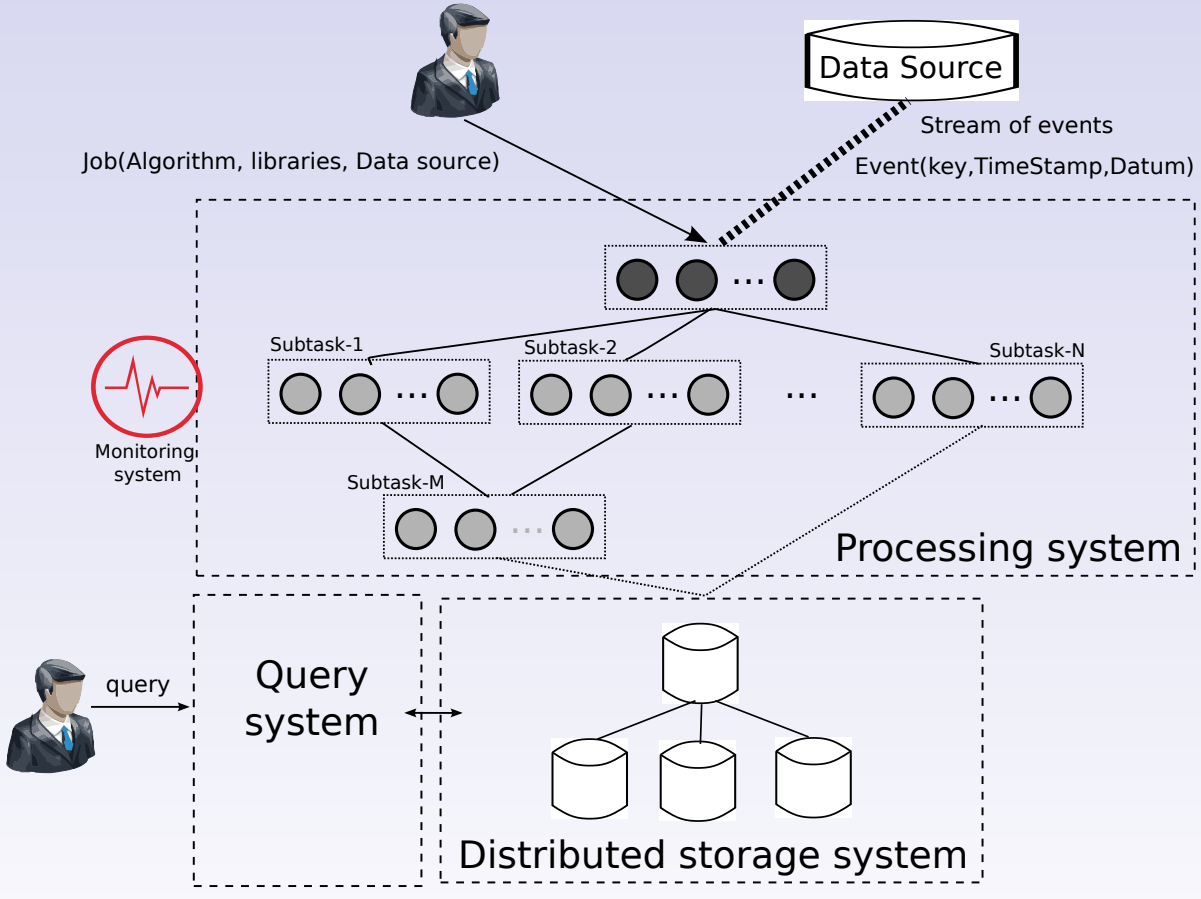
System development
Highest system overview



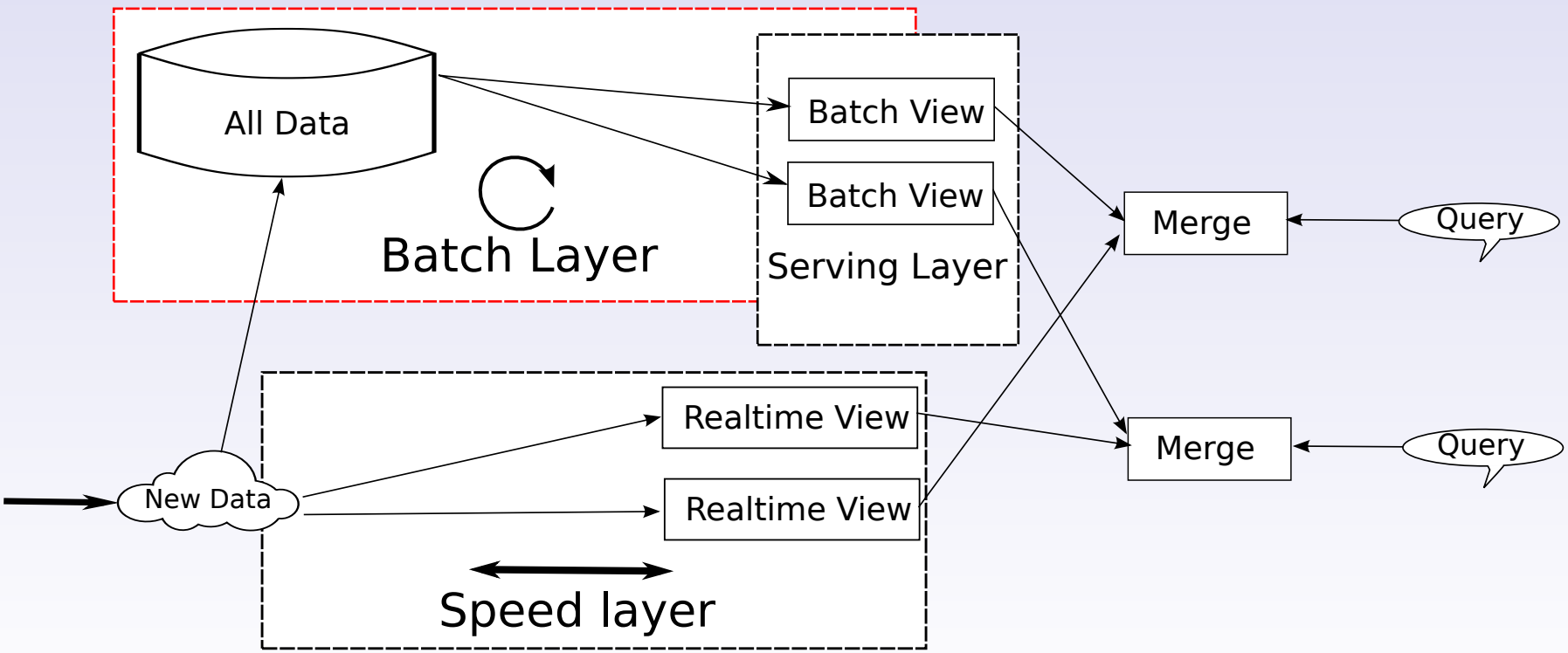
System development
Goals

- Efficient processing of huge datasets
- External and internal data access
- Heterogeneous data management
- Processing of arbitrary functions
- Data relations management
- Infrastructure flexibility

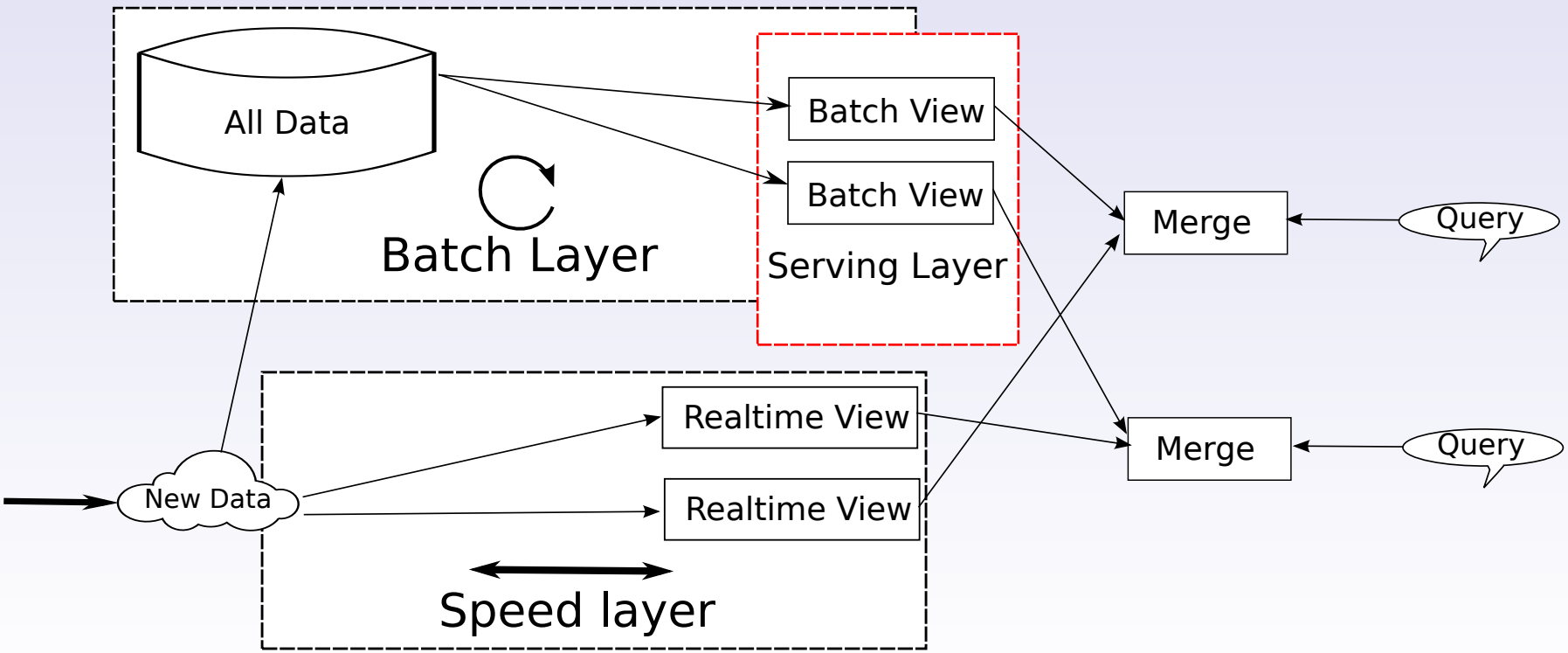
System development
System overview



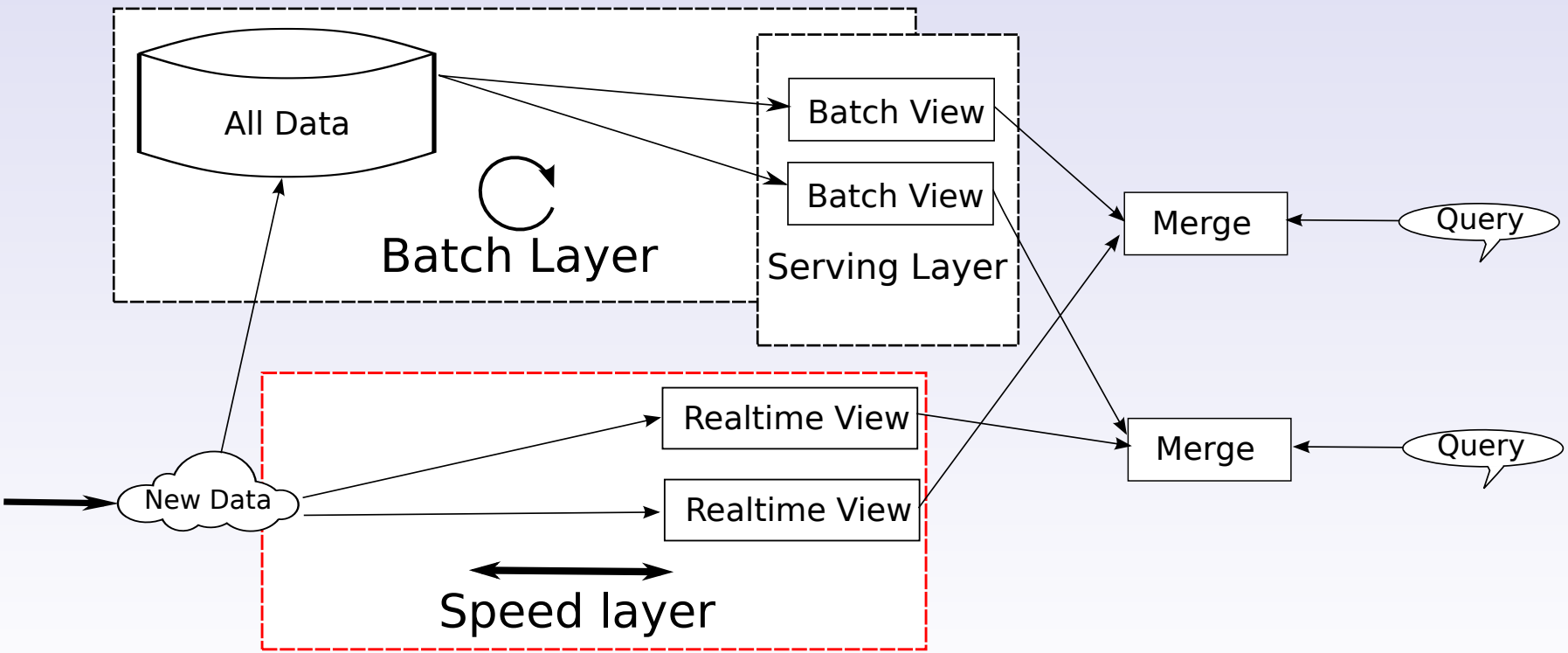
- Batch layer
  - Storage of the master dataset
  - Batch views computation



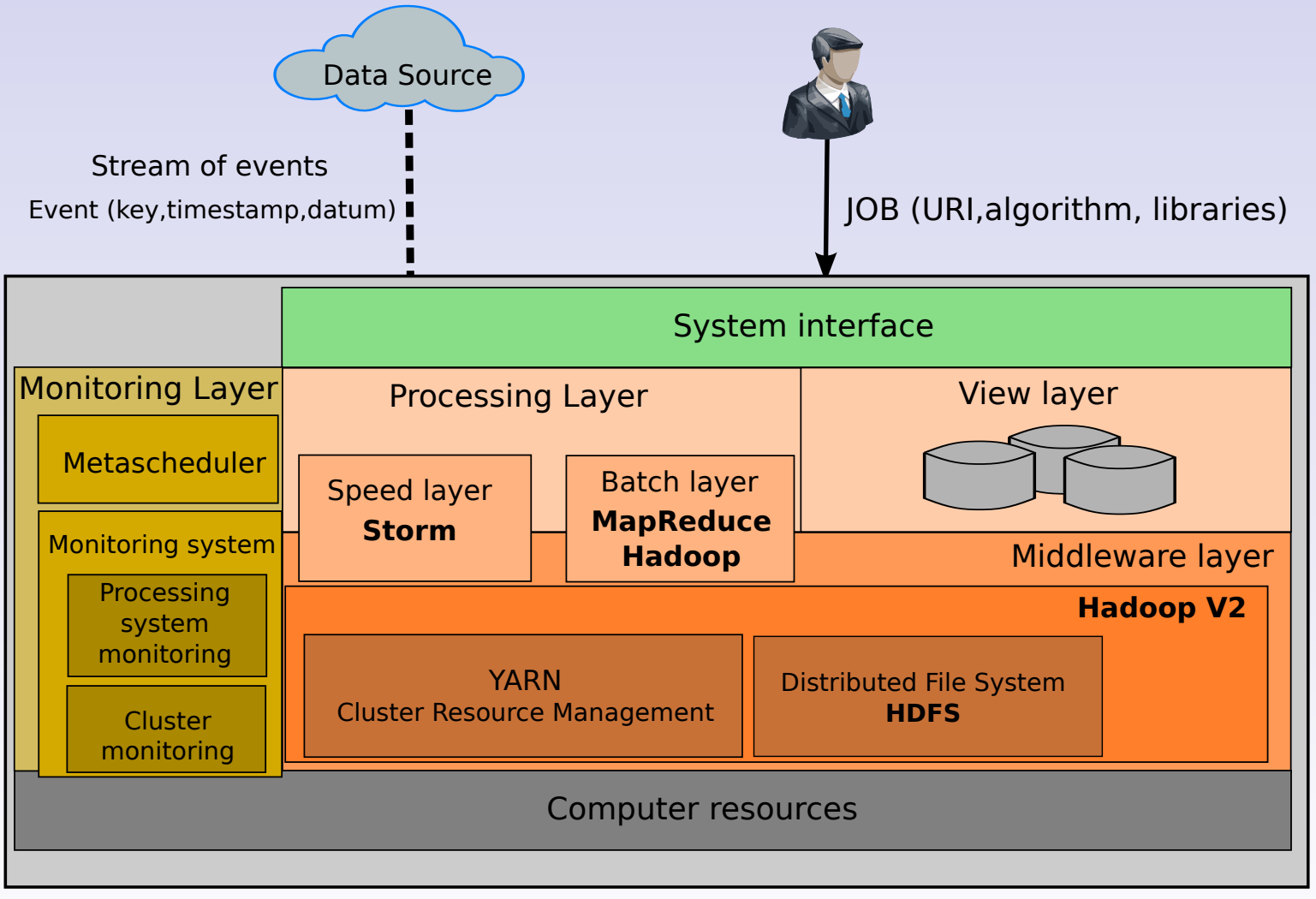
- Serving layer
  - Batch views storage
  - Efficient query system
  - The views are updated whenever the batch layer finishes precomputing a batch view



- Speed layer
  - Realtime processing of arbitrary functions on arbitrary data
  - Real time views computation via incremental updates



System development
General overview of the architecture

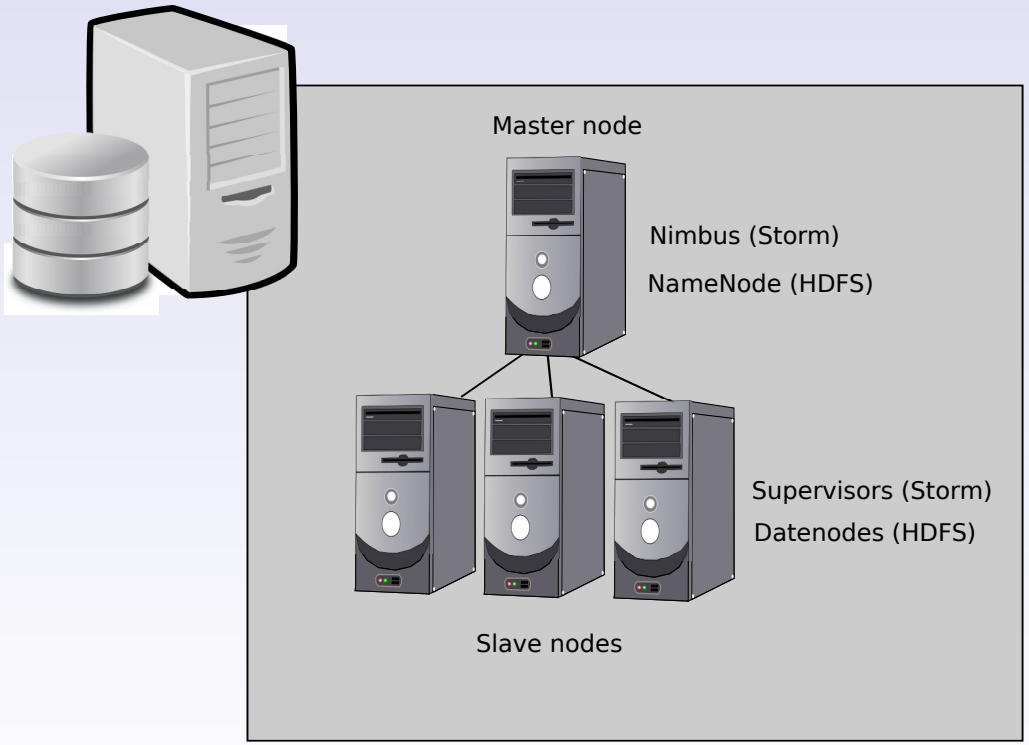


- Data source access
  - URIs (Uniform Resource Identifier)
  - External data: Speed Layer (virtual streams)
  - Internal data: Batch Layer
- Data management
  - Heterogeneous data
  - Data storage
- Data relations
  - Timestamp
  - Specialized Storm topologies
- Processing arbitrary functions
  - Meta-language
  - Scheduler

- Infraestructure flexibility
  - Dedicated hardware infrastructure: it is expensive and very often it is wasted
  - Shared infraestructure: processing systems are not usually adapted.
- Main Challenges
  - Monitoring system to analyze the status of the cluster and jobs
  - Metascheduler to automatically modify the use of the infraestructure according to the monitoring system

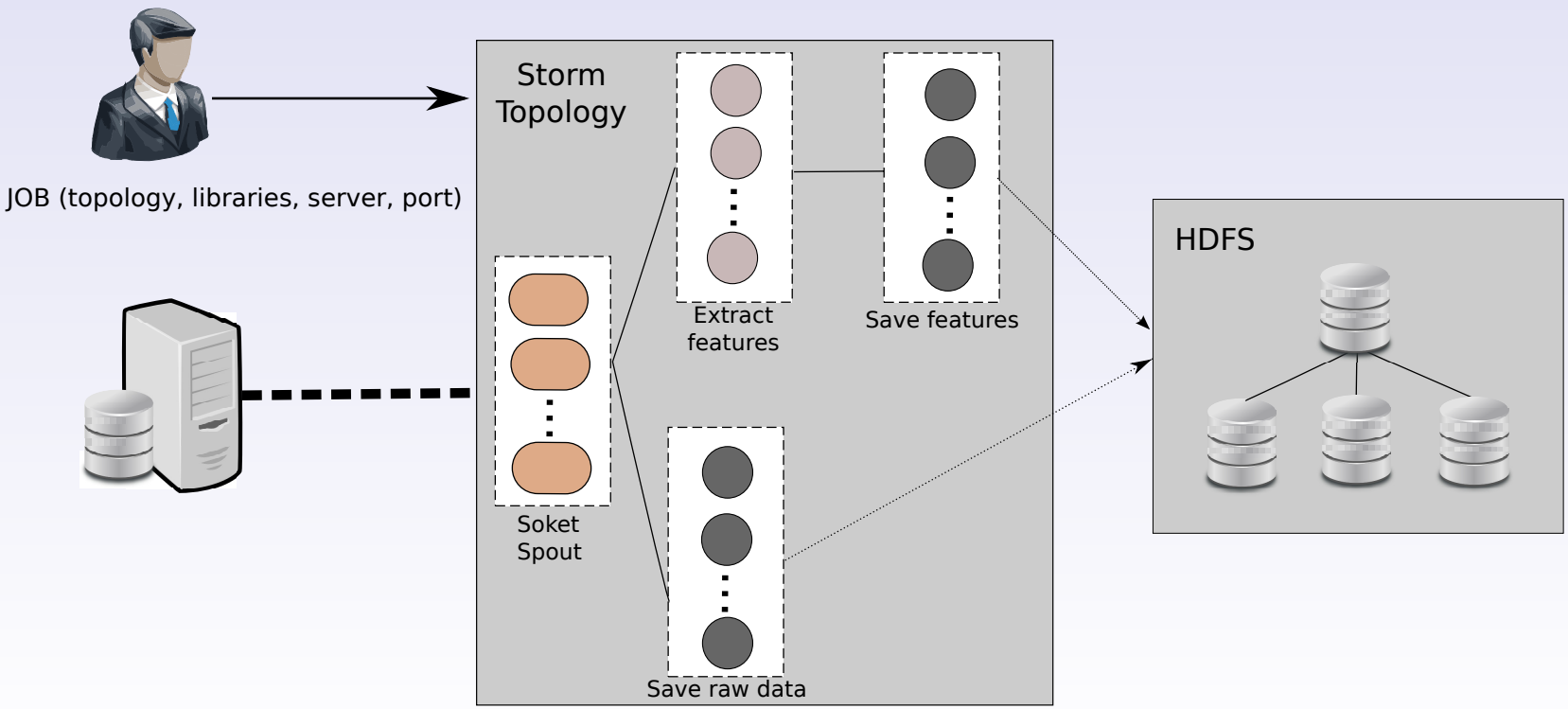
System development
Prototype

- A virtual cluster via Virtual Box was deployed
- Hadoop and HDFS were installed (batch layer)
- Storm was installed (speed layer)



System development
Prototype

- A Spout to get external data was deployed
- Data containers were developed
- A generic bolt was designed to store data
  - Specific implementation to deal with HDFS
  - Bolt takes into account the block size



Ongoing work

- Deployment of the prototype in a real cluster
- Comparative study between the prototype and other processing approaches
  - Sequential computing
  - Grid computing
  - MapReduce
- Infrastructure flexibility
  - Storm flexibility
  - Monitoring system development
    - Cluster status
    - Job status
  - Metascheduler development

Big Data

A general approach to process multimedia datasets

Thank you for your attention!