



**Barcelona  
Supercomputing  
Center**  
*Centro Nacional de Supercomputación*

# Automating Big Data Benchmarking for Different Architectures with ALOJA

Nicolas Poggi, Postdoc Researcher



# Agenda

- 1. Intro on Hadoop performance
  - 1. Current scenario and problematic
- 2. ALOJA project
  - 1. Background
  - 2. Open source tools
- 3. Benchmarking
  - 1. Benchmarking workflow
  - 2. DEMO
- 4. Results
  - 1. HW and SW speedups
  - 2. Cost/Performance
  - 3. Scalability
- 5. Predictive Analytics and conclusions



# Hadoop design

## « Hadoop was designed to solve complex data

- Structured and non structured
- with [close to] linear scalability
- and application reliability

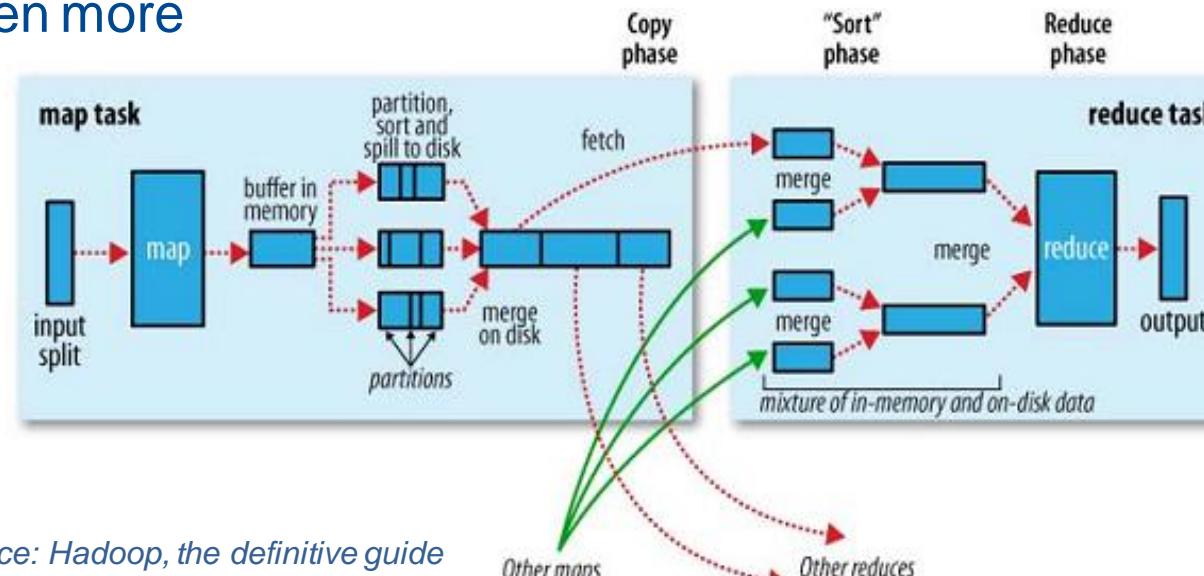


## « Simplifying the programming model

- From MPI, OpenMP, CUDA, ...

## « Operating as a blackbox for data analysts, but...

- Complex runtime for admins
- YARN abstracts even more



# Hadoop highly-scalable but...

« Not a high-performance solution!

« Requires

- Design,
  - Clusters, topology clusters
- Setup,
  - OS, Hadoop config
- Fine tuning required
  - Iterative approach
  - Time consuming



« and extensive benchmarking

# Setting up your Big Data system

## « Hadoop

- > 100+ tunable parameters
- obscure and interrelated
  - mapred.map/reduce.tasks.speculative.execution
  - *io.sort.mb* 100 (300)
  - *io.sort.record.percent* 5% (15%)
  - *io.sort.spill.percent* 80% (95 – 100%)
- Similar for Hive, Spark, HBase



## « Dominated by rules-of-thumb

- Number of containers in parallel:
  - 0.5 - 2 per CPU core

## « Large stack for tuning

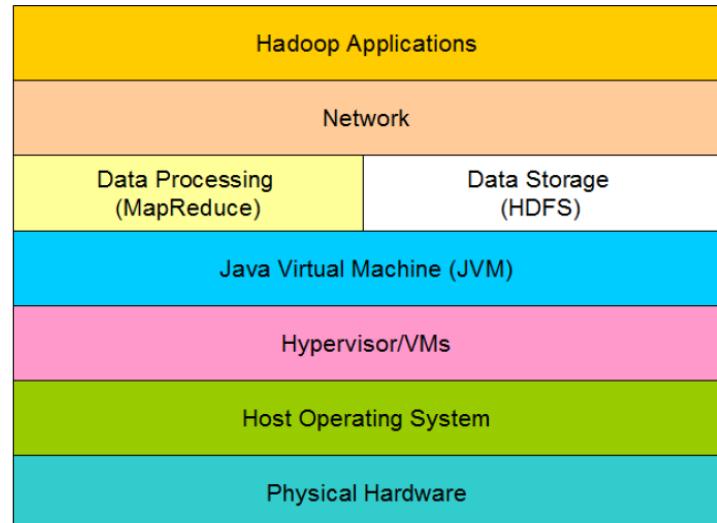


Image source: Intel® Distribution for Apache Hadoop

# How do I set my system, too many options!!!

« Default values in Apache source not ideal

« Large and spread eco system

- Different distributions
- Product claims

« Each job is different

- No one-fits-all solution

« Cloud vs. On-premise

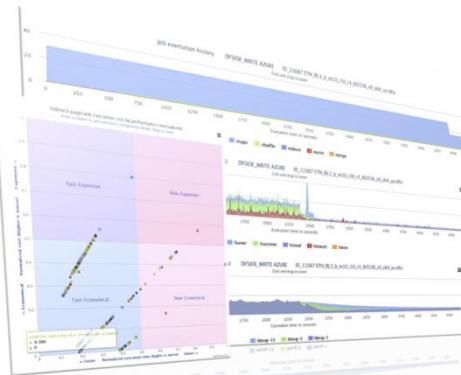
- IaaS
  - Tens of different VMs to choose
- PaaS
  - HDInsight, CloudBigData, EMR

« New economic HW

- SSDs, InfiniBand Networking



# BSC's project ALOJA: towards cost-effective Big Data



<http://aloja.bsc.es>

- « Open research project for **improving the cost-effectiveness** of Big Data deployments

## « Benchmarking and Analysis tools



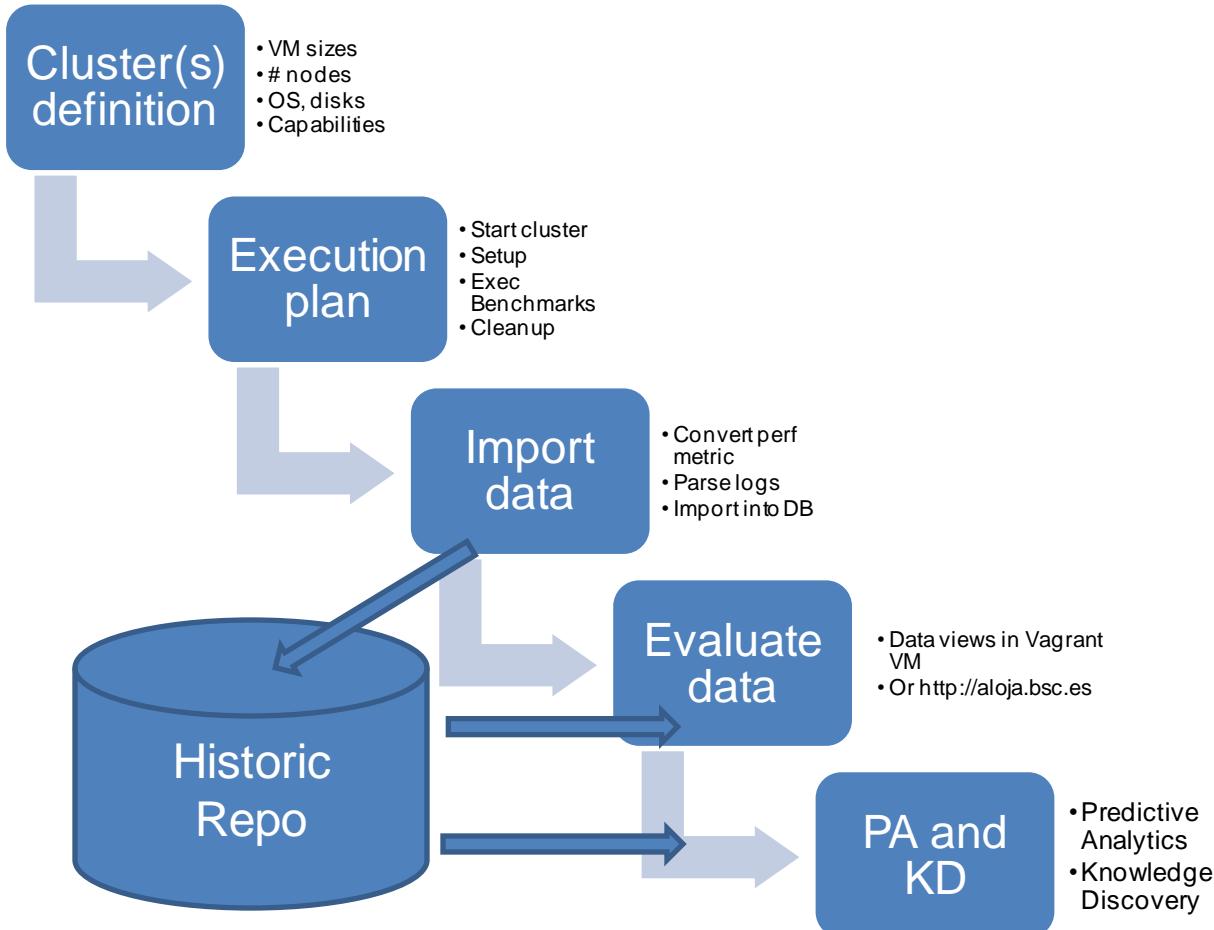
## « Online repository and largest Big Data repo

- **50,000+** runs of HiBench, TPC-H, and [some] BigBench
- Over **100 HW configurations** tested
  - Of different Node/VM, disks, and networks
  - Cloud: Multi-cloud provider including both IaaS and PaaS
  - On-premise: High-end, HPC, commodity, low-power

## « Community

- Collaborations with industry and Academia
- Presented in different conferences and workshops
- Visibility: 47 different countries

# Workflow in ALOJA



# Challenges (circa end 2013)

## « Test different clusters and architectures

- On-premise and HPC
  - Commodity, high-end, appliance, low-power (ARM)
- Cloud IaaS
  - 32 different VMs in Azure, similar in other providers
- Cloud PaaS
  - HDInsight (Windows and Linux), EMR, CloudBigData

## « Different access level

- Full admin, user-only, request-to-install, everything ready, queuing systems (SGE)

## « Different versions

- Hadoop, JVM, Spark, Hive, etc...
- Other benchmarks

## « Problems

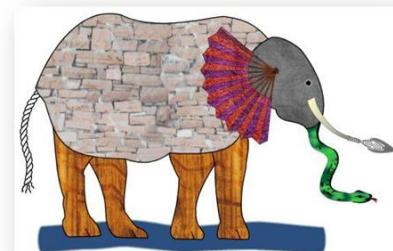
- All systems though for PROD
  - Not for comparison
- No Azure support
- Many different packages
- No one-fits-all solution

## « Dev environments and testing

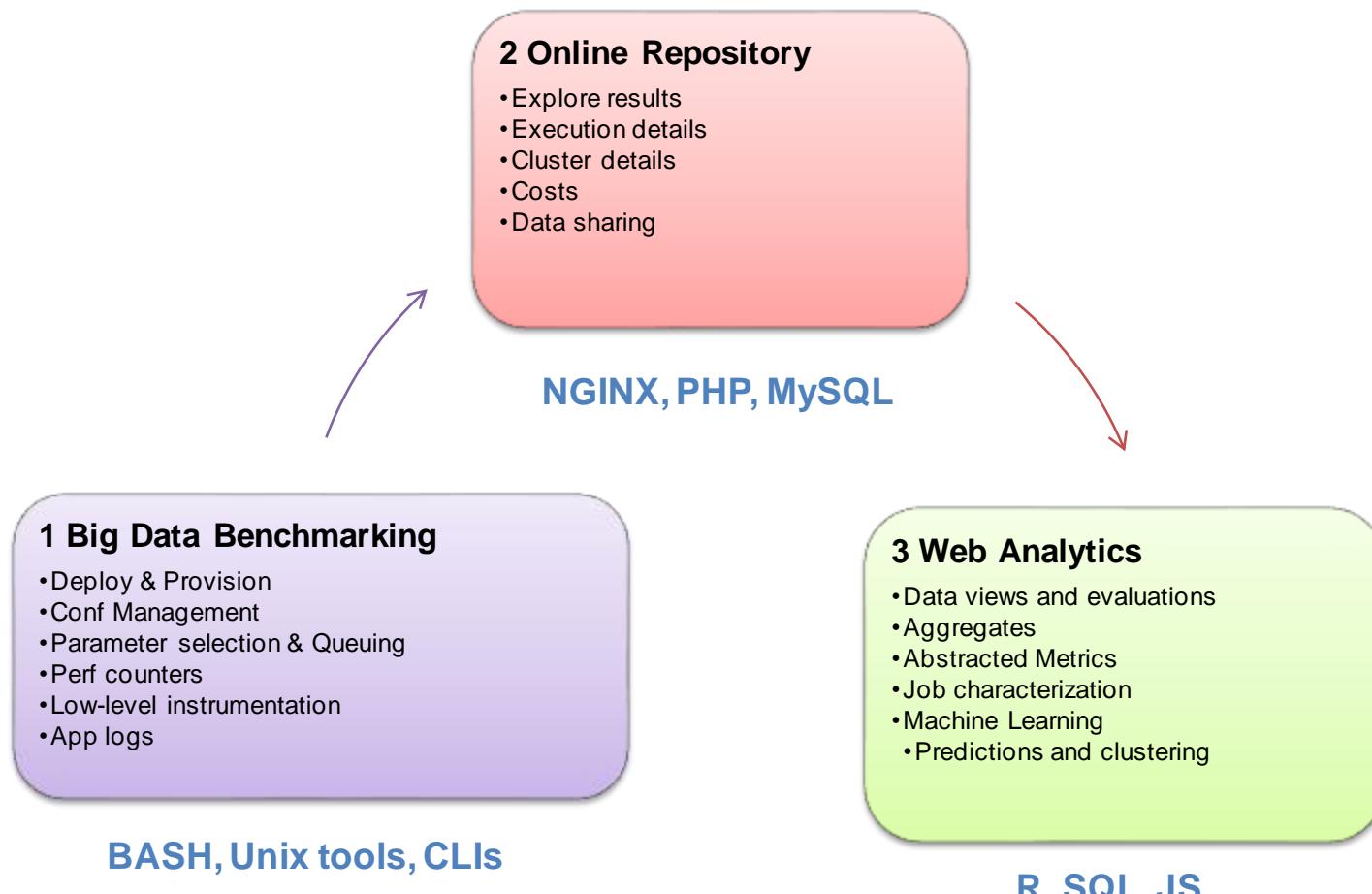
- Big Data usually requires a cluster to develop and test

## « Solution

- Custom implementation
  - Abstracting differences
- Based in simple components
- Wrapping commands



# ALOJA Platform main components



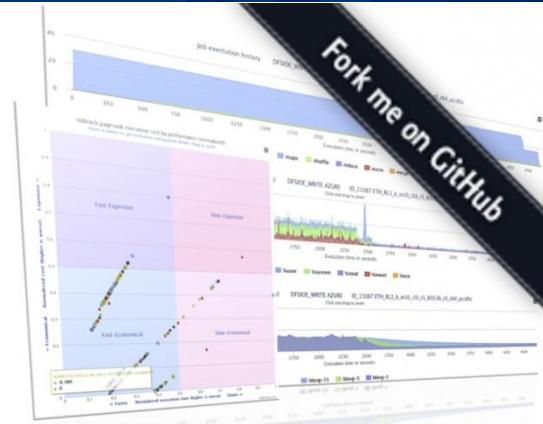
# Extending and collaborating in ALOJA

## « Setting up a DEV environment:

- « Installs a Web Server with sample data
- « Sets a local cluster to test benchmarking

1. Install prerequisites
  - git, vagrant, VirtualBox
2. git clone <https://github.com/Aloja/aloja.git>
3. cd aloja
4. vagrant up
5. Open your browser at: <http://localhost:8080>
6. Optional start the benchmarking cluster

```
vagrant up ./*/
```



# Commands and providers

## Provisioning commands

### « Connect

- Node and Cluster
- Builds SSH cmd line
  - SSH proxies

### « Deploy

- Creates a cluster
- Sets SSH credentials
- If created, updates config as needed
- If stopped, starts nodes

### « Start, Stop

### « Delete

### « Queue jobs to clusters

## Providers

### « On-premise and HPC

- Custom settings for clusters
  - Multiple disk types
  - Different architectures
  - Resource/Job control

### « Cloud IaaS

- Azure, OpenStack, Rackspace, AWS

### « Cloud PaaS

- HDInsight, Cloud Big Data, EMR soon

# Running benchmarks in ALOJA

## « Benchmarking with defaults:

`/repo_location/aloja-bench/run_benchs.sh`

```
ALOJA-BENCH, script to run benchmarks and collect results
Usage:
/vagrant/aloja-bench/run_benchs.sh [-C clusterName <uses aloja_cluster.conf if present or not specified>]
[-n net <IB|ETH>]
[-d disk <SSD|HDD|RL{1,2,3}|R{1,2,3}>]
[-b "benchmark suite" <Big-Bench Hadoop-Examples Hecuba-WordCount HiBench2-1TB HiBench2-min HiBench2 HiBench3HDI HiBench3-min HiBench3 sleep TPCH-hive>]
[-r replicaton <positive int>]
[-m max mappers and reducers <positive int>]
[-i io factor <positive int>] [-p port prefix <3|4|5>]
[-I io.file <positive int>]
[-l list of benchmarks <space separated string>]
[-c compression <0 (disabled)|1|2|3>]
[-z <block size in bytes>]
[-s (save prepare)]
[-N (don't delete files)]
[-t execution type (e.g: default, experimental)]
[-e extrae (instrument execution)]
```

example: `/vagrant/aloja-bench/run_benchs.sh -C vagrant-99 -n ETH -d HDD -r 1 -m 12 -i 10 -p 3 -b HiBench2-min -I 4096 -l wordcount -c 1`

## « To queue jobs:

`/repo_location/shell/exeq.sh`

# ALOJA-WEB

- « Entry point for explore the results collected from the executions,
  - Provides insights on the obtained results through continuously evolving data views.
- « Online **DEMO** at: <http://aloja.bsc.es>

**HiBench Executions on Hadoop**

**BSC - Microsoft Research Centre**

Navigation: [HiBench Runs Details](#) [Hadoop Job Counters Charts](#) [Cost Evaluation](#) [Performance](#)

Click on a **benchmark name** to see execution details.  
Select different rows and **click compare**, to compare charts.  
Search to filter results. Shift+Click to order by multiple columns

Show **10** entries Show / hide columns

ID	Benchmark	Exe Time	Running Cost \$	Net	Disk	Maps	IO SFac	Rep	IO FBuf	Comp	Blk size	Cluster	Files	PARAVER	
<input type="checkbox"/>	20372	dfsioe_read	2990	5.81	ETH	RL1	8	10	2	65536	3	32	Azure L	<a href="#">files</a>	<a href="#">PRV .ZIP</a>
<input checked="" type="checkbox"/>	20371	pagerank	2809	5.46	ETH	RL1	8	10	2	65536	3	32	Azure L	<a href="#">files</a>	<a href="#">PRV .ZIP</a>
<input type="checkbox"/>	20370	sort	657	1.28	ETH	RL1	8	10	2	65536	3	32	Azure L	<a href="#">files</a>	<a href="#">PRV .ZIP</a>
<input type="checkbox"/>	20369	wordcount	1336	2.60	ETH	RL1	8	10	2	65536	3	32	Azure L	<a href="#">files</a>	<a href="#">PRV .ZIP</a>
<input type="checkbox"/>	20368	kmeans	3002	5.84	ETH	RL1	8	10	2	65536	3	32	Azure L	<a href="#">files</a>	<a href="#">PRV .ZIP</a>
<input type="checkbox"/>	20367	dfsioe_write	2640	5.13	ETH	RL1	8	10	2	65536	3	64	Azure L	<a href="#">files</a>	<a href="#">PRV .ZIP</a>
<input type="checkbox"/>	20366	dfsioe_read	3139	6.10	ETH	RL1	8	10	2	65536	3	64	Azure L	<a href="#">files</a>	<a href="#">PRV .ZIP</a>
<input checked="" type="checkbox"/>	20365	pagerank	2612	5.08	ETH	RL1	8	10	2	65536	3	64	Azure L	<a href="#">files</a>	<a href="#">PRV .ZIP</a>
<input type="checkbox"/>	20364	sort	627	1.22	ETH	RL1	8	10	2	65536	3	64	Azure L	<a href="#">files</a>	<a href="#">PRV .ZIP</a>
<input type="checkbox"/>	20363	wordcount	1279	2.49	ETH	RL1	8	10	2	65536	3	64	Azure L	<a href="#">files</a>	<a href="#">PRV .ZIP</a>

Showing 1 to 10 of 3,696 entries (filtered from 4,019 total entries) First Previous 1 2 3 4 5 Next Last

Copy CSV Excel PDF Print

Compare executions:

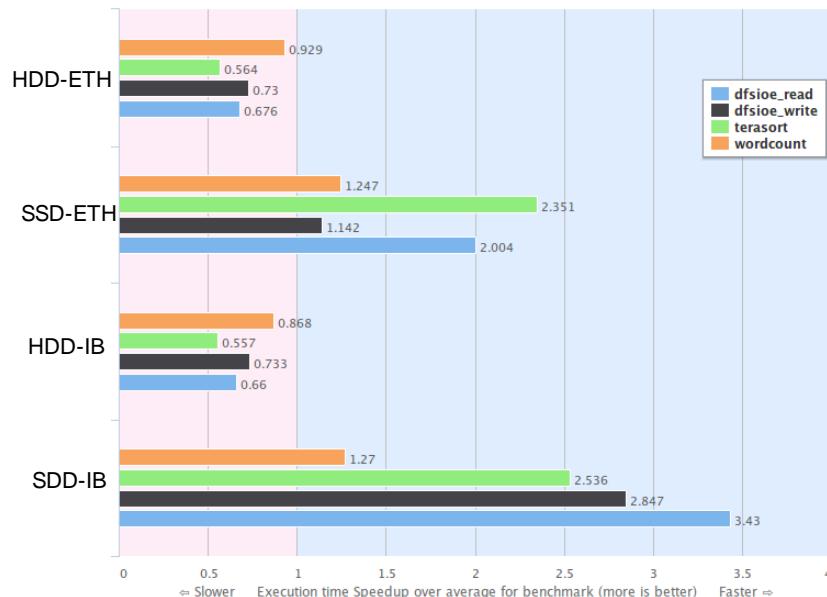
Select rows by clicking on checkboxes and click: [Compare Executions](#)



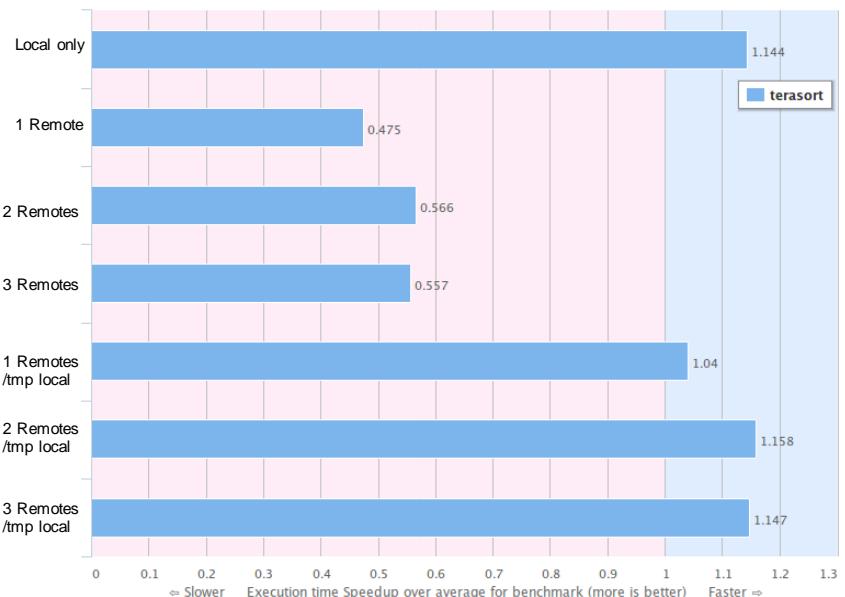
The chart displays the job execution history for the PAGERANK AZURE benchmark. It shows the progression of tasks over time, with different colors representing different phases: map (blue), shuffle (green), reduce (purple), waste (red), and merge (orange). The x-axis represents time in seconds, ranging from 0 to 2750. The y-axis represents the number of tasks or progress, ranging from 0 to 20. The chart shows several distinct phases of execution, with the reduce phase being the most prominent.

# Impact of HW configurations in Speedup

## Disks and Network



## Cloud remote volumes



Speedup (higher is better)

# Clusters by cost-effectiveness



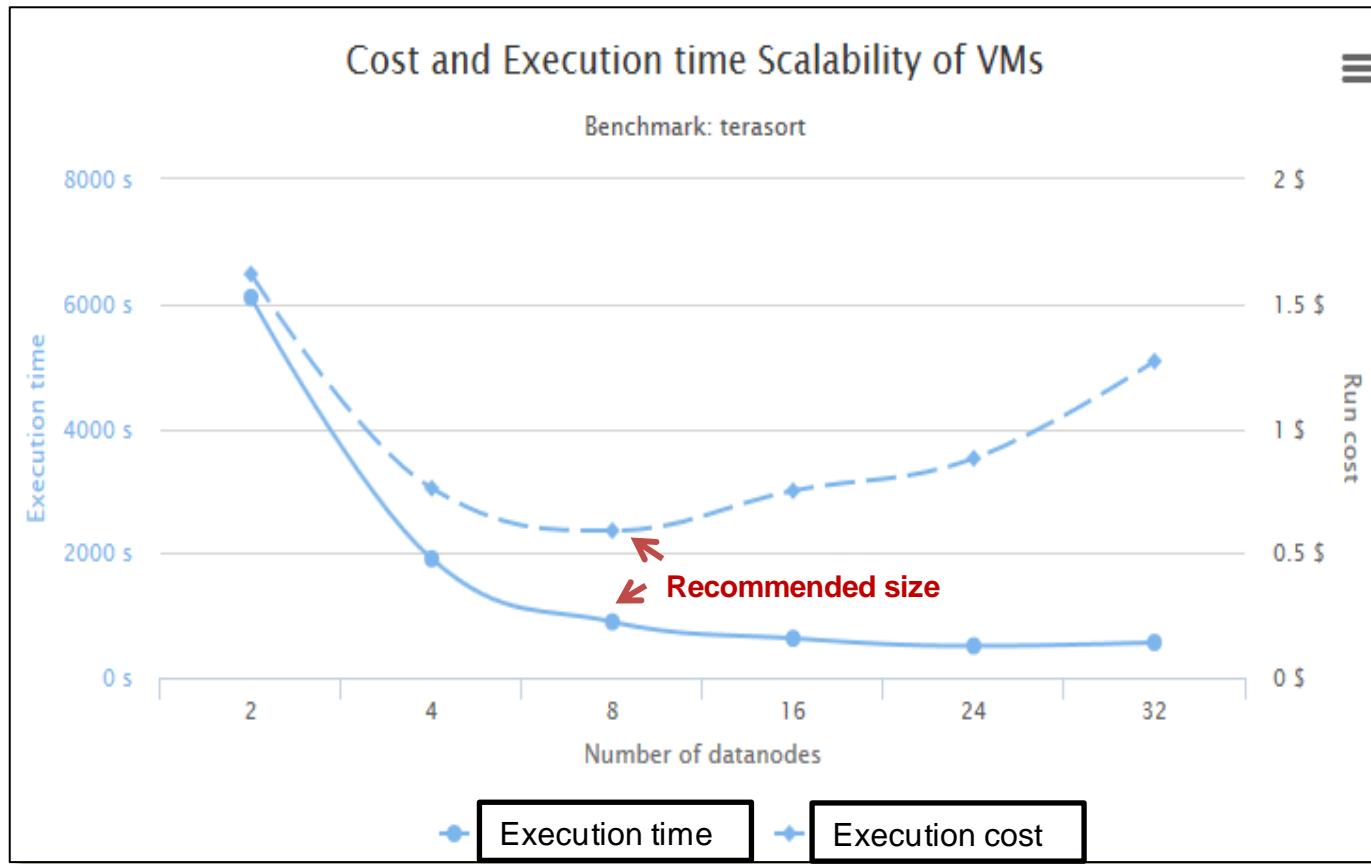
5 clusters ordered by cost-effectiveness

Rank	Cluster	Best execution cost	Best execution time	VM RAM	Datanodes	VM OS	Provider	Type
1	rl-06	0.30 US\$	403 s	8 GB	8	linux	rackspace	IaaS
2	rl-16	0.44 US\$	561 s	8 GB	8	linux	rackspace	IaaS
3	rl-19	0.66 US\$	458 s	15 GB	8	linux	rackspace	IaaS
4	rl-33	0.72 US\$	259 s	30 GB	8	linux	rackspace	IaaS
5	rl-30	0.95 US\$	336 s	30 GB	8	linux	rackspace	IaaS

Cheapest exec

Fastest Exec

# Cost/Performance Scalability of cluster size



- X axis number of data nodes (cluster size)
- Left Y Execution time (lower is better)
- Right Y Execution cost (lower is better)

**ALOJA, Hadoop Benchmark Repository and Performance Analysis Tools**

Home Benchmark Repository Config Evaluations Cost/Perf Evaluation Performance Details Prediction Tools ▾

Welcome to the **ALOJA** project,  
**ALOJA** is an initiative of the **BSC-MSR** research centre in Barcelona to explore Hadoop's potential. You can find introductory [Slides](#) and [Papers](#) in the ALOJA Reference menu.

This site is under constant development and it is in the process of being documented. For more information about the project, please browse the site, the [code](#), and send inquiries, feature requests or bug reports to: [hadoop@aloha-project.org](mailto:hadoop@aloha-project.org).

If you're curious about the name of the project, visit [ALOJA](#).

**Site's content:**

Section	Description
<a href="#">Video DEMO of ALOJA</a>	Brief video showcasing ALOJA's main online features (a bit outdated).
<a href="#">Benchmark Executions</a>	This section presents the benchmark execution repository. It features more than 30,000 executions and counting. This tool allows you to browse, filter, search, and select distinct executions to compare and analyse its execution details.
<a href="#">Hadoop Job Counters</a>	The Hadoop Job Counters sections allows to browse the counters output at each of the Hadoop executions, filter them, and to order by a specific counter the selected runs (or all). This section presents the summary of all the Job execution counters, Map and Reduce specific counters, and the I/O subsystem counters. It also features the details by task: to understand the running time of each Map or Reduce process.

**Prediction Tools ▾**

- Modeling Data
- Collapse Data
- Predict Configurations
- Parameter Evaluation (ML)
- Anomaly Detection
- Minimal Configurations
- Variable Crossing
- Variable Crossing (vs Exec.Time)
- Variable Crossing (vs Prediction)
- Data Summary

**ALOJA Reference ▾**

**Blog** **BSC-MSR Centre**

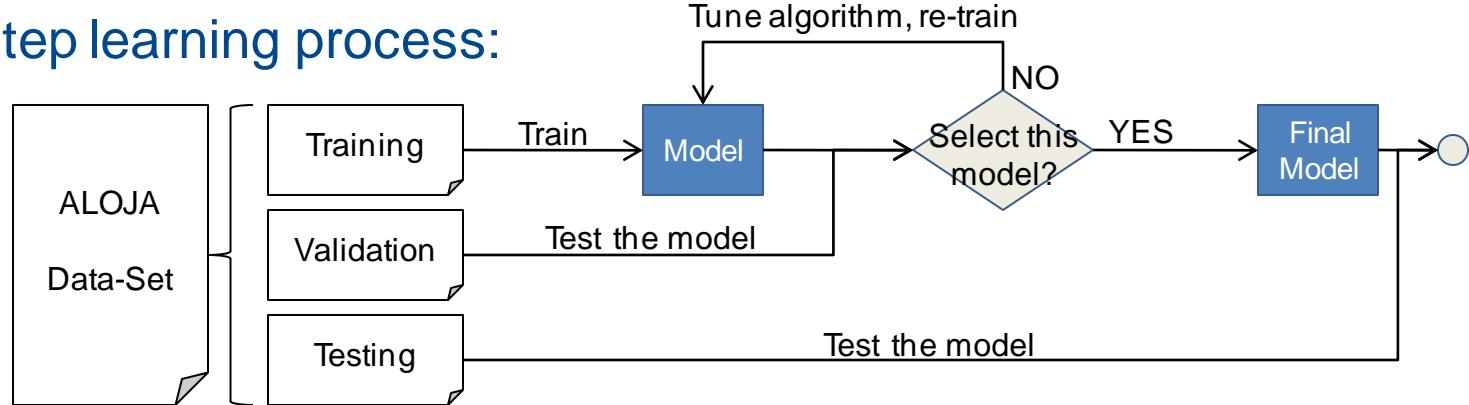
**Fork me on GitHub**

# Predictive Analytics and automated learning

# Modeling and predicting Hadoop time

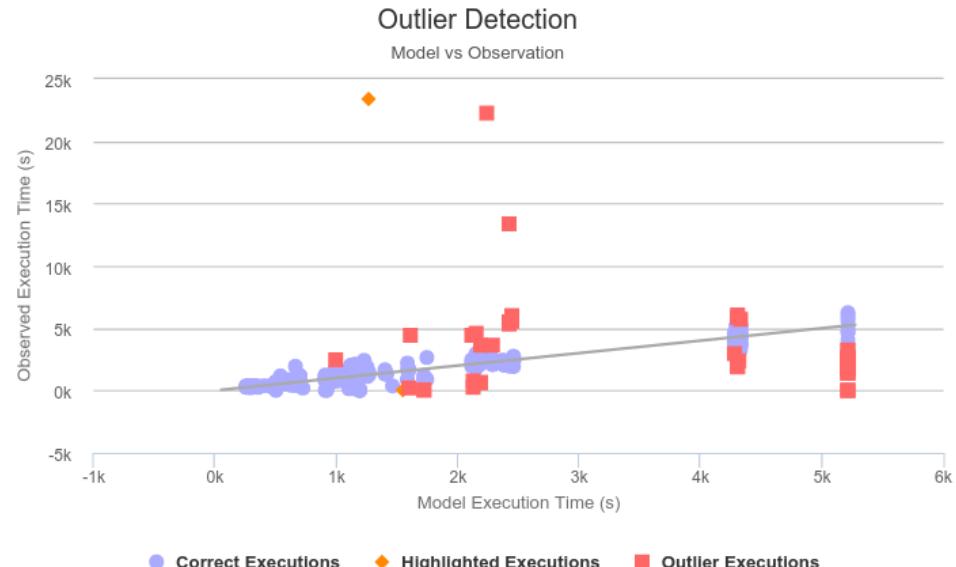
## Methodology

- 3-step learning process:



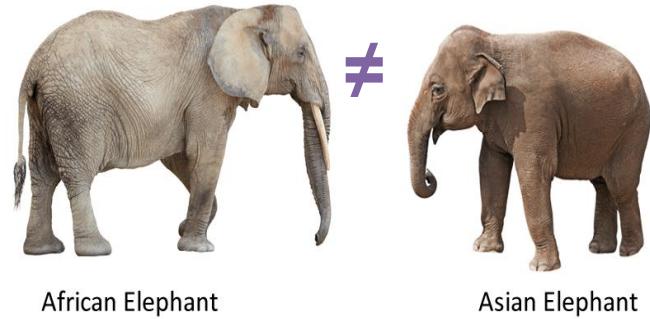
## Use cases

- Anomaly detection
- Predict best configurations
- Guided benchmarking
- Knowledge Discovery



# Concluding remarks

- « In ALOJA we are benchmarking from
  - Low-powered to cloud and super computers
  - Testing both HW components and SW configs
- « Each system has it's own peculiarities
  - ...**and failures!**
  - Different access levels
  - Sharing
    - Public cloud very difficult to measure correctly!
  - Versions of software
- « Benchmarking its fun!, or at least...
  - It will save you €€€ and allow you to scale
- « But it is also **tough**
  - The industry needs more transparency, We still have a lot to do...
- « In ALOJA we provide the benchmarking scripts
  - And also de results, that should be your first entry point
  - We are adding constantly new features
    - Benchmarks, systems providers
- « It is an open initiative, you're invited to participate



# More info:

## « ALOJA Benchmarking platform and online repository

- <http://aloja.bsc.es> <http://aloja.bsc.es/publications>

## « Benchmarking Big Data

- [http://www.slideshare.net/ni\\_po/benchmarking-hadoop](http://www.slideshare.net/ni_po/benchmarking-hadoop)

## « BDOOP meetup group in Barcelona



## « Big Data Benchmarking Community (BDBC) mailing list

- (~200 members from ~80 organizations)
- <http://clds.sdsc.edu/bdbc/community>

## « Workshop Big Data Benchmarking (WBDB)

- Next: <http://clds.sdsc.edu/wbdb2015.ca>

## « SPEC Research Big Data working group

- <http://research.spec.org/working-groups/big-data-working-group.html>

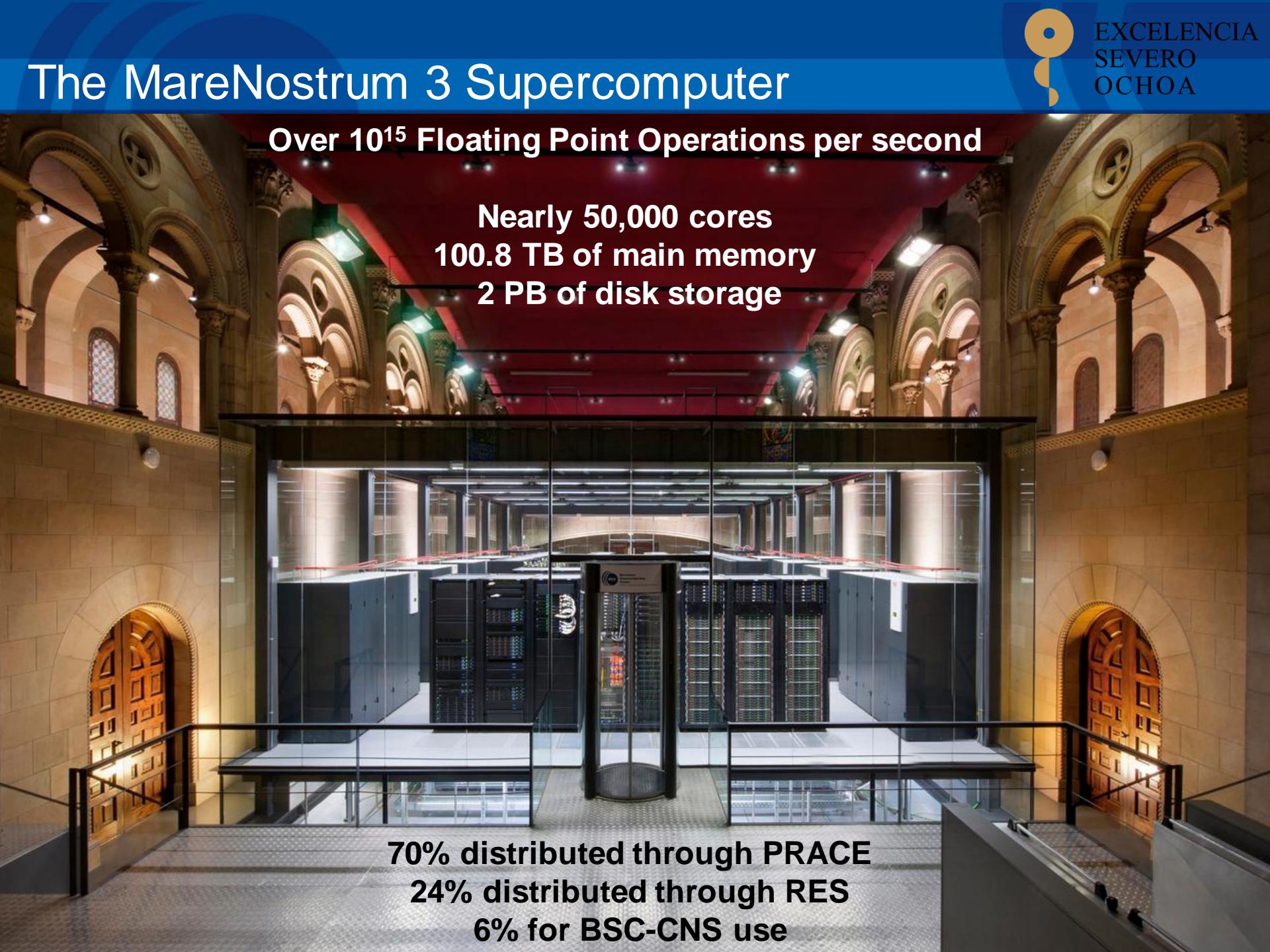


## « Slides and video:

- Michael Frank on Big Data benchmarking
  - <http://www.tele-task.de/archive/podcast/20430/>
- Tilmann Rabl Big Data Benchmarking Tutorial
  - [http://www.slideshare.net/tilmann\\_rabl/ieee2014-tutorialbarurabl](http://www.slideshare.net/tilmann_rabl/ieee2014-tutorialbarurabl)

# The MareNostrum 3 Supercomputer

Over  $10^{15}$  Floating Point Operations per second



Nearly 50,000 cores  
100.8 TB of main memory  
2 PB of disk storage

70% distributed through PRACE  
24% distributed through RES  
6% for BSC-CNS use



**Barcelona  
Supercomputing  
Center**  
*Centro Nacional de Supercomputación*

Thanks!

Q&A

Contact: [nicolas.poggi@bsc.es](mailto:nicolas.poggi@bsc.es)