Big Data

A general approach to process external multimedia datasets

David Mera

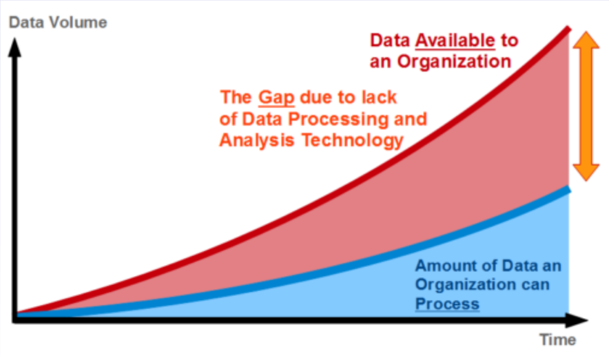
Laboratory of Data Intensive Systems and Applications (DISA)
Masaryk University
Brno, Czech Republic

7/10/2014

Table of Contents

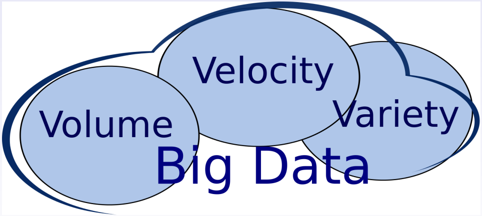
Table of Contents

- Huge new datasets are constantly created.
- “90% of the data in the world today has been created in the last two years", 2013 [1]
- Organizations have potential access to a wealth of information, but they do not know how to get value out of it

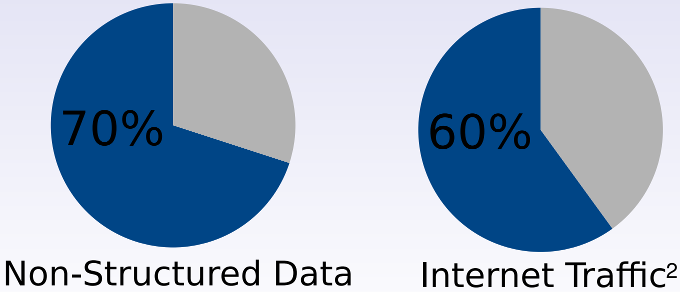


[1] Source: SINTEF. “Big Data - for better or worse”

- Big Data phenomenon
  - **Volume** refers to the vast amount of data generated every second
  - **Variety** refers to the different forms of data
  - **Velocity** refers to the speed at which new data are generated
  - Veracity refers to the reliability of the data
  - Value



- Multimedia Big Data
  - 100 hours of video are uploaded to YouTube every minute
  - 350 millions of photos are uploaded every day to Facebook (2012)
  - Each day, 60 million photos are uploaded on Instagram
  - ...



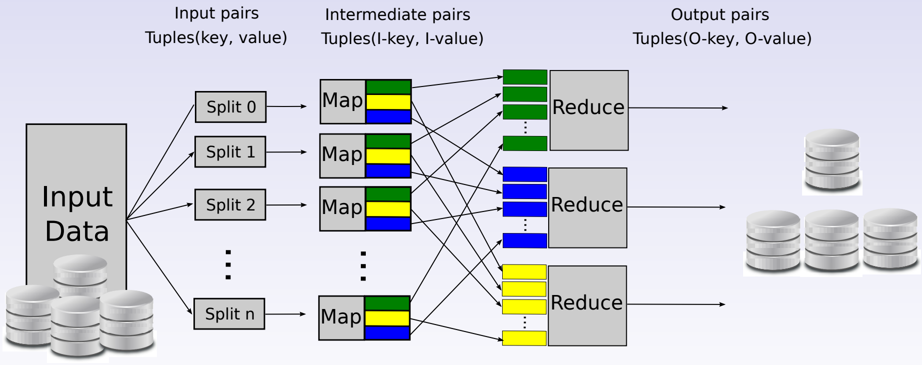
[2]Source: IBM 2013 Global Technological Outlook report

- Getting information from large volumes of multimedia data
  - Content-based retrieval techniques
  - Findability problem
    - Extraction of suitable features $\rightarrow$ Time-consuming task
- Feature extraction approaches
  - Sequential approach $\rightarrow$ not affordable
  - Distributed computing: Cluster computing, Grid computing
    - High computer skills
    - 'Ad-hoc' approaches $\rightarrow$ Low reusability.
    - Lack of handling failures
  - Distributed computing: Big data approaches
    - Batch data: Map-Reduce paradigm (Apache Hadoop)
    - Stream data: S4, Apache Storm.

Table of Contents

- Apache Hadoop characteristics (Map-Reduce paradigm)
  - Batch data processing system
  - Commodity computing
  - No specialized distributed-computing skills are required
  - Machine communication
  - Task scheduling
  - Scalability
  - Handling failures
  - Automatic partition of the input data

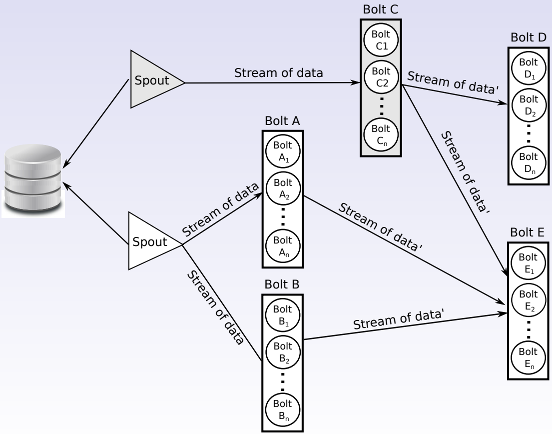
- Map-Reduce paradigm



- Weaknesses and limitations
  - Large files optimization
  - Batch data processing
  - Response time
  - Hard configuration process - iterative optimization
  - Lack of real-time processing
  - The parallelization level cannot be altered in running time

- Apache Storm characteristics
  - **Real-time processing system**
  - Commodity computing
  - No specialized distributed-computing skills are required
  - **Set of generic tools to build distributed graphs of computation**
  - Machine communication
  - Task scheduling
  - Scalability
  - Handling failures
  - **The parallelization can be adapted in processing time**

- Storm runs topologies
  - Streams: unbounded sequence of tuples
  - Spouts: source of streams
  - Bolts: input streams → some processing → new streams

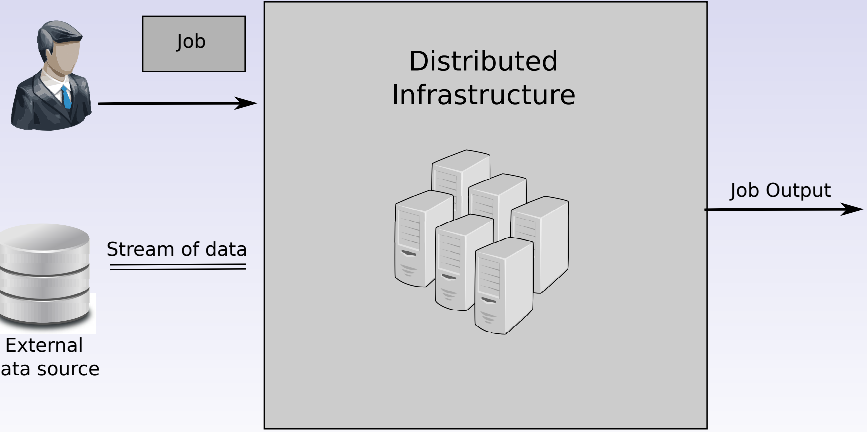


- Weaknesses and limitations
    - Lack of support for processing batch data
    - low-level framework
    - Pull mode
    - Specific scenario configurations

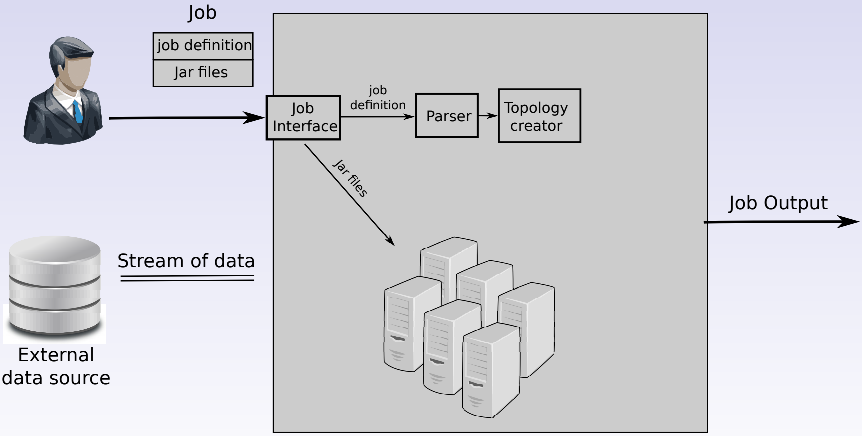
Table of Contents

- Prototype goals
  - Efficient processing of huge external datasets
  - Heterogeneous data management
  - Processing of arbitrary functions
  - Infrastructure flexibility
  - Handling failures

Prototype
General overview



Prototype
General overview



Prototype

Job definition

```
<job>
  <name>...</name>
  <datasource>...</datasource>

  <data save="bool">

    <operators>
      <operator>*

        <class>
          <name>...</name>
          <method>...</method>
        </class>

        <data save="bool">
          <operators>...</operators>
        </data>
      </operator>
    </operators>
  </data>
</Job>
```

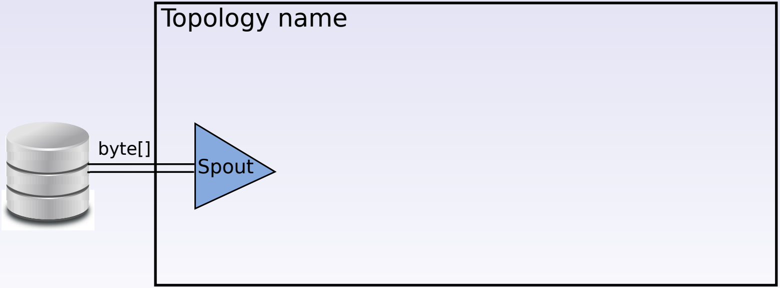
Prototype
Job definition

```
<job>
  <name>...</name> --------------------> Topology name
  <datasource>...</datasource>

 <data save="bool">

    <operators>
      <operator>*

        <class>
          <name>...</name>
          <method>...</method>
        </class>

        <data save="bool">

           <operators>...</operators>
        </data>
      </operator>
    </operators>
  </data>
</Job>
```

Prototype
Job definition

```
<job>
  <name>...</name>  → Topology name
  <datasource>...</datasource>  → Spout

  <data save="bool">

      <operators>
        <operator>*

          <class>
             <name>...</name>
             <method>...</method>
          </class>

          <data save="bool">

              <operators>...</operators>
          </data>
        </operator>
      </operators>
  </data>
</Job>
```

- Spouts
  - Socket
  - Apache Kafka
    - Distributed messaging system



Prototype

Job definition

```
<job>
  <name>...</name>  -----> Topology name
  <datasource>...</datasource>  -----> Spout

 <data save="bool">  -----> Stream of data
                     -----> Save Bolt

    <operators>
      <operator>*

          <class>
              <name>...</name>
              <method>...</method>
          </class>

          <data save="bool">

              <operators>...</operators>
          </data>
      </operator>
    </operators>
  </data>
</Job>
```

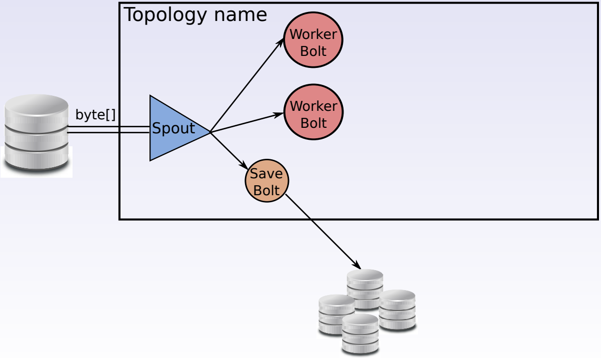
Prototype

Job definition

```
<job>
  <name>...</name> --------------------> Topology name
  <datasource>...</datasource> ----------> Spout

  <data save="bool"> --------------------> Stream of data
                                       └--> Save Bolt
      <operators> -------------------------> Stream processing
        <operator>* ------------------------> Operation

            <class>
              <name>...</name> ----------> Class name (inside Jar file)
              <method>...</method> -----> public byte[] methodName(byte[])
            </class>

            <data save="bool">

              <operators>...</operators>
            </data>
        </operator>
      </operators>
  </data>
</Job>
```

Prototype
Job definition

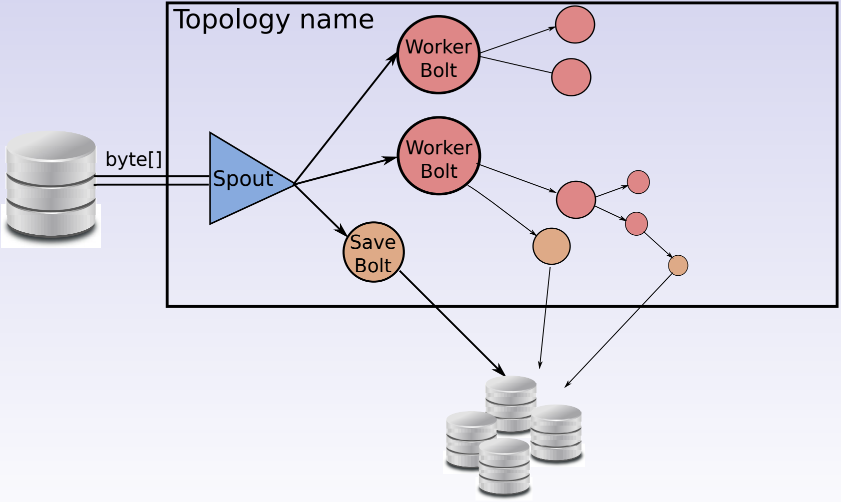
- Bolts
  - SaveBolts
    - Data storage into HDFS
    - Buffer → Hadoop SequenceFiles
  - WorkerBolt
    - Processing tuples
    - public byte[] methodName(byte[])



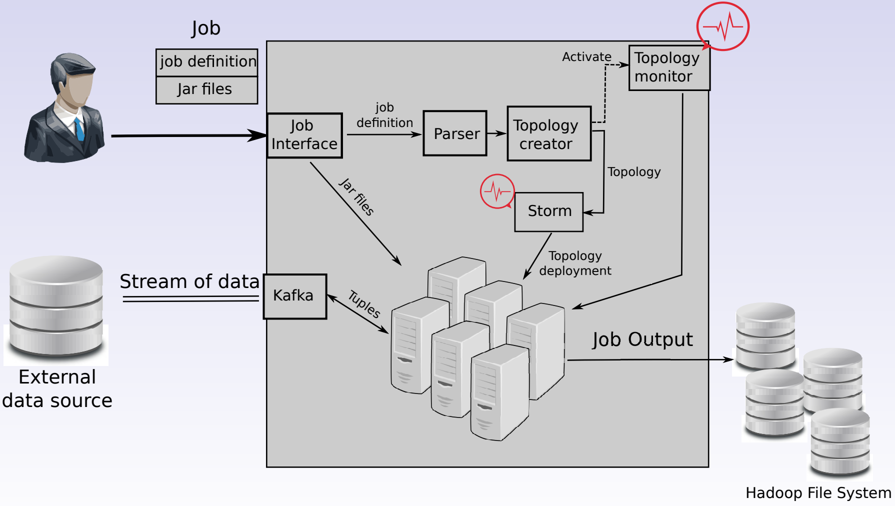
Prototype
Job definition

```
<job>
  <name>...</name>  ------------------>  Topology name
  <datasource>...</datasource>  ------>  Spout

  <data save="bool">  --------------->  Stream of data
                                    └->  Save Bolt
    <operators>  -------------------->  Stream processing
      <operator>*  ------------------>  Operation

        <class>
          <name>...</name>  --------->  Class name (inside Jar file)
          <method>...</method>  ----->  public byte[] methodName(byte[])
        </class>

        <data save="bool">  --------->  Stream of data
                                    └->  Save Bolt

          <operators>...</operators>  ->  ...
        </data>
      </operator>
    </operators>
  </data>
</Job>
```

Prototype
Job definition



Prototype
Job definition



- Internal monitoring system $\rightarrow$ Max pending tuples parameter.
  - Topology starts with a low parameter value.
  - Every ‘X’ seconds the monitor checks the ‘acked’ tuples.
  - First iteration $\rightarrow$ the monitor increases the parameter value.
  - Next iterations:
    - Current ‘acked’ tuples > previous ‘acked’ tuples $\rightarrow$ Increasing parameter value.
    - Current ‘acked’ tuples < previous ‘acked’ tuples $\rightarrow$ Decreasing parameter value.
    - Current ‘acked’ tuples == previous ‘acked’ tuples $\rightarrow$ Doing nothing unless this scenario was repeated ‘X’ times $\rightarrow$ Increasing parameter value.

- External monitoring system
  - Administrator can add rules.
    $Rule = (metric, operator, value, action)$
  - The monitor gets topology metrics every ‘X’ seconds. Each bolt produces a set of metrics.
  - The monitors evaluates each rules using the bolt metrics
  - The monitor applies the rule action in every Bolt which has triggered it.

Prototype
Monitoring system - Example

- Rule1:(capacity,<,0.4,-1)
- Rule2:(capacity,>,0.8,+2)

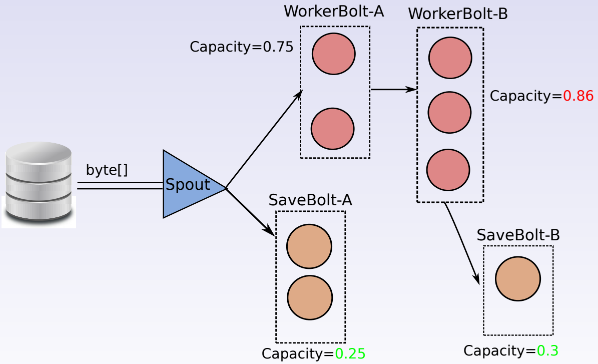


Table of Contents

- Goals
  - Efficient processing of huge external datasets
  - Heterogeneous data management
  - Processing of arbitrary functions
  - Infrastructure flexibility
  - Handling failures
  - **Data relations management**
  - **Efficient processing of huge internal datasets**

Big Data

A general approach to process external multimedia datasets

Thank you for your attention!