Noname manuscript No.
(will be inserted by the editor)

Multimedia Big Data Computing with Spark on MareNostrum Supercomputer

Mouna Makni · Leonel Cruz · Sana Imtiaz · Dani Mora · Mauro Gomez · Jonatan Poveda · Jordi Torres · Ruben Tous

Received: date / Accepted: date

Abstract In this paper we describe the design and evaluation of a framework to enable multimedia Spark workloads on MareNostrum, a petascale supercomputer. As far as we know, this is the first attempt to investigate optimized deployment configurations of this kinkd of workloads on a petascale HPC setup. We present the design of the framework and evaluate the scalability of the system. We examine the impact of different configurations including parallelism, storage and networking alternatives, and we discuss several aspects in executing multimedia big data workloads on a computing system that is based on the compute-centric paradigm. We derive conclusions to facilitate systematic and optimized methodologies for fine-tuning this kind of applications on large clusters.

Keywords Multimdia · Big Data · Spark · HPC · Instagram

1 Introduction

TODO (Ruben + help from Mouna)

2 Related Work

TODO (Ruben + help from Mouna)

Mouna Makni, Sana Imtiaz, Dani Mora, Jordi Torres and Ruben Tous
Universitat Politècnica de Catalunya (UPC). Barcelona, Spain
E-mail: TODO

Leonel Cruz, Mauro Gomez and Jonatan Poveda
Adsmurai. Barcelona, Spain

3 A framework to enable multimedia big data workloads on MareNostrum, an HPC setup

TODO (Ruben)

4 Benchmarking applications

4.1 Bag-of-Words based image classification on Instagram

The bag of visual words model, proposed by Csurka et. al. [2], has been widely used in computer vision. This model produces a description of each image feature using a visual vocabulary, and uses machine learning models to perform image recognition. By collecting information from labeled images having common characteristics, this method can classify and recognize a large number of images from different categories.

Fig. 1: Feature extraction [3]

This technique typically involves two main phases. Training is an offline process for creating the visual dictionary and computing image histograms. First, we detect and describe image feature points as presented in Figure **??**, using SIFT (Scale Invariant Features Transform) [4] in our work. The vocabulary is generated by clustering the resulting descriptors with k-means algorithm [5]. The visual words are clusters centroids which represents representative or common features, as illustrated in 2.

Fig. 2: Visual vocabulary

Each image feature is indexed by the cluster in which it belongs. Thus, each image is represented as a "bag of visual words", by generating an histogram (figure **??**) representing the importance of each entry of the visual vocabulary.

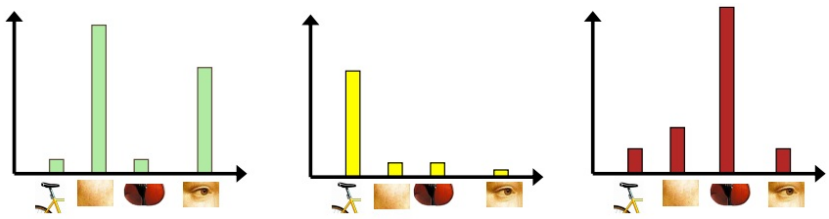

Fig. 3: Image classification

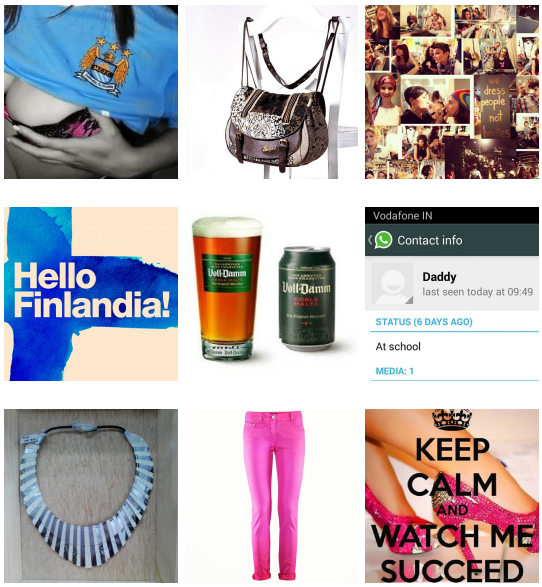

Fig. 4: Example true positives)

The second phase is predicting and represents the effective image classification against previously-labeled images. The process is executed on the given image to be recognized: extract SIFT feature points and calculate a descriptor for each feature point. After that, we match image features descriptors with the visual vocabulary and build the histogram which counts how many times each of the visual words occurs in the image. Images from a specific category will have more representative features from their category as compared to other features present in dictionary. Training images must have appropriate labels describing the class that they represent. We will train an SVM model [1], a binary classifier that builds a linear decision boundary between two classes of inputs. The histograms obtained as a result will be given to the SVM to predict respective category labels.

TODO (Mouna)

4.2 Deep convolutional networks based image classification on Instagram

TODO (Leonel)

4.3 Near-replica image detection on Twitter

TODO (Dani Mora)

5 Results

The main goal of the experiments is to evaluate the speed-up, scale-up, and size-up properties of the proposed framework applied to the selected workloads. To this end, we use datasets up to hundreds of GBs TODO of raw data. The size of RDDs is reported to be 2-5 times larger than that; in our experiments 400GBs of data in the sort-by-key application correspond to an RDD of 1TB. The cluster sizes range from 8 cores up to 1024 (i.e., 64 machines)...TODO

We have submitted and tested several hundreds of jobs to MareNostrum, but we describe only the results that are of significance. Our runs include an extensive set of configurations; for brevity, when those parameters were shown to be either irrelevant or to have negligible effect, we use default values. Each experimental configuration was repeated at least 5 times. Unless otherwise stated, we report median values in seconds.

TODO (Ruben)

5.1 Bag-of-Words based image classification on Instagram

TODO (Mouna)

5.1.1 speed-up

In the first set of experiments, we keep the input dataset constant and we increase the size of nodes/cores running the Spark application; whenever we refer to nodes, we mean MareNostrum machines that run the executors, while the driver always runs on a separate machine; each machine is equipped with 16 cores and 32 GB of RAM. The results from 128 (8 nodes) up to 512 cores (32 nodes) are shown in Figure 7, where we can see that for large datasets in terms of number of records, the training can scale well. In the figure, we present the performance for the most efficient configurations; we discuss these configurations in detail later. TODO

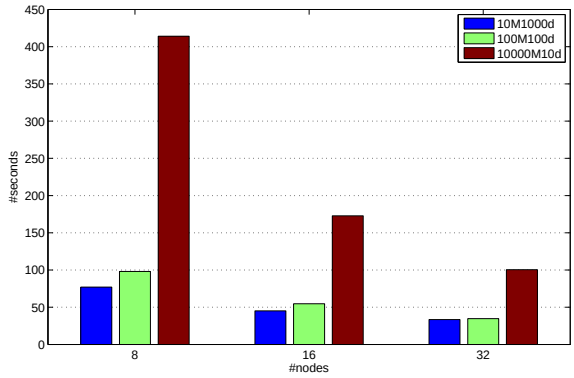

Fig. 5: Times for training....

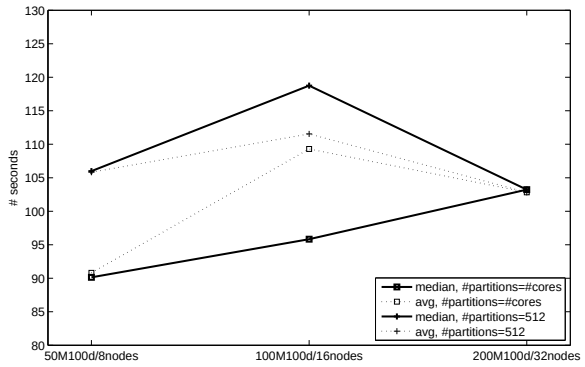

Fig. 6: Times for training....

5.1.2 scale-up

We process the same datasets, and we now modify both the number of records and the number of machines, i.e., the infrastructure scales-out. .... The results are shown in Figure **??**(top). In this figure, we show both the average and the median values. Ideally, all the plots should be horizontal; our system behaves closely to that. ....

5.1.3 size-up

We perform a third set of experiments, to assess the capability of sizing-up. We keep the number of nodes constant (either 16 or 32), and we gradually increase the dataset from 100GBs to 200GBs (raw data sizes). As shown in Figure

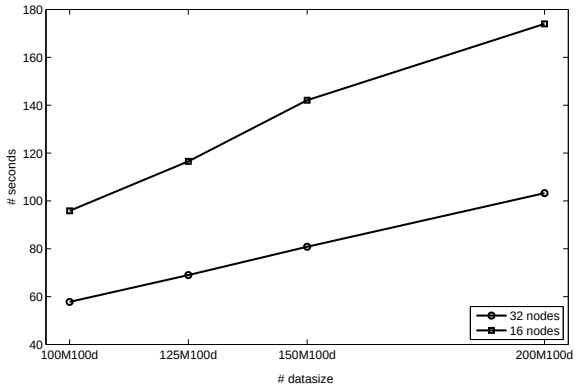

Fig. 7: Times for training....

??(bottom), Spark4MN exhibits a behavior where the curves are (almost) linear. ....

5.2 Deep convolutional networks based image classification on Instagram

TODO (Leonel)

5.3 Near-replica image detection on Twitter

TODO (Dani Mora)

6 Conclusions

The research work presented in this paper..... TODO

Acknowledgements This work is partially supported by the Spanish Ministry of Economy and Competitivity under contract TIN2015-65316-P and by the SGR programme (2014-SGR-1051) of the Catalan Government.

References

1. Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 144–152, New York, NY, USA, 1992. ACM.
2. Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cdric Bray. Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.

3. Robert Fergus. *Visual Object Category Recognition*. PhD thesis, University of Oxford, December 2005.
4. David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.
5. J. Macqueen. Some methods for classification and analysis of multivariate observations. In *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.