Towards Fast Multimedia Feature Extraction: Hadoop or Storm

David Mera, Michal Batko and Pavel Zezula

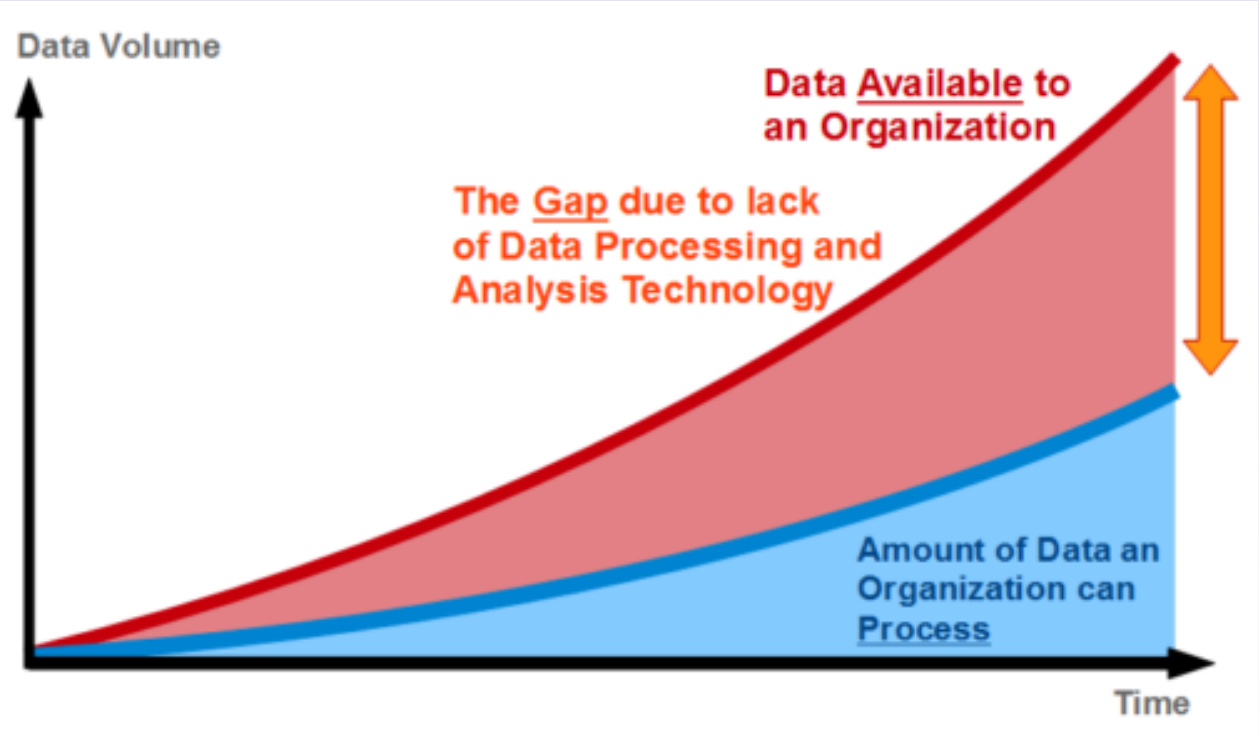
Laboratory of Data Intensive Systems and Applications (DISA)
Masaryk University
Brno, Czech Republic

IEEE International Symposium on Multimedia 2014
Taichung - December 12^{th}, 2014

Table of Contents

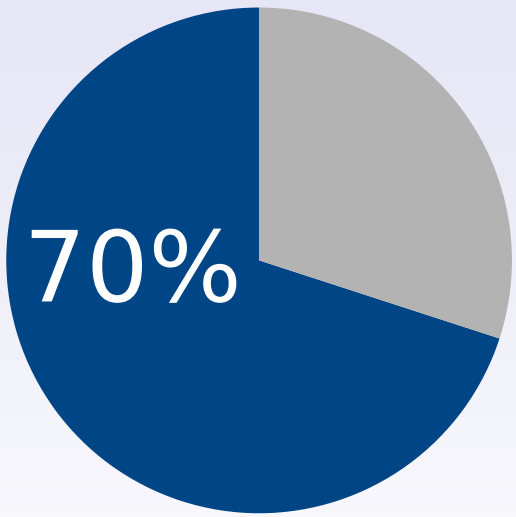
Table of Contents

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

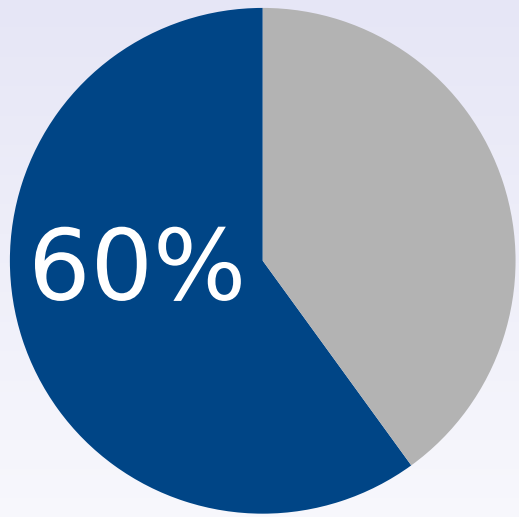


[1] Source: SINTEF. “Big Data - for better or worse”

- 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
  - ...



Non-Structured Data



Internet Traffic[2]

[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)
    - Real-time data processing: S4, Apache Storm

Table of Contents

Main goals

Main objective

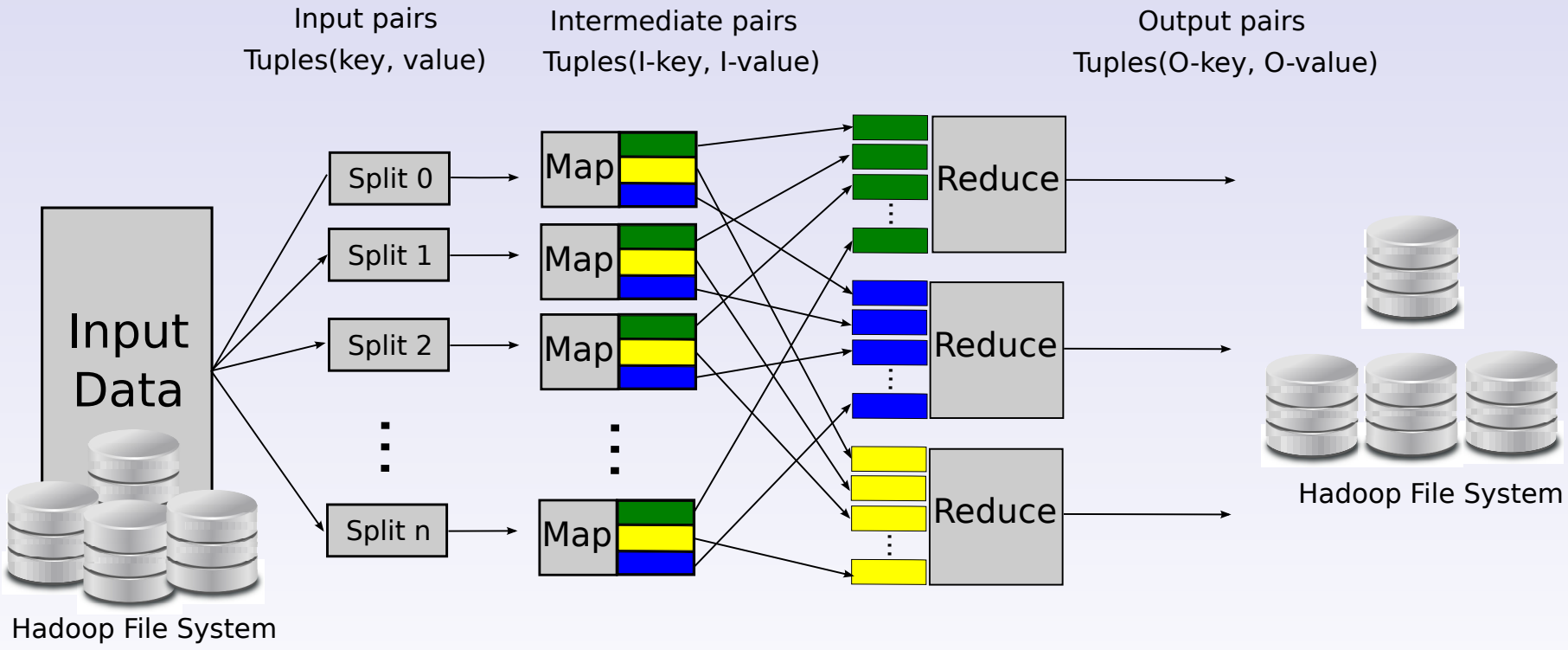
To compare several distributed computing processing frameworks in order to extract suitable features from a multimedia dataset. Specifically, the comparative will be focused on Apache Hadoop[3]and Apache Storm[4].

[3]Apache Hadoop: hadoop.apache.org

[4]Apache Storm: storm.apache.org

Table of Contents

- Map-Reduce paradigm



Processing frameworks
Apache Storm

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

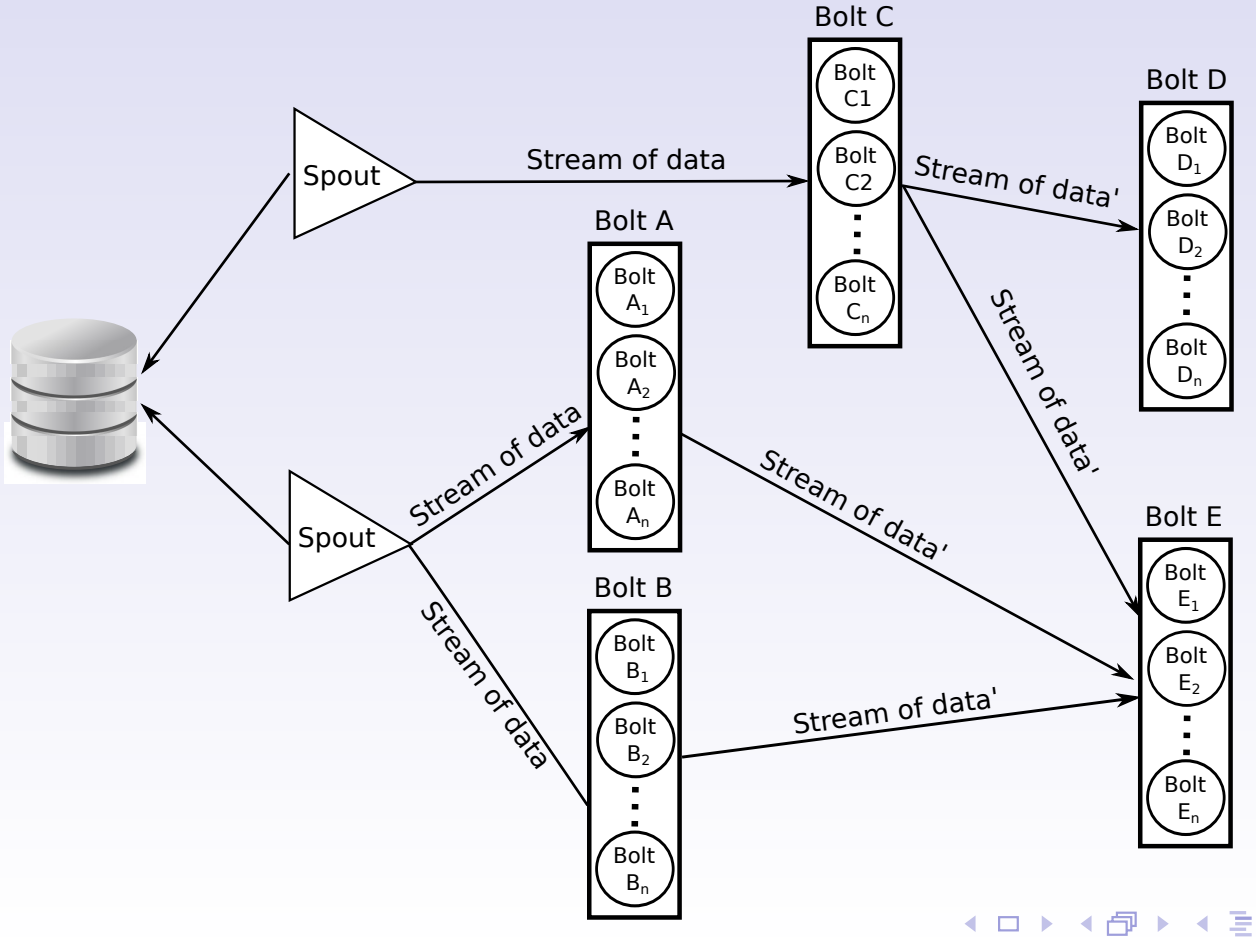
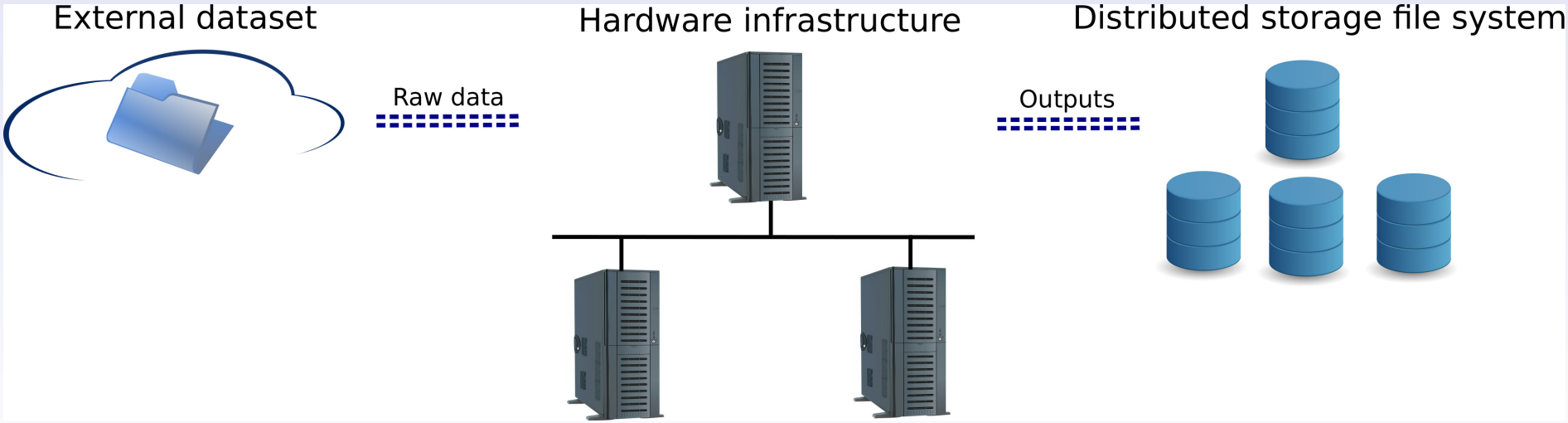


Table of Contents

Case-study: basis

The feature extraction of images stored into external datasets. The resulting features must be placed in a distributed organizational storage.



Testing scenarios
Sub-scenarios

Sub-scenario I

The external dataset must only be processed once.

Sub-scenario II

The external dataset could be processed several times.

Sub-scenario III

The external dataset could be processed several times. However, raw data can not be internally stored due to legal restrictions.

Table of Contents

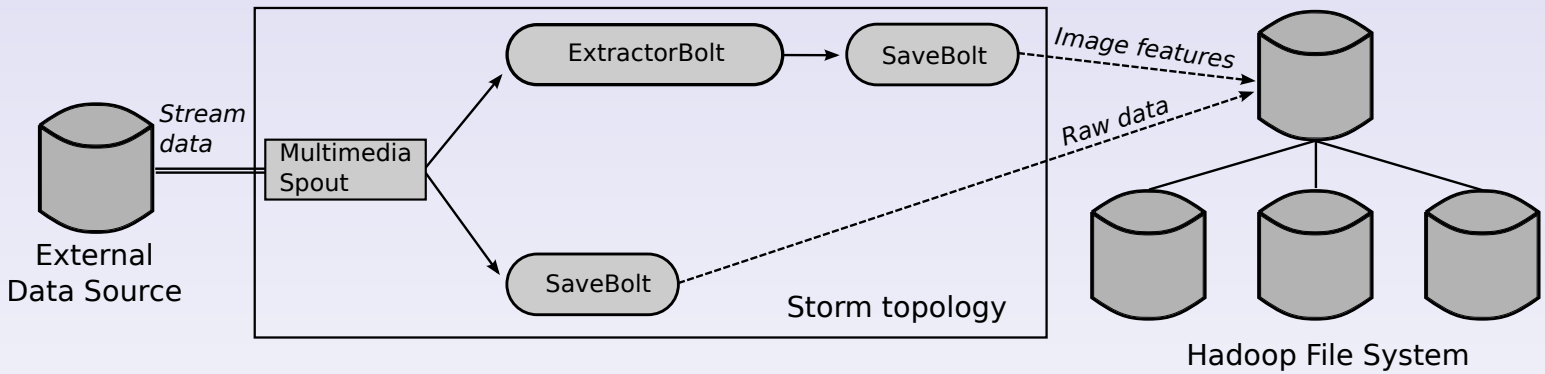
Infrastructure and datasets

- Hardware infrastructure - DISA cluster (4 nodes)
  - 2 x Intel-E5405@2Ghz CPUs
  - 8-physical cores
  - 16GB of RAM
  - 500GB SAS disk
  - Gigabit ethernet
- Dataset
  - One million of JPEG images
  - Average size: 61.9 KB
  - Total size: 61 GB
- Testing subsets
  - 10,000 images
  - 100,000 images
  - 1,000,000 images

Table of Contents

Empirical evaluation
Testing jobs

- Apache Hadoop - MapReduce Job
  - Job for retrieving external multimedia datasets and store them into the HDFS as SequenceFiles
  - Job for extracting image features
- Apache Storm - Topology



- Extraction of MPEG-7 image descriptors: MESSIF library extractor[5]
  - Feature extraction $\approx 0.5sec$ per image.

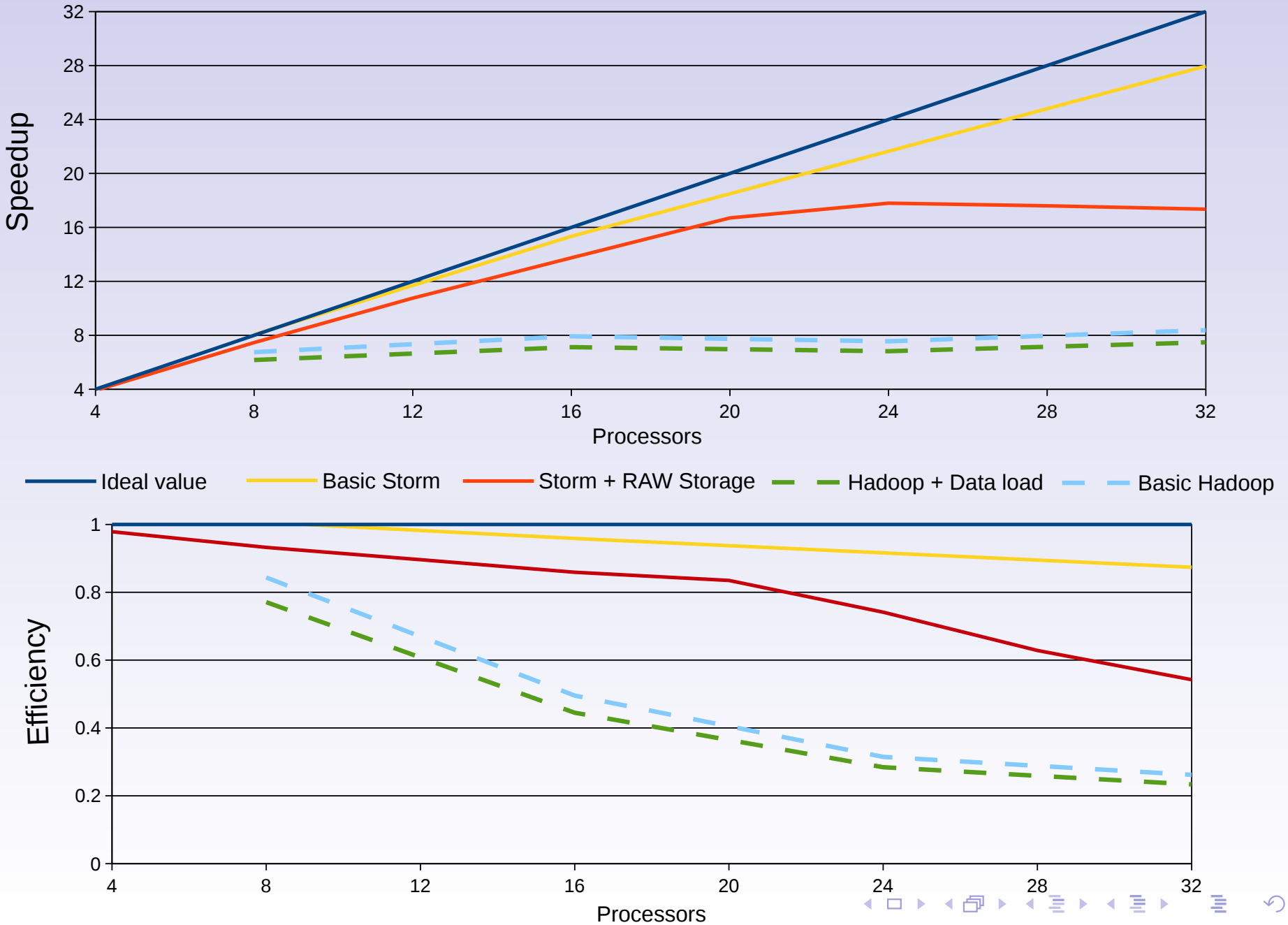
[5] M. Batko, D. Novak, and P. Zezula,"Messif: Metric similarity search implementation framework", in Digital Libraries: Research and Development. Springer, 2007.

Empirical evaluation
Evaluated metrics

- The Speedup ‘S’ measures how the rate of doing work increases with the number of processors k, compared to one processor
  - $S(k) = SeqJob(data) \div ParallelJob(data, k)$.
  - Ideally, $S(k) = k$
- Efficiency ‘E’ measures the work rate per processor
  - $E(k) = S(k) \div k$
  - Ideally, $E(k) = 1$
- Processing time

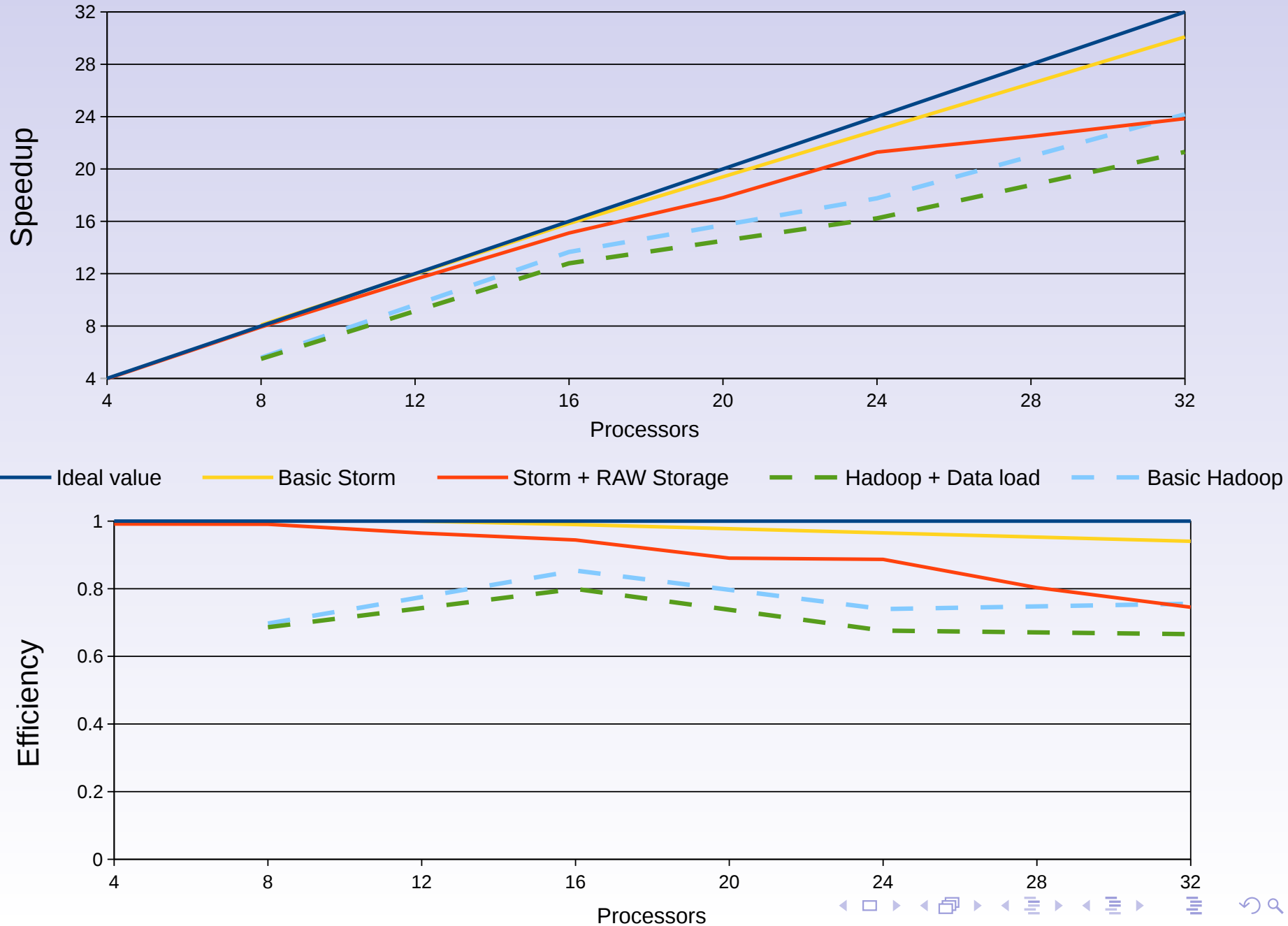
Empirical evaluation

Scalability experiments - 10,000 images



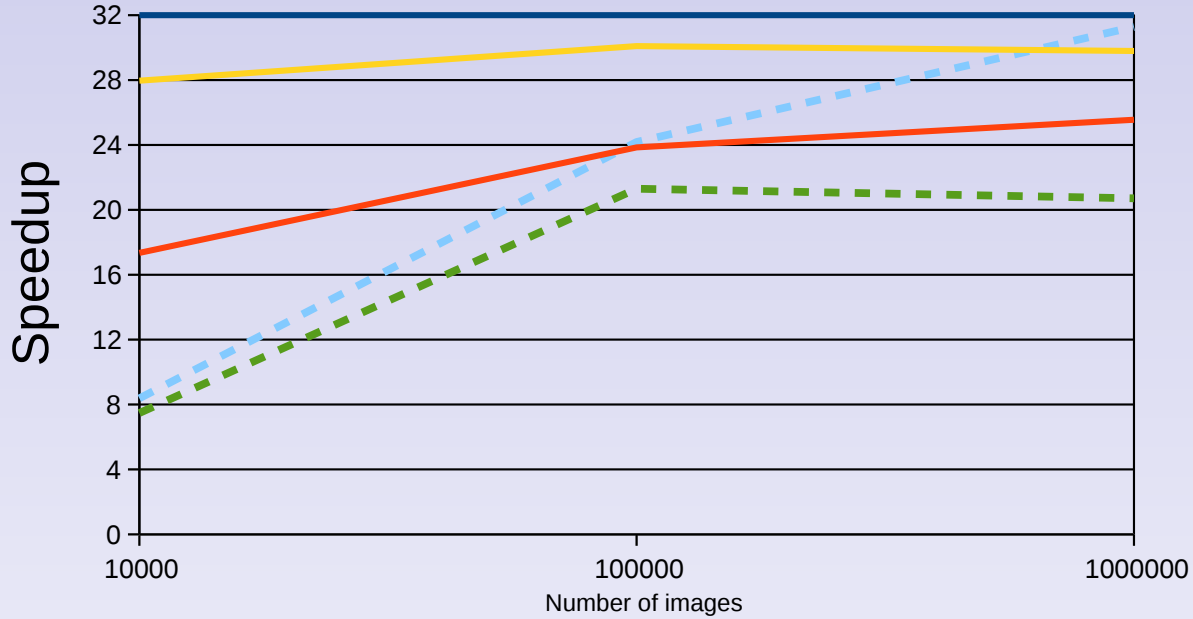
Empirical evaluation

Scalability experiments - 100,000 images



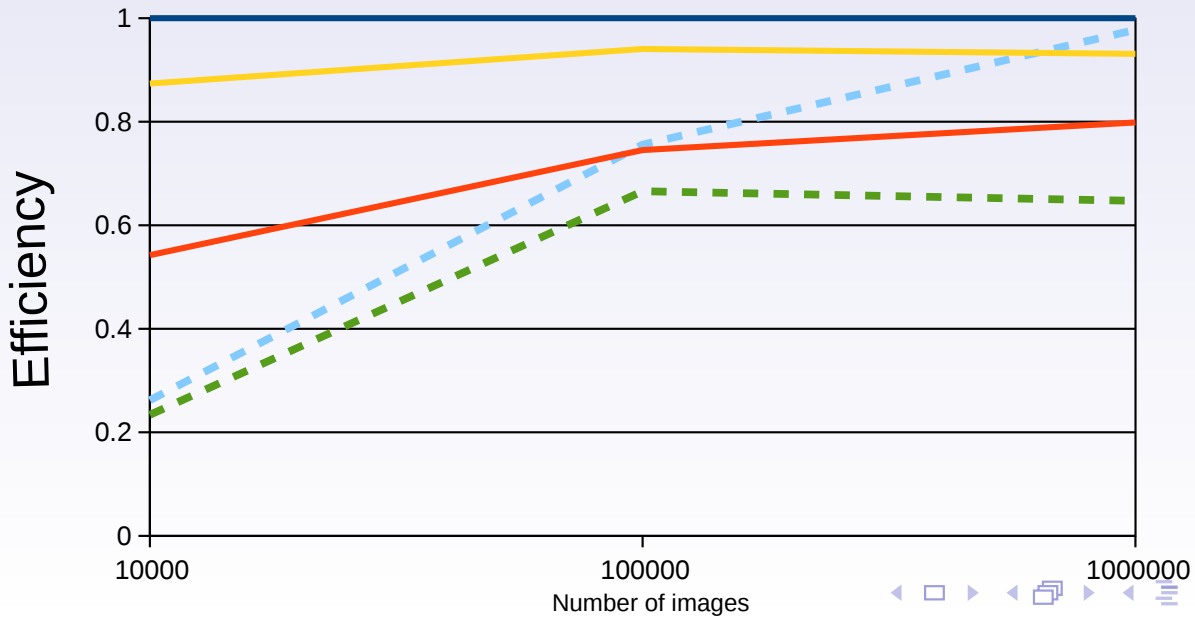
Empirical evaluation

Scalability experiments - 1,000,000 images



32 processors

Ideal value | Basic Storm | Storm + RAW storage | Hadoop + Data load | Basic Hadoop



32 processors

Empirical evaluation

Processing time - 1,000,000 images

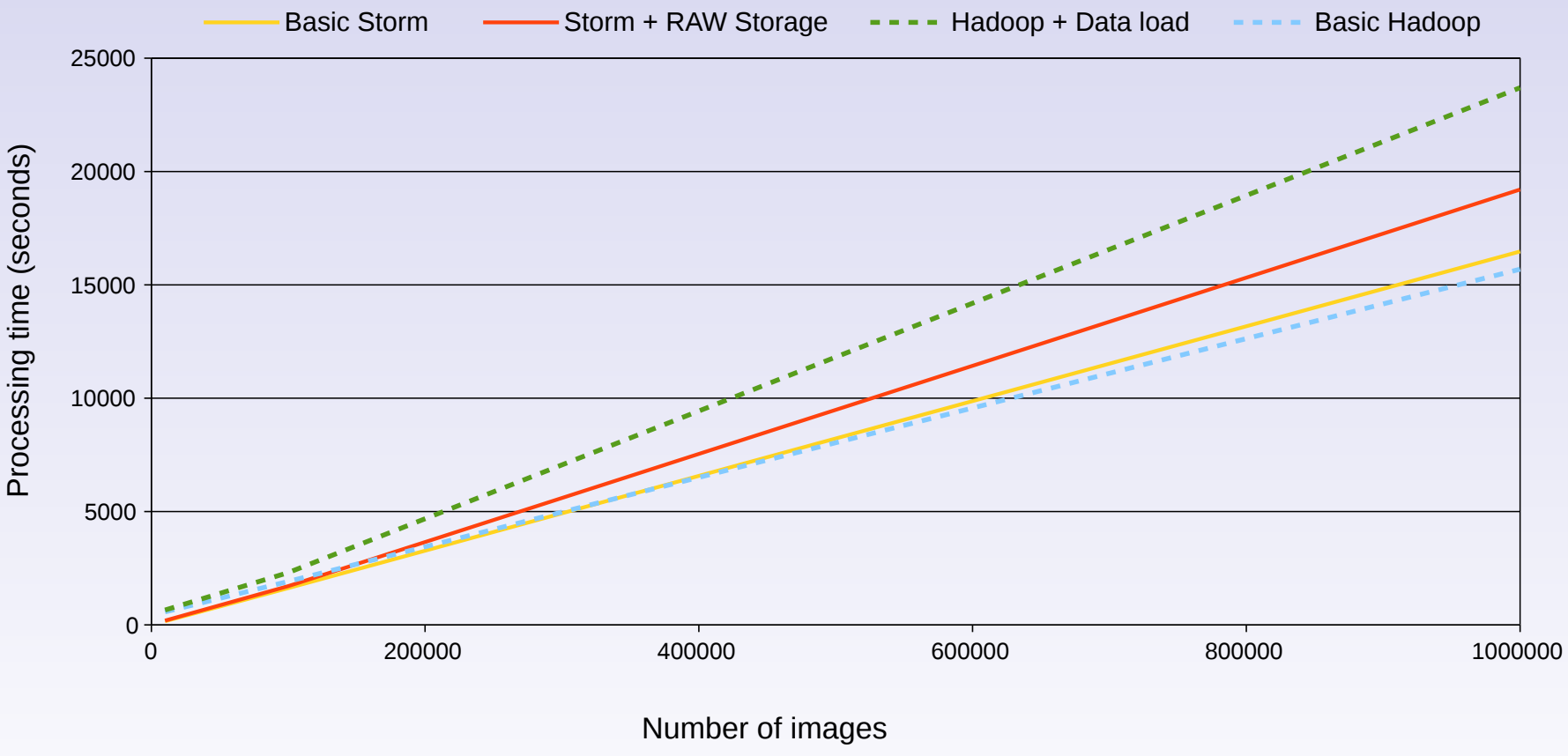


Table of Contents

- Sub-scenario 1: external data must only be processed once
  - Hadoop is less adecuate due to the data retrieval penalty
- Sub-scenario 2: external data could be processed several times
  - Apache Hadoop take advantage of data internally stored
  - Hybrid solution:
    - The first iteration: Apache Storm
    - The following iterations: Apache Hadoop
  - Exception: small-medium datasets which don't need to be stored
- Sub-scenario 3: external data could be processed several times. However, they cannot be stored.
  - Apache Storm has shown good performance for processing external datasets as long as they do not need to be stored

Conclusions and ongoing work
Conclusions

- Scalability: Storm scales better in small infrastructures, while Hadoop takes advantage of big ones
- Input data management: Hadoop requires data arrangement with small-medium images
- Configuration: Hadoop requires an iterative tuning of its configuration
- Job implementation: Storm is a low-level framework
- Job results: Hadoop must fully process data before showing results

Conclusions and ongoing work
Ongoing work

- New experiments
- A general adaptive system for processing multimedia datasets

Towards Fast Multimedia Feature Extraction: Hadoop or Storm

Thank you for your attention!

David Mera
dmera@mail.muni.cz