



## **Informe Técnico**

**Abstract:** This document is intended to provide an overview of the technical aspects related to running GAIA Big Data calculations at CESGA's new Hadoop on-demand platform.

---

Document Id.:	<b>CESGA-2013-003</b>
Date:	<b>16/12/13</b>
Responsible:	Javier Cacheiro
Status:	<b>FINAL</b>

---

# GAIA

## Running Big Data at CESGA



Document identifier: **DO\_SIS\_GAIA\_Technical\_Report\_V9.odt**

Date: **19/12/2013**

Document status: **DRAFT**

Document link:

License:



**Abstract:** This document is intended to provide an overview of the technical aspects related to running GAIA Big Data calculations at CESGA's new Hadoop on-demand platform.

Copyright notice:

Copyright © CESGA, 2013.

See [www.cesga.es](http://www.cesga.es) for details on the copyright holder.

GAIA is an ambitious mission of the ESA to chart a three-dimensional map of our Galaxy. For more information about GAIA please see <http://sci.esa.int/gaia/>

You are permitted to copy, modify and distribute copies of this document under the terms of the CC BY-SA 3.0 license described under <http://creativecommons.org/licenses/by-sa/3.0/>

Using this document in a way and/or for purposes not foreseen in the previous license, requires the prior written permission of the copyright holders.

The information contained in this document represents the views of the copyright holders as of the date such views are published.

THE INFORMATION CONTAINED IN THIS DOCUMENT IS PROVIDED BY THE COPYRIGHT HOLDERS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE MEMBERS OF THE EGEE-III COLLABORATION, INCLUDING THE COPYRIGHT HOLDERS, OR THE EUROPEAN COMMISSION BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THE INFORMATION CONTAINED IN THIS DOCUMENT, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Trademarks: Hadoop is a registered trademark held by The Apache Software Foundation. All rights reserved"

The icons used in this document were obtained from:

<http://www.iconarchive.com>

<http://www.iconarchive.com/show/icloud-icons-by-ahdesign91.html>

<http://www.archlinux.org/packages/extra/any/oxygen-icons/download>

<http://www.iconarchive.com/show/vista-hardware-devices-icons-by-icons-land.html>

<http://hortonworks.com/blog/a-set-of-hadoop-related-icons/>

## Document Log

Version	Date	Comment	Author
0	24/10/2013	Definition of the document structure	Javier Cacheiro
1	28/10/2013	GAIA related information	Diego Fustes
2	27/11/2013	Consolidating information	Javier Cacheiro
3	2/12/2013	Section 5 initial version	Javier Cacheiro
4	2/12/2013	Section 5 completed	Javier Cacheiro
5	3/12/2013	Consolidating information	Javier Cacheiro
6	5/12/2013	Consolidating information	Javier Cacheiro
7	12/12/2013	Ready for moderation	Javier Cacheiro
8	16/12/2013	Updated taking into account comments from Diego Fustes, Javier García and Ignacio López	Javier Cacheiro
9	19/12/2013	Ready for publication	Javier Cacheiro

Content

<b>1 Introduction.....</b>	<b>4</b>
1.1 Purpose of the document.....	5
1.2 Application Area.....	5
1.3 References.....	5
1.4 Document Amendment Procedure.....	5
1.5 Terminology.....	5
1.6 Conventions.....	6
<b>2 Executive Summary.....</b>	<b>7</b>
<b>3 Structure of this document.....</b>	<b>8</b>
<b>4 GAIA.....</b>	<b>9</b>
4.1 Introduction.....	9
4.2 Description of the Problem to be Solved.....	9
4.3 Resources Needed.....	10
<b>5 Designing the Big Data Solution.....</b>	<b>11</b>
5.1 Introduction.....	11
5.2 Analysing the possibilities.....	11
5.2.1 Supercomputers.....	11
5.2.2 CESGA IaaS Platform.....	12
5.3 The Birth of the CESGA Hadoop on demand service.....	18
<b>6 Results.....</b>	<b>21</b>
<b>7 Conclusions.....</b>	<b>26</b>

# 1 Introduction

## 1.1 Purpose of the document

This document provides a summary of the experience and knowledge gained during the execution of GAIA MapReduce jobs at CESGA.

## 1.2 Application Area

This document is intended for readers interested in the analysis of astronomical data and, in general, in Big Data solutions for scientific analysis; with its aim being to summarise the achievements and issues found running GAIA simulations at CESGA. To aid readers with specific interests, the sections of most relevance to several audiences are summarised in *Section 3 Structure of this document*.

## 1.3 References

**Table 1: Table of references**

<b>R1</b>	GAIA mission, <a href="http://sci.esa.int/gaia/">http://sci.esa.int/gaia/</a>
<b>R2</b>	J. Bruijne, Science performance of Gaia, ESA's space-astrometry mission, <i>Astrophysics and Space Science</i> 341 (1) 31-41. doi:10.1007/s10509-012-1019-4. URL <a href="http://dx.doi.org/10.1007/s10509-012-1019-4">http://dx.doi.org/10.1007/s10509-012-1019-4</a> .
<b>R3</b>	C. A. L. Bailer-Jones et al., The Gaia astrophysical parameters inference system (Apsis). Pre-launch description, ArXiv e-prints arXiv:1309.2157.
<b>R4</b>	Reimers, D. et al. The Hamburg/ESO survey for bright QSOs. II. Follow-up spectroscopy of 160 quasars and Seyferts. <i>Astronomy and Astrophysics Supplement</i> , v.115, p.235.
<b>R5</b>	<i>Running Hadoop in the cloud</i> , Álvaro Simón, Javier López Cacheiro, et al; Ibergrid 2013 proceedings.
<b>R6</b>	Big data challenges, an insight into the Gaia Hadoop solution, Pierre-Marie Brunet, Alain Montmorry, et al; Spaceops 2012 proceedings.
<b>R7</b>	Data Management at Gaia Data Processing Centers, Pilar de Teodoro, Alexander Hutton, et al

## 1.4 Document Amendment Procedure

This document is under the responsibility of CESGA. Amendments, comments and suggestions should be sent to Javier Cacheiro (jlopez [at] cesga.es).




## 1.5 Terminology

**Table 2: Glossary**

<b>ESA</b>	European Space Agency
<b>GAIA</b>	Gaia is an ambitious mission to chart a three-dimensional map of our Galaxy, the Milky Way
<b>Hadoop</b>	Apache project to provide an open-source implementation of frameworks for reliable, scalable, distributed computing and data storage
<b>IaaS</b>	Infrastructure as a Service (IaaS) is a cloud computing service model based on providing plain computing and storage resources on-demand

## 1.6 Conventions

This document uses several conventions to highlight certain words and phrases and draw attention to specific pieces of information.

	This icon indicates tips that could be useful for the reader.
	This icon indicates important considerations that are easily missed.
	This icon indicates critical aspects that should not be overlooked.

## 2 Executive Summary

Gaia: The 'impossible space mission' ready to fly  
*BBC News, 21 August 2013*

Gaia is an ambitious mission from the European Space Agency (ESA) whose objective is to produce a three-dimensional map of our Galaxy, the Milky Way. This will be the first highly accurate census of the Milky Way, measuring positions, motion and properties of hundreds of thousands of stars. The Gaia's satellite is expected to be launched by December 19, 2013.

Gaia will pinpoint about a billion stars and build a profile on each and every one. The main astrophysical properties of astronomical objects observed by Gaia will be derived by a *software pipeline*, which is being developed by an international consortium, the Gaia Data Processing and Analysis Consortium (DPAC). DPAC arose as an international collaboration with memberships from all over Europe, which nowadays includes a community of over 400 scientists and software engineers from more than 20 countries. DPAC is organized in several coordination units (CUs) and responsible for a well-defined set of tasks in the Gaia data processing effort. CU8 was in charge of classifying the observed astronomical sources by both supervised and unsupervised algorithms, and of producing an outline of their main astrophysical parameters.

In preparation for Gaia data, the work presented in this technical report addresses the unsupervised analysis of the Hamburg/ESO survey. The Hamburg/ESO survey (HES) is a digital objective prism survey covering the total southern extragalactic sky. HES is composed by a total of 4.5 million objects with corresponding spectroscopy. Therefore, it is very difficult to search for interesting objects and to characterize the different classes of objects. At this work we allow for knowledge discovering by computing a *Self-Organizing Map* (SOM) that provides a model that summarizes the information embedded in the survey. The amount of computation required to train a SOM with 5 million spectra is very high and a distributed version of the learning algorithm is required. Apache Hadoop was chosen because it provides scalability for very big datasets, as the ones expected from Gaia and other future surveys.

The execution of Gaia jobs at CESGA presented several challenges both from the application and infrastructure point of view. The MapReduce application had to be improved and the Hadoop cluster had to be optimized for maximum performance.

The main challenges encountered are explained in detail this document as well as the solutions adopted with the hope that they can serve as a guidance for other users or administrators facing similar problems.

The work performed in the scope of this project ultimately led to the birth of the CESGA *Hadoop on demand service* which leverages CESGA's cloud IaaS platform to run Big Data.



### 3 Structure of this document

The document is structured as follows:

In Section 4 we provide an introduction to the GAIA mission of the European Space Agency, giving a description of the problem to be solved and the resources needed.

In Section 5 we provide a description of the Big Data solution deployed to address GAIA needs. We include an analysis of the possibilities considered as well as a description of the final CESGA Hadoop on demand service created.

In Section 6 we give a summary of the results obtained from the unsupervised analysis of the Hamburg/ESO survey.

Finally, in Section 7 we recapitulate the main conclusions of this work.

## 4 GAIA

The Milky Way is nothing else but a mass of innumerable stars  
planted together in clusters.  
*Galileo Galilei*

### 4.1 Introduction

Gaia [R1] is a cornerstone mission from the European Space Agency (ESA) that is expected to be launched by December 2013. It will provide the first highly accurate 6-D map of the Milky Way, measuring positions, parallaxes, and motions to the microarcsec level. The satellite's complex instrumentation, mode of operation, astrophysical main objectives, and its expected scientific performance have been extensively reviewed elsewhere, see for example [R2]. Since Gaia is the first non-biased survey of the entire sky down to approximately magnitude 20, it is raising enormous expectation from a wide range of astronomical research areas, going from Solar System to Cosmology, especially after it was decided that the final archives, containing the observations and basic astrophysical products, will be made public immediately after being produced.

The main astrophysical properties of astronomical objects observed by Gaia will be derived by a software pipeline, which is being produced by an international consortium, the Gaia Data Processing and Analysis Consortium (DPAC). DPAC arose as an international collaboration with memberships from all over Europe, which nowadays includes a community of over 400 scientists and software engineers from more than 20 countries. DPAC is organized in several coordination units (CUs) and responsible for a well-defined set of tasks in the Gaia data processing effort. CU8 was in charge of classifying the observed astronomical sources by both supervised and unsupervised algorithms, and of producing an outline of their main astrophysical parameters.

The CU8 Astrophysical parameters inference system (see [R3] Bailer Jones et al.), is subdivided into several work packages: DSC (Discrete Source Classifier) is the main package for classification, whereas GSP-Phot (General Stellar Parameterizer - Photometry) and GSP-Spec (General Stellar Parameterizer - Spectroscopy) are the main parameterization packages. There are a number of additional packages dedicated to more specific tasks, such as Quasar/Galaxy parameterization (QSOC and UGC, respectively) or specific stellar population parameterizers (ESP). Finally, there are two packages dedicated to the unsupervised analysis of the raw data, OCA (Object Cluster Analysis) and OA (Outlier Analysis).

This work focuses on the implementation of unsupervised analysis (OA) methods in big data environments.

### 4.2 Description of the Problem to be Solved

In preparation for Gaia data, we have addressed the unsupervised analysis of the Hamburg/ESO survey. The Hamburg/ESO survey (HES, see [R4] Reimers et al.) is a

digital objective prism survey covering the total southern extragalactic sky. It is based on Kodak IIIa-J plates which have been taken with the 1m ESO Schmidt telescope and its 4° prism. The spectral coverage is 3200-5300 Å, at an resolution of 15 Å at H gamma. HES has a broad range of scientific goals. compile samples of high-redshift, bright quasars suited for high-resolution spectroscopy, study the evolution of the most luminous part of the QSO population, look for the metal-poorest stars, white dwarfs. Etc.

HES is composed by a total of 4.5 million objects with corresponding spectroscopy. Therefore, it is very difficult to search for interesting objects and to characterize the different classes of objects. At this work we allow for knowledge discovering by computing a Self-Organizing Map (SOM) that provides a model that summarizes the information embedded in the survey. The SOM is a powerful tool that provide quality clustering and dimensionality reduction at the same time. It projects the dataset in a two dimensional grid of neurons where the data topology is preserved. Each neuron behaves as a cluster, such that it contains a bunch of objects that share the same properties, since they are very similar each other. Furthermore, close neurons in the map are similar, while distant neurons are dissimilar, as a result of the topology preservation.

In order to analyse the HES spectroscopic data, we compute a SOM with 30x30 neurons. To do so, we first initialize the map randomly, with patterns close to the dataset mean. Then, we apply the batch SOM learning procedure iteratively. The neighbourhood around the winning neuron is decreased until 0, when only the winning takes part in the neuron weight updating. The process runs during around 500 iterations. Since the amount of computation required to train a SOM with 5 million spectra is very high, we needed to implement a distributed version of the learning algorithm, in this case using Apache Hadoop as distributed computing platform. Hadoop has been chosen because it provides scalability for very big datasets, as the ones expected from Gaia and other future surveys.

### **4.3 Resources Needed**

The computation of the SOM with HES data needs a Hadoop cluster where to perform the MapReduce simulations. The number of nodes should be as big as possible, since, fortunately, the algorithm is completely distributable and scales well with the number of nodes.

For the final simulations a Hadoop cluster of approximately 100 nodes is required in order to complete the analysis in a reasonable amount of time (days).

## 5 Designing the Big Data Solution

### 5.1 Introduction

The Big Data community is working to extend the scalability of traditional databases (RDMS) using new technologies based on a share-nothing architecture. Gaia analysed different Big Data solutions [R6, R7] before deciding to use the Hadoop platform to perform its data analytics.

Hadoop is an open-source framework that implements the MapReduce computational paradigm on top of a parallel HDFS filesystem. Running Hadoop in a traditional supercomputing centre presents several challenges due to the fact that most of these centres rely on clusters using shared storage and Infiniband technologies that are very good at solving large problems programmed using MPI.

When talking about Hadoop we are talking about a share-nothing architecture that benefits from having dedicated local storage in each of the nodes of the cluster and, in general, makes little use of low latency communications so a Gigabit network usually is enough to its purposes. In this sense we have recently benchmarked the possibilities of Hadoop in federated environments like the EGI FedCloud infrastructure [R5] and we saw little influence in the fact that some of the nodes were at a remote location.

For the work described here the problem is simplified because we have only to deploy a local hadoop cluster inside a given data centre.

In the following sections we describe the different alternatives analysed as well as the final configuration that led to the creation of the Hadoop on demand service.

### 5.2 Analysing the possibilities

The first question to decide was where we could actually deploy our Big Data solution. For this we performed an analysis of the different platforms available at CESGA in order to decide which one fit better to our purposes.

The platforms analysed were the current supercomputers deployed at CESGA and the newer IaaS Platform based on OpenNebula.

#### 5.2.1 Supercomputers

The first alternative to run Hadoop at CESGA was to run it on top of one of the existing supercomputers. Currently there are two supercomputers available: Finisterrae (FT) and SVG. The first is a cluster based on old Itanium processors with a dedicated Infiniband network while the later is a heterogenous cluster of X86\_64 processors.

Obviously Finisterrae was discarded because of its IA64 architecture which makes difficult to run Java applications—at the end Hadoop is not more than a Java

application. So the only real option was to use SVG, indeed its cluster architecture based on X86\_64 nodes offered a good alternative to run Hadoop.

After choosing SVG as the next step was to decide the best way to integrate Hadoop in the cluster operations. SVG uses a batch system to automatically distribute jobs between the nodes so, in order to run Hadoop, we had to make sure that the nodes were not used by other jobs. To achieve this objective there were two options:

- Tight integration with the batch system, so that the Hadoop cluster is deployed through the batch system
- Advanced reservation of nodes needed for the cluster and then launching the Hadoop cluster manually

In the first case there were some alternatives available for SGE—the current SGE batch system—but they did not seem good enough because they were not able to guarantee key aspects like the fact that the nodes that run the HDFS and the TaskTracker nodes are the same. The TaskTracker are run on demand as SGE jobs but the HDFS filesystem is already pre-deployed at certain nodes.

So the best alternative to achieve our aim seemed to use advanced reservations and then launch the Hadoop cluster on those nodes. To help in this process dedicated *init.d* scripts were developed to start the DataNode and TaskTracker services.

The initial Hadoop clusters were deployed using this procedure, but the advanced reservation procedure showed to be inefficient to perform quick deployments, which was required for the initial tests. It is certainly a good alternative for scheduled production runs but inefficient for testing and quick deployments.

### 5.2.2 CESGA IaaS Platform

The next alternative evaluated to achieve faster deployments was using the CESAG IaaS cloud platform based on OpenNebula. This infrastructure has similar hardware resources than SVG but it is less saturated in terms of usage so it is easier to get enough resources to run a Hadoop cluster in a short amount of time. In general only larger deployments required coordination with the people on charge of the cloud platform, and mainly due to limitations in the public IP address space used in the platform which restricted how many nodes could be run concurrently.

All Hadoop nodes run as virtual machines using KVM. In order to increase the performance of the nodes several optimisations were performed including using virtio drivers, CPU passthrough, raw disk images, on the fly scratch disk creation, disabling virtio disk cache, and increasing the read ahead factor in the OS. Each Hadoop slave is assigned one virtual CPU and one physical CPU and 1GB of memory. The disk is shared with other instances running in the same host.

The hosts are HP Proliant SL2x170z G6 server with two Intel Xeon E5520 processors, one SATA disk and Gigabit connectivity.

In order to simplify the later configuration, we created a customized virtual machine (VM) image template that includes all the software necessary to run Hadoop. This image is based on Scientific Linux 6.4 and contains the following customizations:

- **iptables** is configured to allow inter-cluster communication.
- **SSH**: enable root access through ssh public key authentication `authorized_keys`.
- **Modules**: we make use of the Modules package to simplify the deployment of the software needed to run Hadoop.
- **Java**: we use Oracle Java JDK 1.7.0\_21 because it offers better performance than the OpenJDK version and supports well GAIA software. It is also configured through Modules to allow simple migration to other Java versions included like Oracle Java JDL 1.6.0\_38.
- **Hadoop**: both Hadoop-1.0.3, Hadoop-1.0.4 and Hadoop-1.1.2 are included. The configuration using Modules makes easy to include additional versions and decide the one we want to use when starting the Hadoop cluster. The default version is Hadoop-1.1.2.



The following changes were applied to the OpenNebula VM templates to improve performance

- Activate **virtio** module for disk and network devices.
- Use small **RAW** images (4GB) for the O.S. instead QCOW2.
- Generate Hadoop HDFS local disk storage and swap on the fly for each instance
- Disable KVM disk cache.
- Change the IO scheduler algorithm to **deadline**.
- Increase disk **readahead** value: `blockdev --setra 8192 /dev/vdc`.
- Use **host-passthrough CPU mode**.



Below we include extended details about the optimizations used for slave nodes.

Hadoop slave OpenNebula template:

```
DISK=[
  BUS="virtio",
  CACHE="none",
  DRIVER="raw",
  IMAGE_ID="hadoop",
  TARGET="vda",
  TYPE="OS" ]

DISK=[
  BUS="virtio",
  CACHE="none",
  FORMAT="ext4",
  SIZE="31480",
  TARGET="vdc",
  TYPE="fs" ]
```

```
NIC=[
  MODEL="virtio",
  NETWORK_ID="8" ]

RAW=[
  DATA="<cpu mode='host-passthrough' />",
  TYPE="kvm" ]
```

Contextualization script optimizations:

```
# virtual-guest
# Deadline scheduler
# Readahead extended to 8KB from 512 bytes
for disk in vda vdb vdc; do
    echo deadline > /sys/block/${disk}/queue/scheduler
    blockdev --setra 8196 /dev/${disk}
done
```

/etc/sysctl.conf optimizations:

```
# Minimal preemption granularity for CPU-bound tasks:
# (default: 1 msec# (1 + ilog(ncpus)), units: nanoseconds)
kernel.sched_min_granularity_ns = 10000000

# This option delays the preemption effects of decoupled workloads
# and reduces their over-scheduling. Synchronous workloads will
# still
# have immediate wakeup/sleep latencies.
kernel.sched_wakeup_granularity_ns = 15000000

# swapping low. It's usually safe to go even lower than this on
# systems with
# server-grade storage.
vm.swappiness = 30

# The generator of dirty data starts writeback at this percentage
# (system default
# is 20%)
vm.dirty_ratio = 40
```

After having the cluster running, the next step is to prepare the nodes to run Hadoop. Our Hadoop cluster will consist of one *master* node and a variable number of *slave* nodes depending on the size of the cluster. For example in the case of a 101 node cluster, one node will be configured as master and the remaining 100 as slaves. The

master node will run the NameNode, SecondaryNameNode and JobTracker services and each of the slaves will run just one TaskTracker and one DataNode services due to the fact that we are using small VM instances with just 1GB of RAM and 1 CPU we decide to run just one tasktracker and one DataNode per slave.



The tuned configuration parameters are given in Table 1. The column type identifies the configuration file where the parameter is set.

Parameter	Type	Value
fs.inmemory.size.mb	core-site	200MB
io.file.buffer.size	core-site	128KB
mapreduce.task.io.sort.factor	core-site	100
mapreduce.task.io.sort.mb	core-site	100
mapred.tasktracker.map.tasks.maximum	mapred-site	1
mapred.tasktracker.reduce.tasks.maximum	mapred-site	1
mapred.reduce.tasks	mapred-site	0.95 x num. slaves
dfs.datanode.du.reserved	hdfs-site	1GB
dfs.block.size	hdfs-site	<b>16MB</b>
dfs.replication	hdfs-site	3
HADOOP_HEAPSIZE	hadoop-env	512MB slaves / 1GB master

*Table 1: Tuned Hadoop configuration for GAIA. It includes both the HDFS and MapReduce parameters used.*

These parameters have been configured to adapt the Hadoop cluster to resources available—1GB of RAM and 1CPU per slave node and 2GB RAM and 2 CPU for the master—. The *dfs.replication* parameter indicates how many copies we want to store of each block, in this case we use three replicas that will allow different nodes to execute the same mapreduce tasks—speculative execution—mitigating the performance degradation that could be experienced in a heterogeneous cluster with VMs running under hypervisors with different load conditions. The *mapred.tasktracker.map.tasks.maximum* and *reduce.tasks.maximum* are set to 1 because we only have 1 CPU available to run both services and the HADOOP\_HEAPSIZE is set to 512MB to allow both running at the same time without exhausting the node's memory. The number of reduce tasks, *mapred.reduce.tasks*, is set to 95% of the number of slaves in the Hadoop cluster.

The *dfs.block.size* was carefully selected to split the problem trying to obtain a good compromise between the level of parallelism and the time it takes each map task, a **dfs.block.size** value of **16MB** proved to be a good compromise generating around **5**



**maps/node.** Ideally we should be in the 10-100 maps/node range but in this case the size of the problem to be solved for each map will be too small and the time needed to expand new map task would represent an important overhead.

It also proved very effective to increase the reduce tasks from 1 to a number equivalent to the 95% of the slave nodes. This guarantees an appropriate level of parallelism during the reduce phase of the calculation.

To reduce the initial deployment time several optimizations were also implemented in OpenNebula that allowed to reduce the time to deploy a 101 node cluster from 2'5 hours to just 18 minutes. In the following table the deployment times after applying the optimizations are compared with those obtained using Amazon EC2 and EBS storage for reference. For the Amazon AWS tests we used both Whirr and a slightly modified version of our scripts to take advantage of the possibility that offers Amazon AWS to start an array of instances in just one command. *Whirr* was not able to start more than 21 instances due to the high number of requests that its underlying framework *jclouds* generates when increasing the number of cluster nodes which causes Amazon to block the whirr client because it exceeds the request rate limit established by AWS.

# nodes	Amazon EC2 + EBS using Whirr (m1.small)	Amazon EC2 + EBS using our hadoop-start script (m1.small)	CESGA OpenNebula using our hadoop-start script
10	14m51s	3m11s	5m40s
21	27m50s	2m58s	4m53s
51	Failed	9m43s	13m03s
101	Failed	Failed (4m57s) <sup>1</sup>	18m31s

Table 2: Hadoop cluster startup times.

## Initial test case

To perform a fast benchmark of the solution we used a small test case with simulated data of the GAIA experiment. The input file size is 2.2GB:

2.2G gaia/input/Test-G150\_Lambda\_NoExtinction.txt

<sup>1</sup> In the case of the 101 cluster we were unable to get the 101 instances working. In all our 101 tests we always found that some of the instances did not have connectivity with the other instances in the security group. We re-tried several times but we were always hit by this issue. The time reported is the average time to start the cluster and configure the available nodes.

It took around 10 min per iteration with a 11 nodes Hadoop cluster that used 10 of these nodes as data nodes.

This was a reasonable result with a similar performance to what was obtained in the SVG cluster.

### Scalability Analysis

In order to analyse scalability we decided to duplicate the input file 5 times—by simply concatenating its contents 5 times—to increase the run time for 1 iteration. The final input file has a size of 11GB:

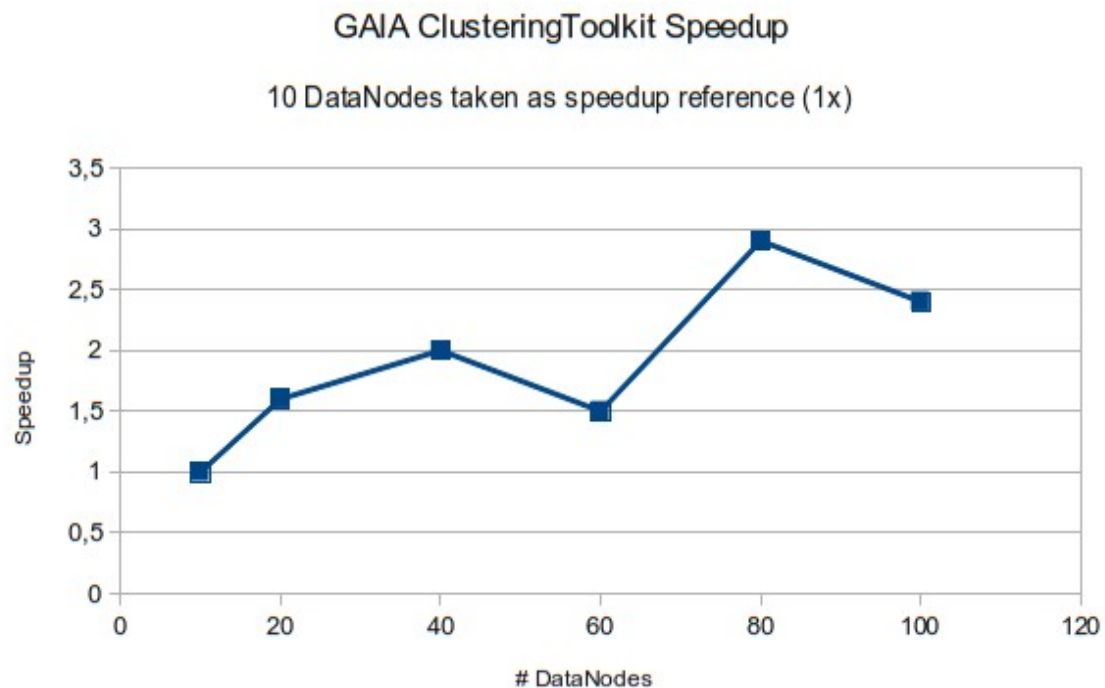
11G gaia/inputx5/Test-G150\_Lambda\_NoExtinction-5x.txt

The HES dataset that would be analysed later is around 8GB so this is a reasonable assessment of the performance of the system.

In the following table we show the results of the scalability analysis. Take into account that each cluster configuration reserves 1 node as master node so only N-1 nodes are used as slaves—running each 1 DataNode and 1 TaskTrack. Each node has 1 CPU and 1GB of RAM.

# nodes	dfs.block.size	dfs.replication	put	ClusteringToolkit -Hadoop-1.0	Speedup
11	128MB	2	17m24s	47m6s	1x
21	64MB	2	8m21s	29m16s	1.6x
41	32MB	2	7m2s	23m18s	2.0x
61	16MB	2	8m9s	31m36s	1.5x
61	32MB	2	9m48s	30m55s	1.5x
81	16MB	2	6m47s	16m28s	2.9x
81	16MB	4	17m46s	17m31s	2.7x
101	16MB	2	10m12s	20m1s	2.4x
101	32MB	2	12m26s	21m6s	2.2x

*Table 3: Gaia scalability using input file x 5 (11GB) and 1 iteration*



*Figure 1: Gaia scalability analysis*

To explain the odd behavior seen at 61 and 101 configurations we must take into account that scalability results are greatly influenced by external conditions outside our control like:

- The virtual machine distribution, i.e. how OpenNebula distributed the VMs between the physical hosts
- The load due to other VMs running in the same physical hosts

In any case Hadoop manages quite well a heterogeneous environment like this one, coping quite well with the fact that some slaves are running up to four times slower. In this sense replication and speculative execution show quite effective to reduce the total time of the map/reduce job.

### **5.3 The Birth of the CESGA Hadoop on demand service**

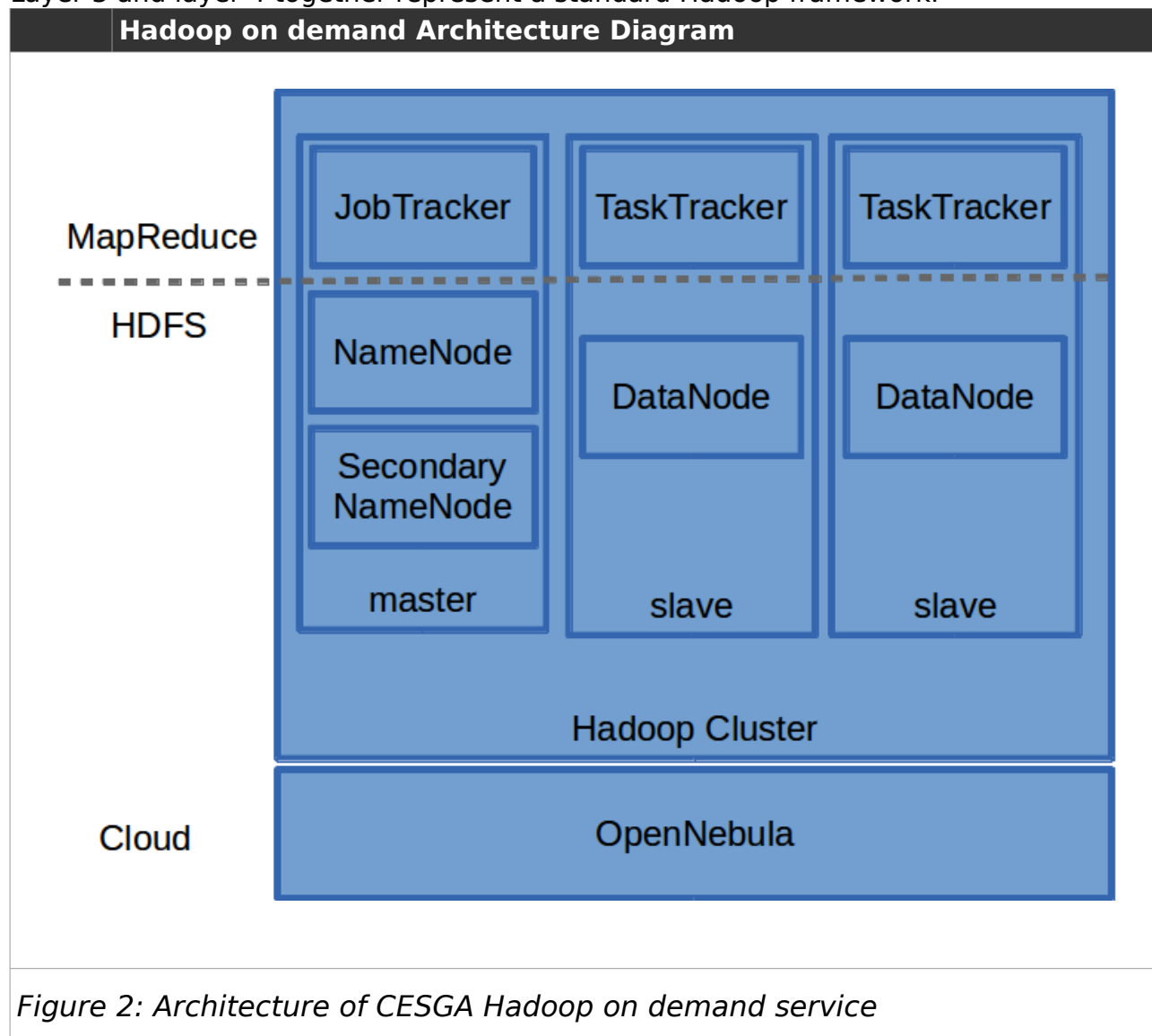
Taking advantage of the experience with GAIA we developed a framework that will allow other users to take advantage of BigData deploying their own Hadoop clusters on top of CESGA's laas platform.

The service was designed trying to simplify the usage for new users in a way that the underlying cloud layer is hidden for them. The architecture diagram is presented in Figure 2.

It represents an architecture with four layers:

- Layer 1: Cloud IaaS layer based on OpenNebula
- Layer 2: Hadoop cluster based on KVM virtual machines
- Layer 3: HDFS layer
- Layer 4: MapReduce layer

Layer 3 and layer 4 together represent a standard Hadoop framework.



A Hadoop cluster with  $N$  slaves can be started very easily by simply executing the following command:

```
hadoop-start -s N
```

The connection details for our new Hadoop cluster will appear in the screen once the cluster has started:

```
-----  
Configuration finished!"  
-----  
You can connect to your new Hadoop cluster through ssh:  
ssh hadoop@193.144.35.169  
You can monitor the status of the cluster in the following addresses:  
JobTracker Web Interface: http://193.144.35.169:50030/jobtracker.jsp  
NameNode Web Interface: http://193.144.35.169:50070/dfshealth.jsp  
-----  
In case of problems don't hesitate to contact our Systems' Department:  
Email: sistemas[at]cesga.es  
Phone: 981569810  
-----
```

It is also possible to connect to they new Hadoop cluster just executing:

```
hadoop-connect
```

You can check the status of your cluster using:

```
hadoop-status
```

Once they are finished, copy the results back and destroy the cluster freeing the resources:

```
hadoop-stop
```

We think that with these four simple commands every user can start using Hadoop on top of CESGA's cloud infrastructure.

At this point the Hadoop cluster is ready for use and you can use the standard Hadoop framework to upload your datasets and to run MapReduce jobs.

For example, you can upload through scp the dataset that you want to analyse to the hadoop master node and once there you can *put* it in HDFS:

```
scp mydataset hadoop@<hadoop-master>:
```

```
hadoop fs -put mydataset
```

Finally you can run your MapReduce job using the usual hadoop commands:

```
hadoop jar MyJob.jar MyJob dataset out
```

To check the status of your jobs you can connect to:

```
http://193.144.35.169:50030/jobtracker.jsp
```

For a brief introduction about how to use hadoop

```
http://hadoop.apache.org/docs/r1.1.2/commands_manual.html
```

## 6 Results

This section shows the results obtained by means of the learning of a SOM with HES spectra, using a Hadoop cluster, as it was described in the previous sections.

The obtained map can be visualized in different ways in order to look for interesting patterns in the data. For example, Figure 3 shows the number of hits received by each neuron in the map. Light areas correspond to neurons with a high number of hits, while dark areas correspond to neurons with a low number of hits.

Additionally, the U-Matrix, shown in Figure 4, shows the distances between each neuron and its immediate neighbours, such that a dark colour indicates that the neuron is distant to the neighbours, while a light colour indicates that the neuron is close to its neighbours.

Both figures are useful in order to unveil the multidimensional distribution of the data, in such a way that the researcher can easily find outliers and identify clusters of “normal” observations.

Using these maps as a guide, it is possible to visualize the neuron prototypes (weights) in order to navigate the dataset. For example, Figures 5 and 6 correspond to the prototypes of neurons located at (30,30) and (1,30). They seem to represent failures in the acquisition or in the digitalization of the spectra. On the other hand, figures 7 and 8 correspond to the prototypes of neurons located at (15,16) and (6,11), which can be identified as spectra of stars as they are expected in the HES survey.

Therefore, the SOM visualizations provide the researcher with the ability of distinguish between normal observations and outliers, and enables an easy exploration of the data, based on the visualization of the neuron prototypes.

In the future, expert astronomers will explore the whole SOM and extract knowledge from it, allowing them to understand which kind of astronomical objects are populating the HES survey and to discover novel ones.

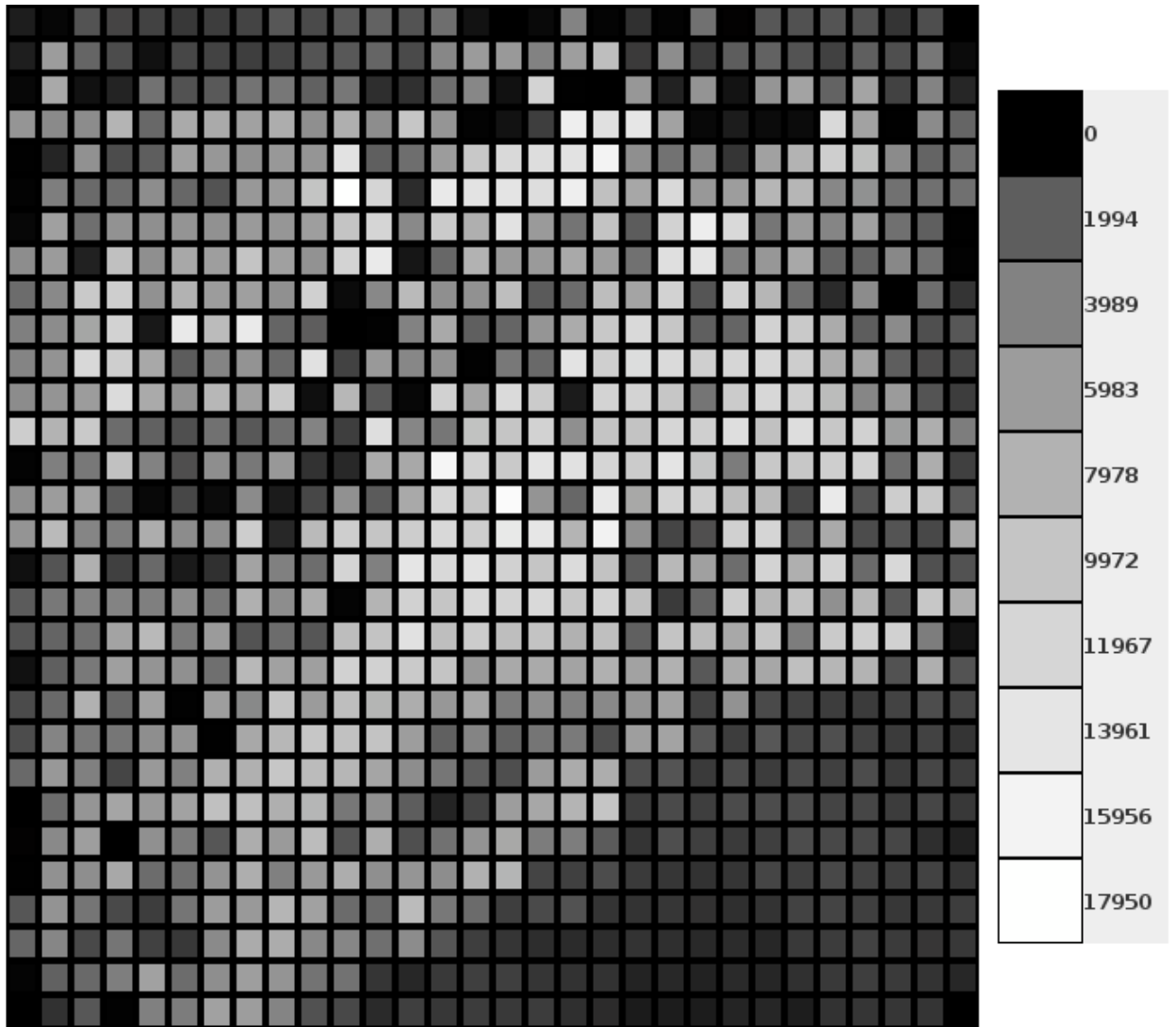


Figure 3: Hits per neuron in the SOM

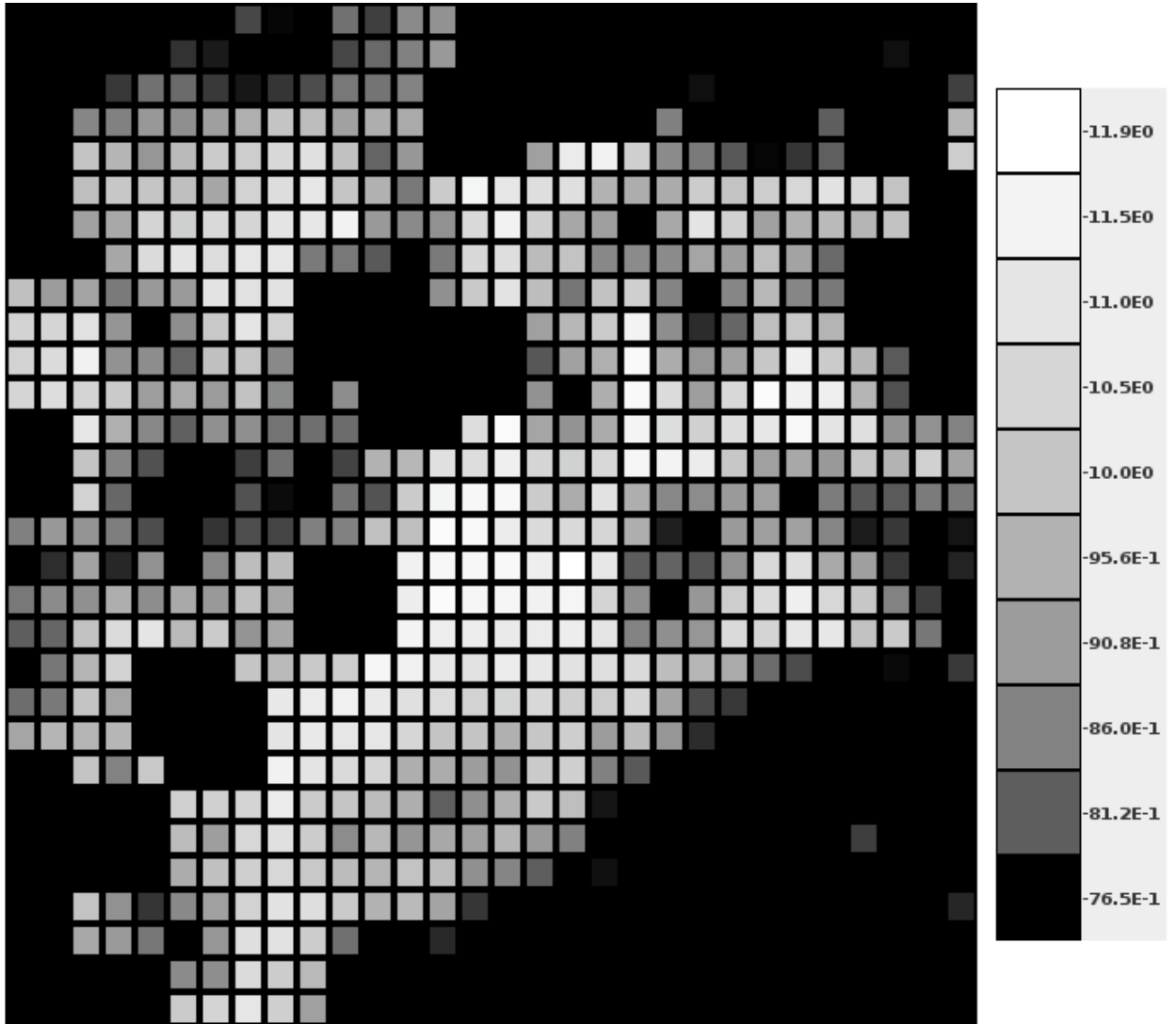


Figure 4: U-Matrix



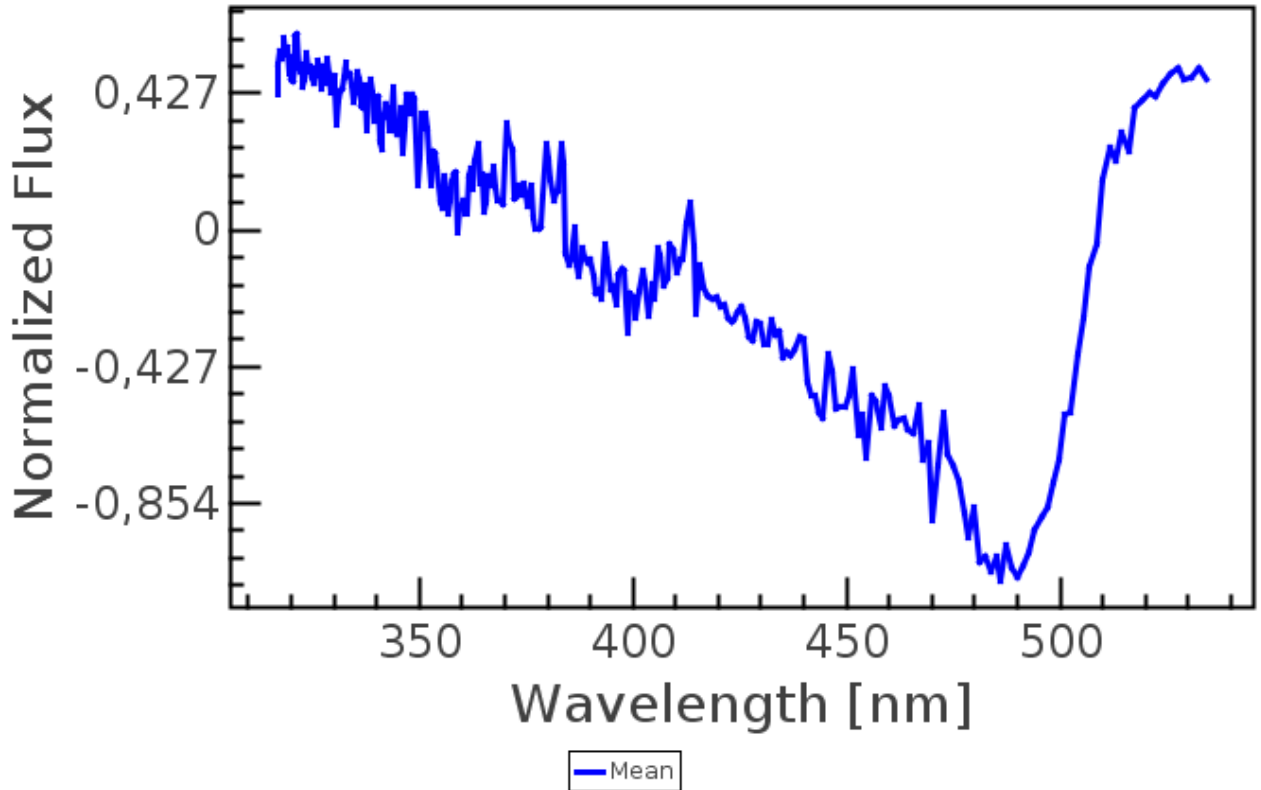


Figure 5: Neuron at (30,30)

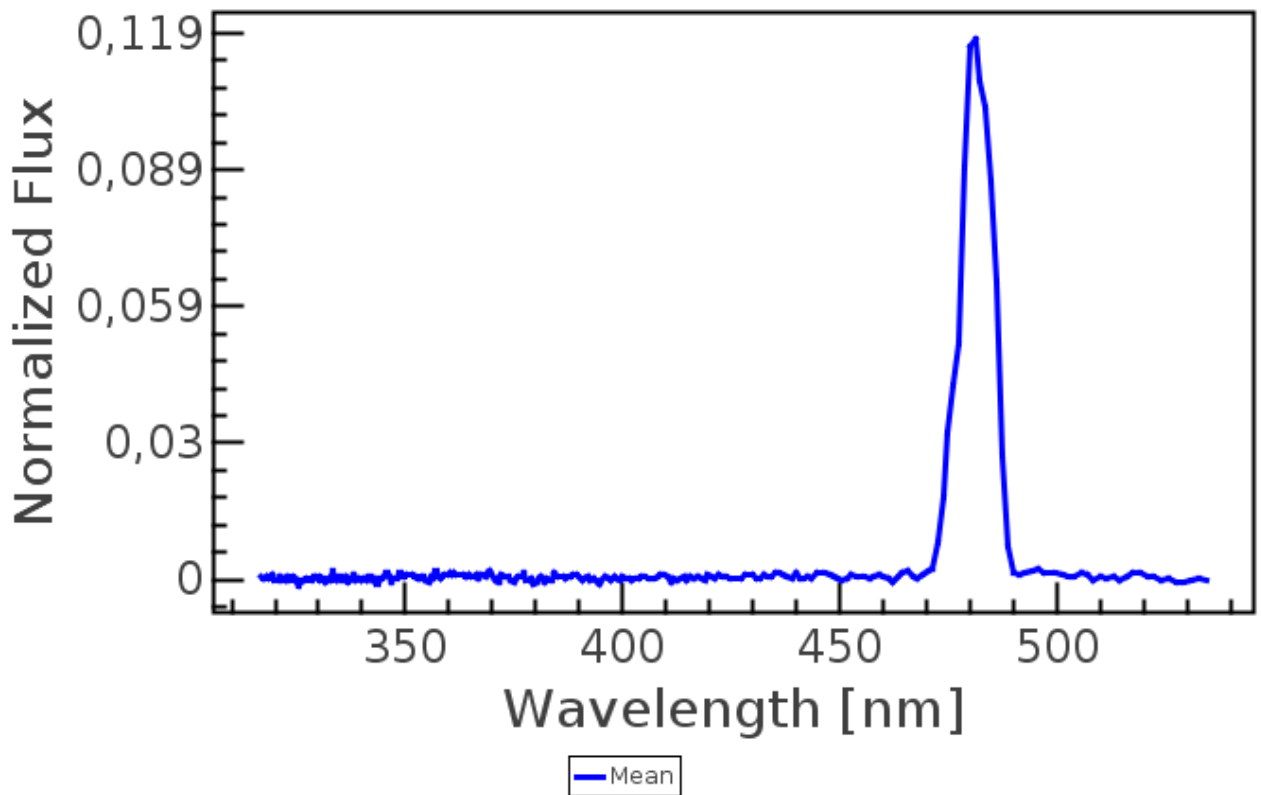


Figure 6: Neuron at (1,30)

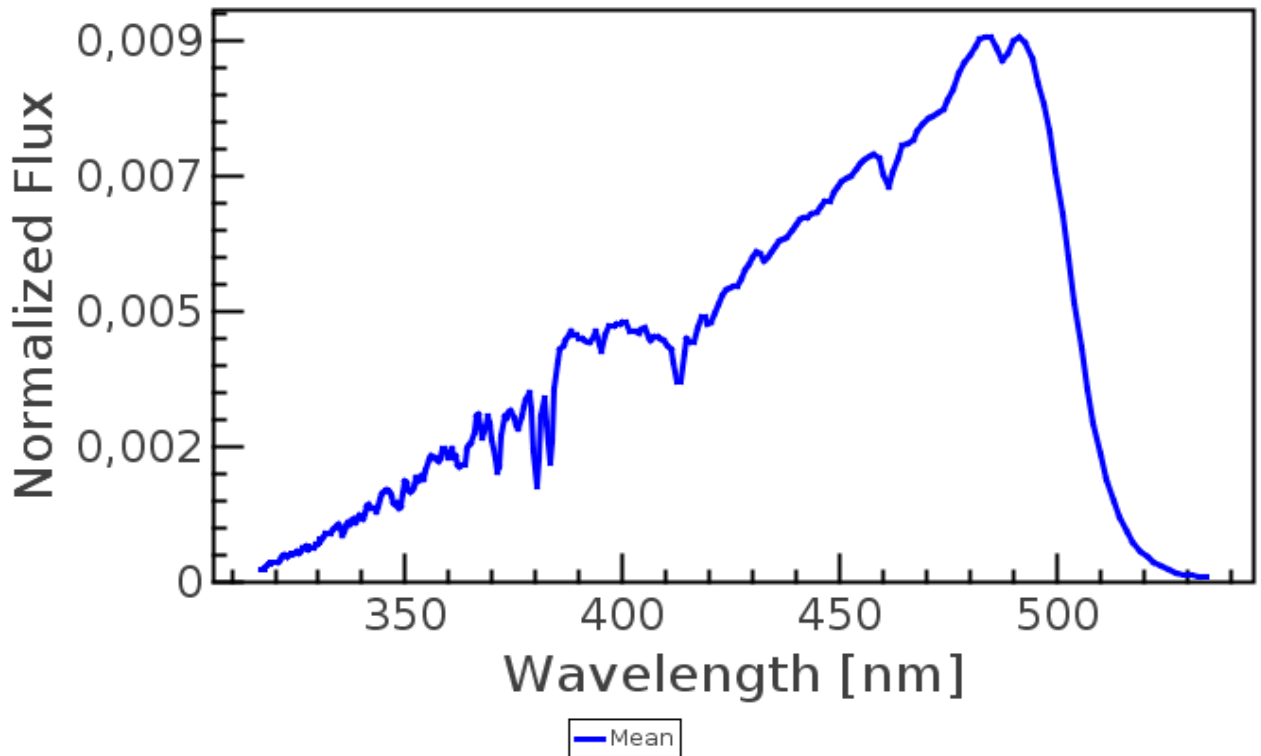


Figure 7: Neuron at (15,16)

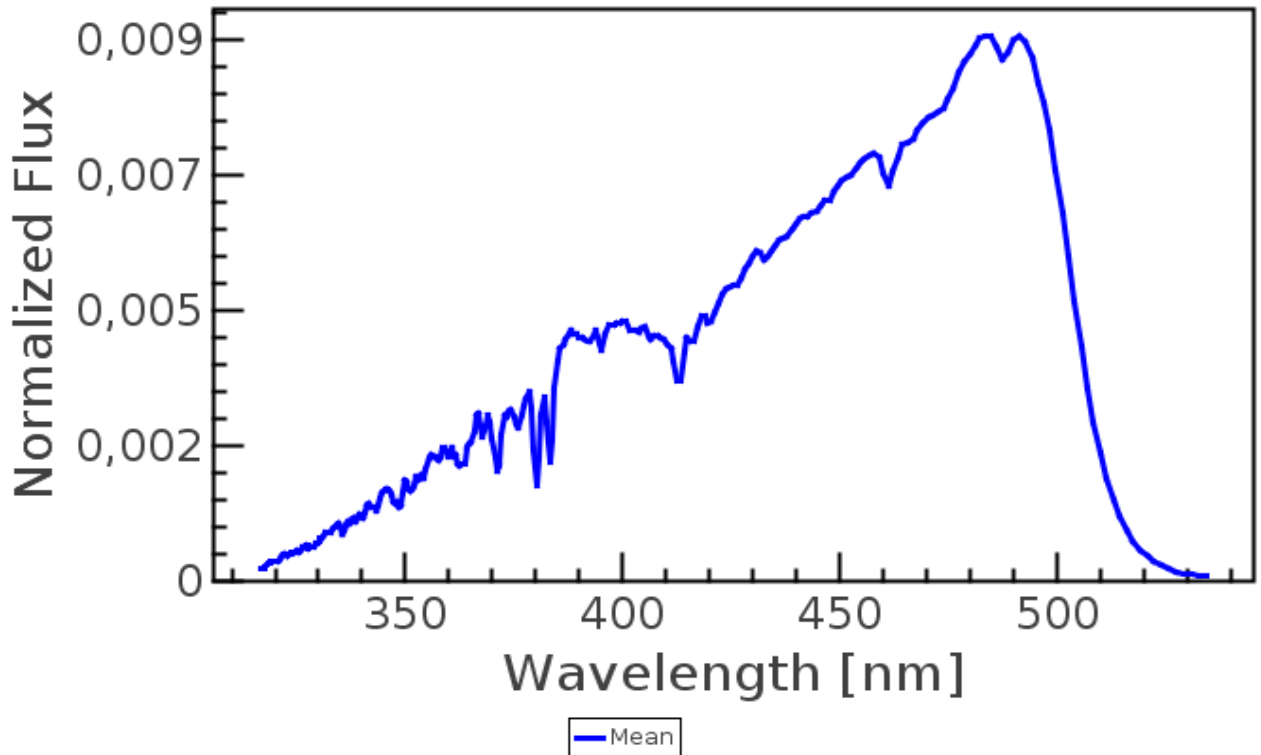


Figure 8: Neuron at (15,16)

## 7 Conclusions

Astronomy is becoming a data-intensive science. The forthcoming missions like Gaia will survey all sky, getting data with high precision for billions of stars. However, such an advance comes with a price, and it is the increase of difficulty in the data analysis.

At this work we have implemented a version of the SOM algorithm that allows to compute SOM maps that represent concisely millions of spectra from astronomical surveys. Such maps allow the researchers to extract knowledge from big datasets by facilitating the data navigation and exploration.

The Hadoop implementation of the algorithm makes it scalable for processing terabytes, or even petabytes of information. Hadoop is a great tool to distribute the computation of a SOM learning procedure, allowing to process enormous quantities of data. At this work, we have addressed the computation of a SOM that learns 5 million spectra from the HES survey with success. Therefore, the approach is promising for the processing of even larger datasets, as the one expected from Gaia.

When running Hadoop on a virtual environment it is important to optimize the virtual machines for maximum I/O performance, in general KVM is more tricky to optimize than Xen but in this technical report we showed which were the key aspects to modify in order to get the maximum performance from our OpenNebula/KVM cloud infrastructure.

After properly tuning the HDFS and MapReduce configuration, Hadoop manages quite well the execution on our heterogeneous cloud environment, coping quite well with the fact that some slaves are running up to four times slower. In this sense replication and speculative execution show quite effective to reduce the total time of the map/reduce job.

The new Big Data service will allow CESGA's users to take advantage of Big Data providing them a very simple way to have a Hadoop cluster up and running in just minutes.