HTCondor @ IAC

Instituto de Astrofísica de Canarias

Tenerife - La Palma, Canary Islands. Spain

Antonio Dorta (adorta@iac.es)

Outline

1. Introduction to **HTCondor**
2. HTCondor @ **IAC**: our problems & solutions
3. **`ConGUSTo`**, our monitoring tool

IAC

The “Instituto de Astrofísica de Canarias (IAC)” is an internationalized Spanish research centre.

It has headquarters and observatories in two islands (Tenerife and La Palma), with over 25 different telescopes, including the biggest optical-infrared telescope in the world at the present time. They are set in an environment of excellent astronomical quality, constituting the European Northern Observatory (ENO)

Condor (HTCondor)

- **HTC**ondor**: HTC (High Throughput Computing)** vs **HPC** (High Performance Computing): “FlopM”, “FlopY” vs Flops
- HTCondor: **open source** (Apache License, Version 2.0) specialized workload management system for compute-intensive jobs (“***job scheduler***”) developed by the University of Wisconsin-Madison, USA
- Institutions using HTCondor: NASA, CERN, Dreamworks, FermiLab, many Universities and research centers all around the world.
- It can be installed on different OS, hardware, etc: linux, MS Windows, OS X, Solaris… 32/64 bits…

Condor (HTCondor)

- It makes possible to execute codes in a wide set of different and heterogeneous resources: local PCs, Clusters, Supercomputers, Grids, **Clouds**, etc.

- **HTCondor-G:** Mostly for **Grid** and Cloud: pre-WS and WS Globus, Nordugrid ARC, UNICORE and Amazon EC2 (also with other batch systems, like Torque/PBS and LSF). Ex.: LHC Experiment at CERN

- Resources are divided per **slot** (cores): memory, disk, etc. Each slot can execute a job submitted by other users

Condor (HTCondor)

- Advantages also in normal “computers”, like better use of our resources: it makes possible to run your applications in other computers when they are **idle** (also hibernate/suspend and WOL)
- Most programs does not need any modification (or very trivial ones)
- **Queue system**: submit file specifying program, arguments, inputs and outputs… and wait for results!
- New versions 8.4.x (Sept 2015) have simplified the syntax with new and powerful commands

Condor: How does it work?

- Pool of slots:

- Job submission. Queue of jobs:

- Matching (according to the requirements, status, *universes*):
  classAds: slots (~150 attr.) | jobs (~100 attr.)

- Copy: exec+inputs | remote execution | copy “back”: results

Installation and configuration

- Download from http://research.cs.wisc.edu/htcondor/
  - Odd: Development Release (last 8.**5**.0, Oct 2015)
  - Even: Current Stable Release (last 8.**4**.2, Nov 2015)
  - Source code, native packages, **stripped**/unstripped **tarballs**

- **Highly configurable:**
  - There is a **knob** almost for everything
  - v7.x: Huge config file (~ 3000 lines)
  - v8.x: Precompiled defaults, small config file (~100 lines)
  - **Specific** setup: Auth method, security, paths and domains, policies, …
  - Machine **roles**: **Central Manager** (collector & negotiator), **Submit**, **Execute**, Checkpoint server. **Daemons** → total: 20+, in use: 5-7
  - Common config file and specific ones depending on roles

HTCondor: Useful shell commands

Total: ~ 100 shell commands, but users mostly use a few of them (3 - 10)

- **condor_status** (-avail, -run, -master, -server, -state, -total, -sort, -long, -constraint, -submitters, -format, ...)
- **condor_submit** <submit_file> (-append, ...)
- **condor_q** (-global, -hold, -analyze, -run, ...)
- condor_tail, condor_ssh_to_job
- condor_hold, condor_release, condor_rm
- condor_history (-userlog, -long, -constraint, …)
- condor_version, condor_userprio, condor_logview, condor_prio, ...

HTCondor: Useful shell commands

- **condor_status**

Name	OpSys	Arch	**STATE**	**ACTIVITY**	LoadAv	Mem	ActvtyTime
slot1@machine1	LINUX	X86_64	**Owner**	Idle	0.050	1976	0+00:10:04
slot2@machine2	LINUX	X86_64	**Unclaimed**	Idle	0.100	2132	0+00:11:14
slot4@machine3	LINUX	X86_64	**Claimed**	**Busy**	0.050	4022	0+00:05:51
slot5@machine3	LINUX	X86_64	**Claimed**	**Suspended**	0.300	4022	0+00:15:35

	Total	Owner	Claimed	Unclaimed	Matched	Preempting	Backfill
Total	770	307	397	66	0	0	0

- **condor_q** **(jobID: Cluster.Process)**

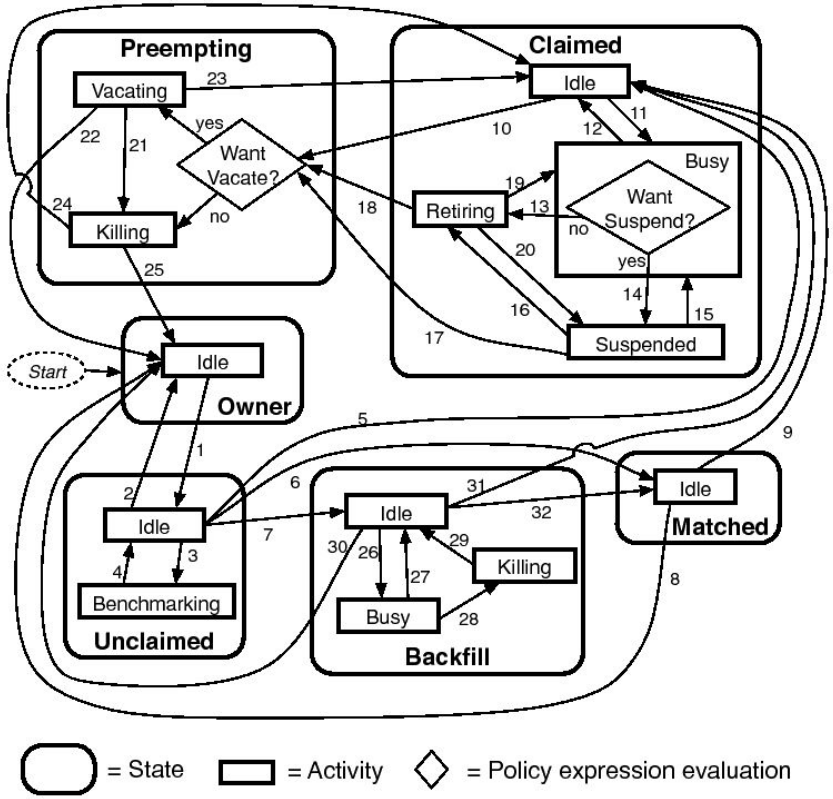
```
-- Schedd: machine.ll.iac.es : <161.72.216.99:9618?sock=20661_61b6_4>
```

ID	OWNER	SUBMITTED	RUN_TIME	**ST**	PRI	SIZE	CMD
418.0	jsmith	3/13 17:00	0+00:37:32	**I**	0	317.4	myprogram -c 0
418.**1**	jsmith	3/13 17:00	0+00:30:25	**<**	0	488.3	myprogram -c 2
418.**2**	jsmith	3/13 17:00	0+00:31:10	**R**	0	231.4	myprogram -c 4
418.**3**	jsmith	3/13 17:00	0+00:23:25	**S**	0	134.4	myprogram -c 6
418.**4**	jsmith	3/13 17:00	0+00:42:17	**>**	0	623.1	myprogram -c 8
418.**5**	jsmith	3/13 17:00	0+00:26:52	**H**	0	432.6	myprogram -c 10

I:idle **<**: transferring input **R**: running **S**: suspended
>: transferring output **H**: on hold **C**: completed **X**: removed

HTCondor: States and Activities

- **STATES: Owner, Unclaimed, Claimed**, Matched, Preempting, Backfill
- **ACTIVITIES:**
  - **Idle**
  - **Busy**
  - **Suspended**
  - Benchmarking
  - Checkpointing
  - Vacating
  - Retiring
  - Killing



HTCondor: Priorities

- **Jobs Priority**: `priority` = …, `condor_userprio`
- **Users' Priority:** `condor_userprio`
  - **Real priority** depends on consumed time and number of slots
  - **Best priority: 0.5** (initial), then it tries to "slowly" "chase" the number of slots that are being used at any time
  - **Effective priority** = Real priority * Priority Factor (1000 by def.)
  - **Assignation of slots**: inversely proportional to effective priority
  - **Preemption policy**
  - `nice_user`

HTCondor Universes (runtimes)

- Vanilla
- Scheduler
- Grid
- Java
- Docker
- Standard
- Local
- Parallel
- VM

HTCondor: Commands in Submit files

- Total: ~ 200 commands and attributes, but most times we only need to specify 10-15 (sometimes up to 20)

- Basically, we need to specify:
  - Executable and its arguments
  - Inputs
  - Outputs
  - Send to the queue

Condor: Example

```
##########################################
# HTCondor Example
##########################################

N     = 5
ID    = $(Cluster).$(Process)
FNAME = condor_example

output   = $(FNAME).$(ID).out
error    = $(FNAME).$(ID).err
log      = $(FNAME).$(Cluster).log

should_transfer_files   = YES
when_to_transfer_output = ON_EXIT
transfer_input_files    = data$(Process)/
transfer_output_files   = data.out
transfer_output_remaps  = "data.out=data$(Process).out"

executable = operate.py
arguments  = input$(Process).dat

queue $(N)
```

Condor: Predefined functions and macros

- Evaluate expressions: **`eval`**`(), ...`
- Flow control: **`ifThenElse`**`(), ...`
- Manipulate strings: **`strcat`**`(),`**`substr`**`(),`**`strcmp`**`(),`**`size`**`() ...`
- Manipulate lists: **`stringListSize`**`(), `**`stringListSum`**`(), ...`
- Manipulate numbers: **`round`**`(), `**`floor`**`(), `**`ceiling`**`(), `**`pow`**`(), ...`
- Check and modify types: **`isReal`**`(), `**`isError`**`(), `**`int`**`(), ...`
- Working with times: **`time`**`(), `**`formatTime`**`(), `**`interval`**`(), ...`
- Random: **`random`**`(), `**`$RANDOM_CHOICE`**`(), `**`$RANDOM_INTEGER`**`(), ...`

Condor: Monitoring jobs

- `condor_tail, condor_ssh_to_job <id>`
- `condor_history -userlog, condor_logview, …`
- `condor_history -long XXX.YYY | grep LastRemoteHost`
- `condor_hold -constraint, …, condor_realese`
- `requirements = …  rank = …`
- `noop_job = !stringListMember("$(Process)","X,Y,...")`
- `on_exit_hold / on_exit_remove`
- `concurrency_limits = XXX:N` (DEFAULT_LIMIT: 1000, N = 1000/LIM)

NOTE: *Colors:* ***Submit file command*** ***Shell command***

More info in FAQs!!!

Condor: More useful commands

- **priority=**…, **condor_prio**: higher values will be executed earlier (also **nice_user** = True) → **condor_userprio -allusers**
- **deferral_time**, **deferral_window**, **cron_hour**, **cron_minute**, ...
- ***idlvm_with_condor.sh***: Executing IDL programs with IDL VM

- **condor_submit_dag**: dependencies, **DAG** (Directed Acyclic Graph)
- **condor_compile**: Condor Standalone Checkpointing Mechanism

NOTE: *Colors:* ***Submit file command*** ***Shell command***

More info in FAQs!!!

Condor: SCM (checkpointing)

`condor_compile`: Standard Universe, but also SCM (**Standalone Checkpointing Mechanism**). Stop and restart programs* written in **`fortran`, `C`, `C++`**…

1. **`condor_compile`** `gcc myprogram.c -o myprogram` *(Compile it)*

2. **`strip`** `myprogram` *(Remove extra info, optional)*

3. **`setarch x86_64 -R`** `./myprogram` *(Run it!)*

   *Condor: Notice: Will checkpoint to **myprogram.ckpt***
   Condor: Notice: Remote system calls disabled.

4. **`ps`** `aux | grep myprogram` *(Get the Process ID)*

   *adorta **24795** 100 0.0 2596 520 pts/13 R+ 15:44 1:02 **myprogram***
   ~~*adorta 25014 0.0 0.0 114704 960 pts/14 S+ 15:45 0:00 grep --color=auto myprogram*~~

5. **`kill -USR2`** `24795` *(Write a checkpoint)*
   `kill -TSTP` `24795` (or **`Ctrl+Z`**) *(Write a checkpoint and quit)*

6. `setarch x86_64 -R ./myprogram` **`-_condor_restart`** `myprogram.ckpt` *(Restart!)*

1. **: there are some limitations (system calls, etc). Static libraries!!*

More info in FAQs!!!

New versions...

- Avoid problems with firewalls, outgoing connections
- Improve support to GPUs
- Static slots → Partitionable and dynamic slots… Fragmentation!!

HTCondor: Documentation

- Condor at Wisconsin University (official): http://research.cs.wisc.edu/htcondor/

- **Condor at SIEpedia:** http://www.iac.es/sieinvens/siepedia (HOWTOs → Condor)
  Based on our experience with IAC's users. Constantly updated!!!

  1. Introduction
  2. Useful Commands
  3. FAQs
  4. Submit files
  5. Condor and IDL

IAC

Good scenario for HTCondor: each night of observations produces a huge amount of raw data (images, spectra, calibration files, etc.) to be processed:

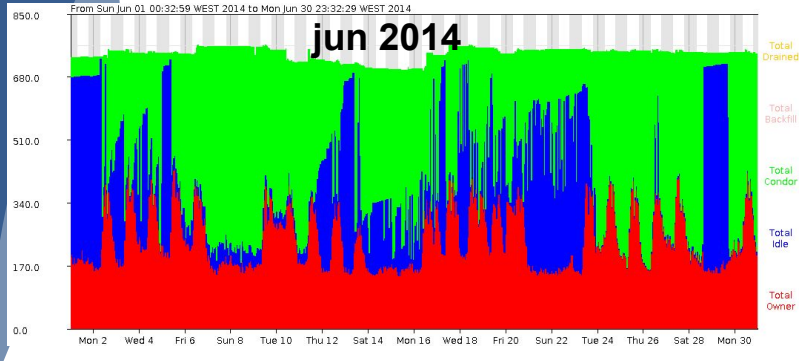
- Imaging data reduction
- Spectral analysis
- Characterizations of stars, galaxies, ...
- Numerical simulations
- Massive photometric reductions
- Models testing and evaluation
- ...

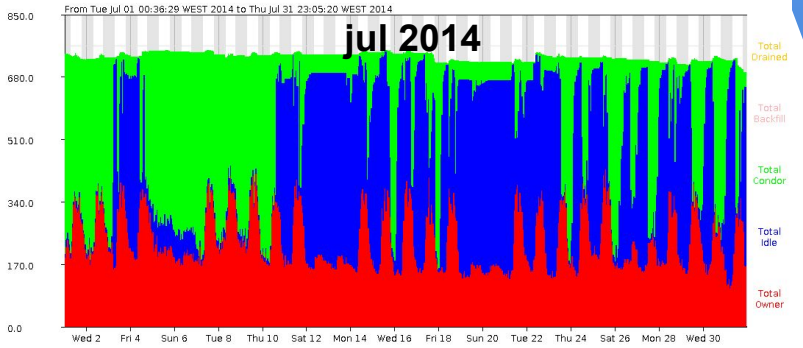
HTCondor @ IAC

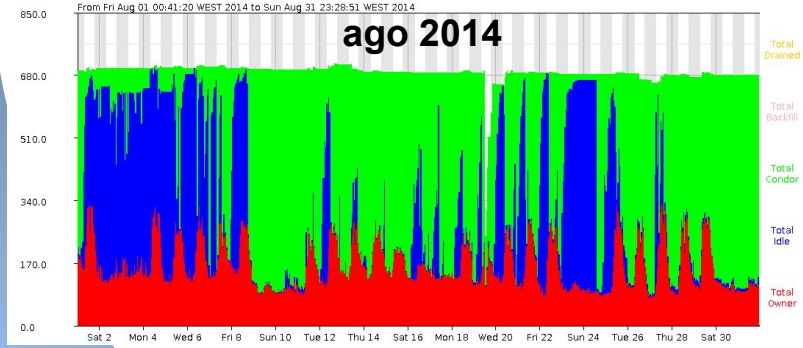
- Working since 2004, one of our Supercomputing resources
- Currently (2015):
  - Pool mainly formed by our researchers' desktops
  - ~225 machines, ~950 slots
  - Heterogeneous hardware, homogenous software
  - ~200 researches, 60+ active HTCondor users
  - ~2.400.000 CPU hours in 2014
  - We have recently obtained FEDER funds to expand and modernize our HTCondor pool

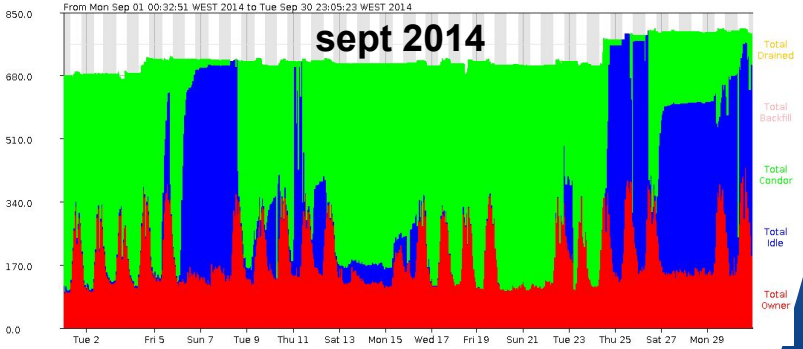
HTCondor: 2014

(http://nectarino)

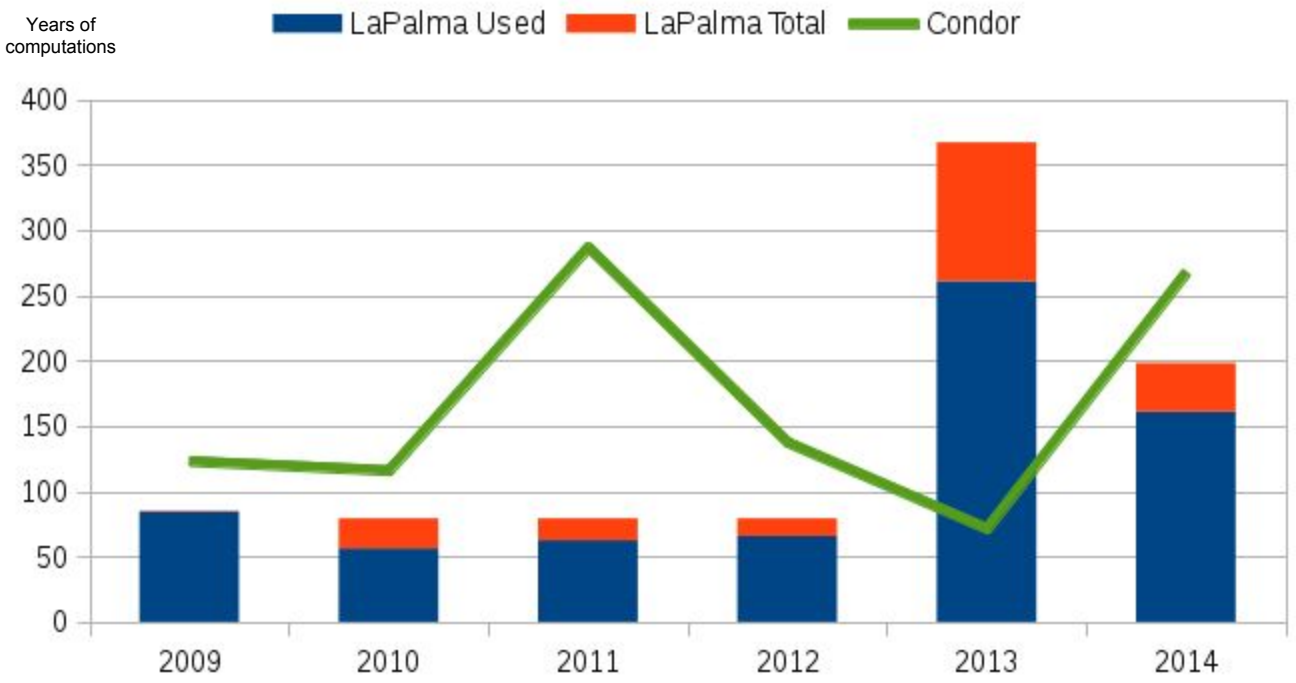








Some stats: LaPalma - Condor @ IAC



Problems and our solutions

- Programs that require a simple “click” to begin (ex: IDL VM)

  We have developed a simple script to “simulate” the click using **`Xvfb`** (X virtual framebuffer) in order to create a virtual X11 server and **`xautomation`**: “Control X from the command line for scripts, and do "visual scraping" to find things on the screen.”

- Licenses & massive access to shared disks

  Master: `CONCURRENCY_LIMIT_DEFAULT = 1000  [N = 1000 / MAX]`
  Submission script: `concurrency_limits = user$(Cluster):20,user_total:10,global=5`

- Noise in offices → **`condor_time_restrict`**

Problems and our solutions

[...]$ condor_time_restrict

-h, --help (or no arguments) : Print this help

<machine> : Print time restrictions for the given machine

-l, --listall : List restrictions for all machines

--set_restriction <machine> <time_ini> <time_end> [--days <day_ini> <day_end>] [-f] [--noreconfig]
: Add or modify a time restriction.
Time should be specify in 24h format (HH:MM)
--days: By default, restrictions are set from Monday
to Friday, you can change days range using [--days <D1> <D2>],
where days= 0:Sunday, 1:Monday, ..., 6:Saturday
-f, --force: do not ask for confirmation
--noreconfig: do not apply changes immediately
(they will be applied in next reconfiguration)

--remove_restriction <machine> [-f] [--noreconfig]
: Remove a restriction on the given machine
-f, --force: do not ask for confirmation
--noreconfig: do not apply changes immediately

--disable <machine> [-f] [--noreconfig]
: Disable a machine (it will never execute HTCondor jobs)
-f, --force: do not ask for confirmation
--noreconfig: do not apply changes immediately

Problems and our solutions

- Guarantee minimum disk space available

```
SUBMIT_EXPRS =   MachineTotalDiskString
MIN_FREE_DISK_FOR_REMOVE = ( MATCH_TotalDisk - $(MIN_FREE_DISK_KB) )
SYSTEM_PERIODIC_REMOVE = MATCH_EXP_MachineTotalDisk =!= UNDEFINED && (DiskUsage > $(MIN_FREE_DISK_FOR_REMOVE))
```

- Problems with firewalls → **`shared_port`**
- Problems detecting user's activity → `kbbd`, `condor_usb_fix`
- Hibernation of idle machines, wake up them when needed
- Nested loops, specific numeric format for input/outputs, ...

Problems and our solutions

- Master/Slaves applications
- Checking and controlling jobs that are completed in too short or long time / held jobs… “black holes”
- Waiting for input data → deferral Time
- Benchmarks, maintenance operations, backfill
- HTCondor, the scapegoat → **ConGUSTo**

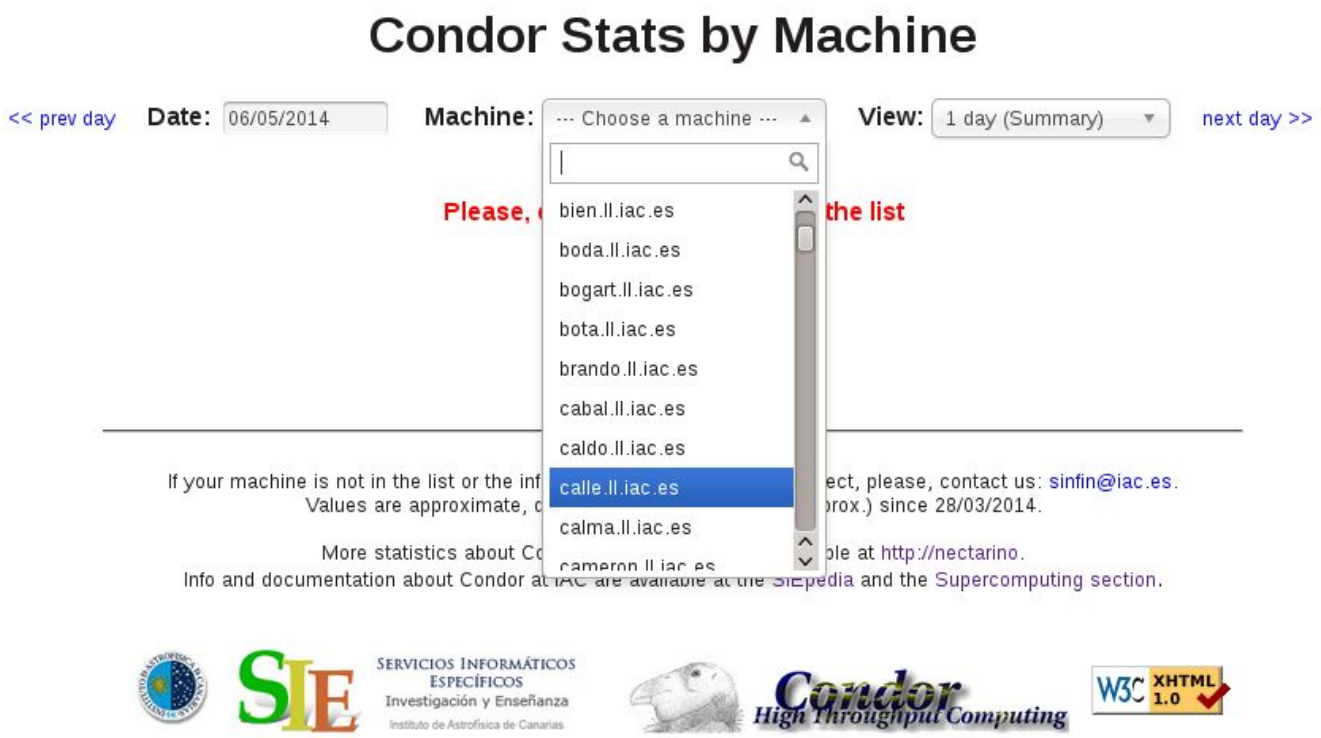
ConGUSTo

- **ConGUSTo:** HT**Con**dor **G**raphical **U**nified **S**upervising **To**ol http://arxiv.org/abs/1412.5847
- Main objective: Make HTCondor “transparent” so that users are able to check when their machines are being used or were used. Also make the administration tasks easier.
- Web-based tool: use any browser to get graphical view of all jobs executed in a specific machine, with more detailed info about each of them. Friendly and intuitive way to display data.
- Panoramic view of all machines and slots to see the current HTCondor status in a glance. Panoramic view is highly configurable with a wide range of options and filters.

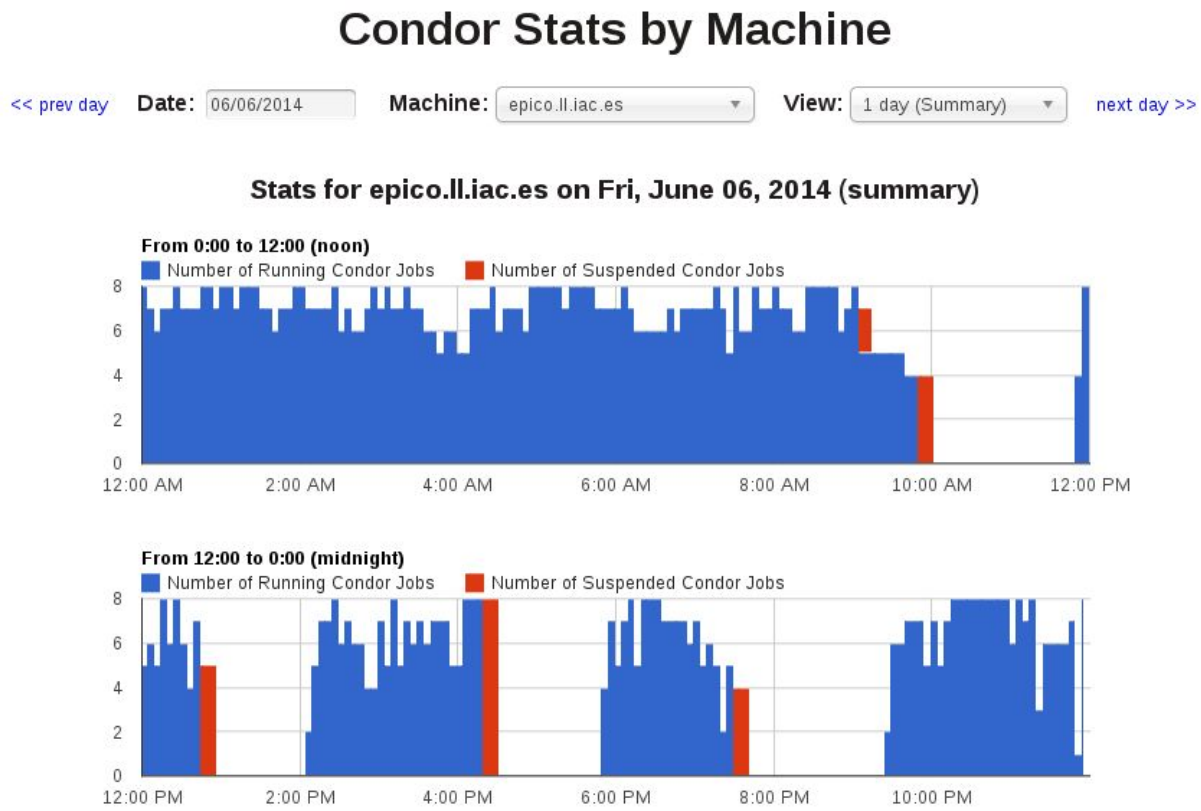
ConGUSTo

- Easy installation: ConGUSTo only needs to be installed on one machine (typically the web server). No data base is required: the data, stored in plain text with a format similar to CSV, are generated by just one of the machines in the pool.
- Developed before Condor used GANGLIA!!
- Fully compatible with previous versions of HTCondor. It shows data that at the present moment Ganglia cannot supply, such as information per slot, time restrictions, last time a machine executed a HTCondor job, and much more.
- Adding new pieces of information to show is easy.

ConGUSTo (index)



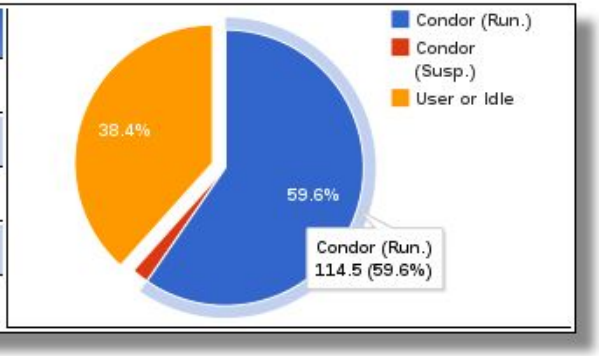
ConGUSTo (summarized daily view)



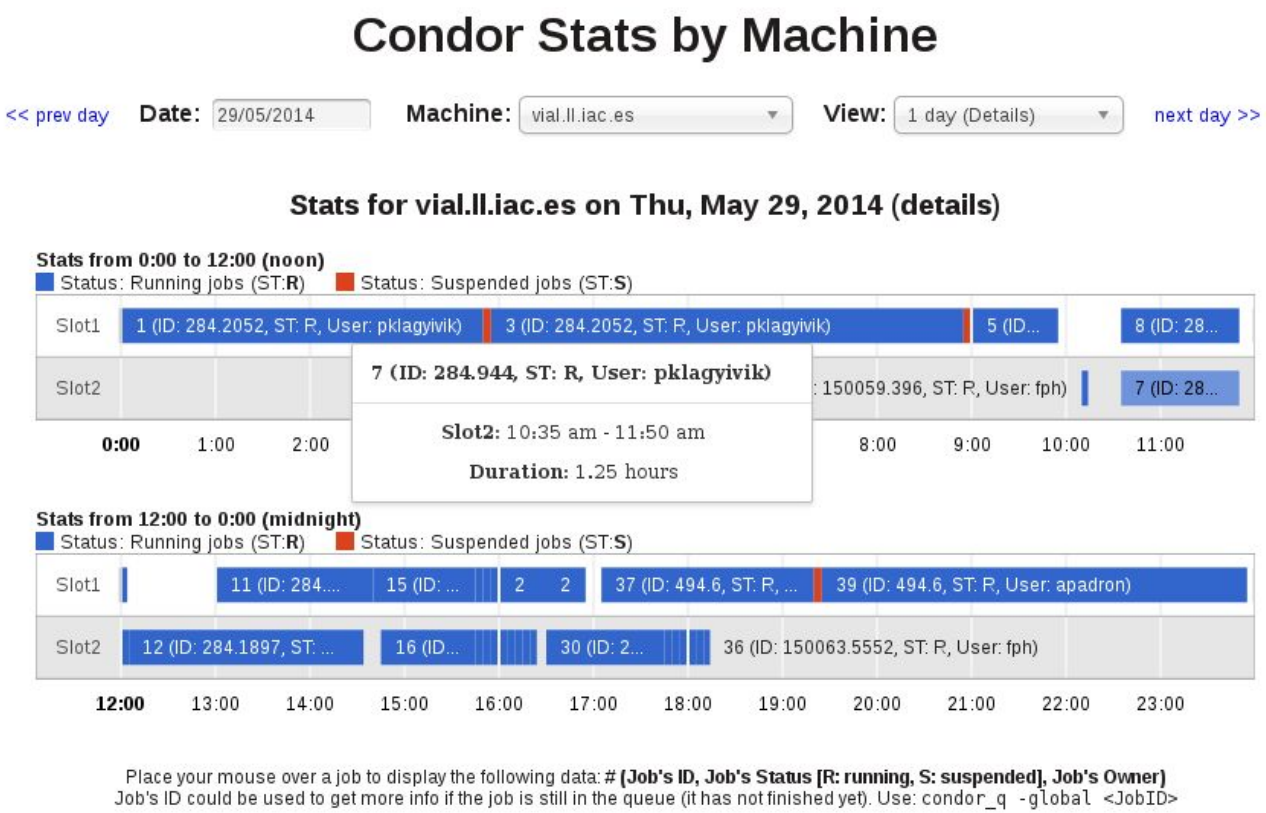
ConGUSTo (summarized daily view)

Stats summary

epico.ll.iac.es (slots: 8)	Perc.	Total (all slots)		Average per slot	
Max. available time	100.00%	192.00 h.	8.00 d.	24.00 h.	1.00 d.
User / Idle time	38.37%	73.67 h.	3.07 d.	9.21 h.	0.38 d.
Condor total time	**61.63%**	**118.33 h.**	4.93 d.	**14.79 h.**	0.62 d.
Condor (Run. Jobs: 96.76%)	59.64%	114.50 h.	4.77 d.	14.31 h.	0.60 d.
Condor (Susp. Jobs: 3.24%)	2.00%	3.83 h.	0.16 d.	0.48 h.	0.02 d.



ConGUSTo (detailed daily view)



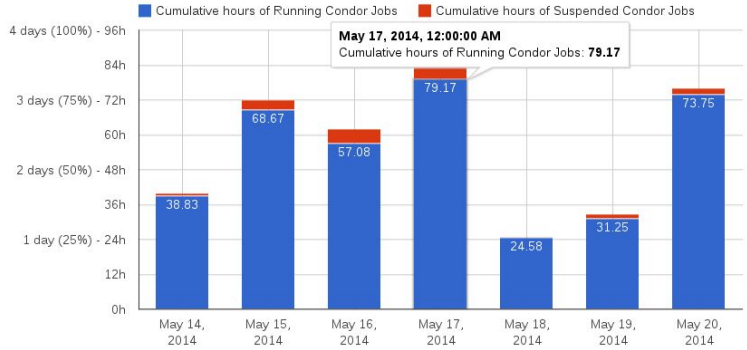
ConGUSTo (weekly and monthly views)

Condor Stats by Machine

<< prev week Date: 14/05/2014 Machine: renta.ll.iac.es View: 1 week since date next week >>

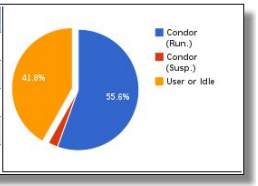
Stats for renta.ll.iac.es for week from Wed, 14/05/2014 to Tue, 20/05/2014

This machine has 4 slots (cores). Total hours per day is 96 h. (4 days). Average Condor use per day is 55.86 h. (58.18%)



Stats summary

renta.ll.iac.es (slots: 4)	Perc.	Total (all days & slots)		Average per day		Avg.(d/slot)
Max. available time (7 days)	100.00%	672.00 h.	28.00 d.	168.00 h.	4.00 d.	24 h.
User / Idle time	41.82%	281.01 h.	11.71 d.	40.14 h.	1.67 d.	10.04 h.
Condor total time	**58.18%**	**390.99 h.**	16.29 d.	**55.86 h.**	2.33 d.	**13.96 h.**
Condor (Run. Jobs: 95.48%)	55.56%	373.33 h.	15.56 d.	53.33 h.	2.22 d.	13.33 h.
Condor (Susp. Jobs: 4.52%)	2.63%	17.66 h.	0.74 d.	4.42 h.	0.18 d.	0.63 h.

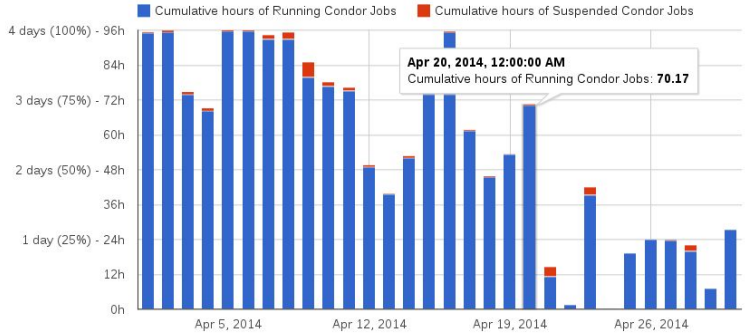


Condor Stats by Machine

<< prev month Date: 01/04/2014 Machine: renta.ll.iac.es View: 1 month since date next month >>

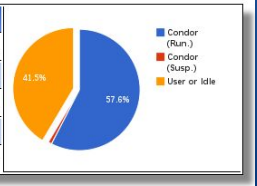
Stats for renta.ll.iac.es for month from Tue, 01/04/2014 to Wed, 30/04/2014

This machine has 4 slots (cores). Total hours per day is 96 h. (4 days). Average Condor use per day is 56.20 h. (58.54%)

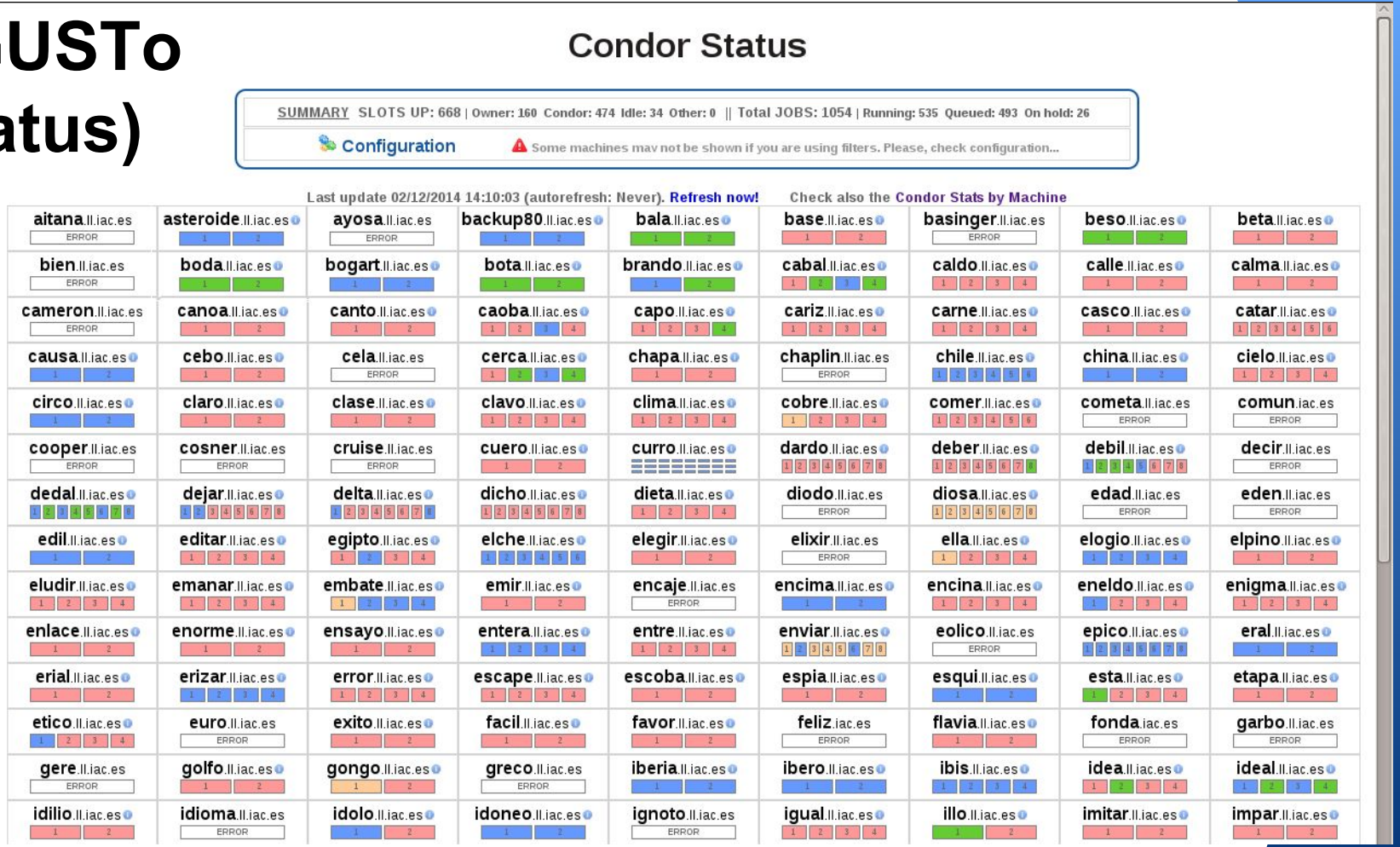


Stats summary

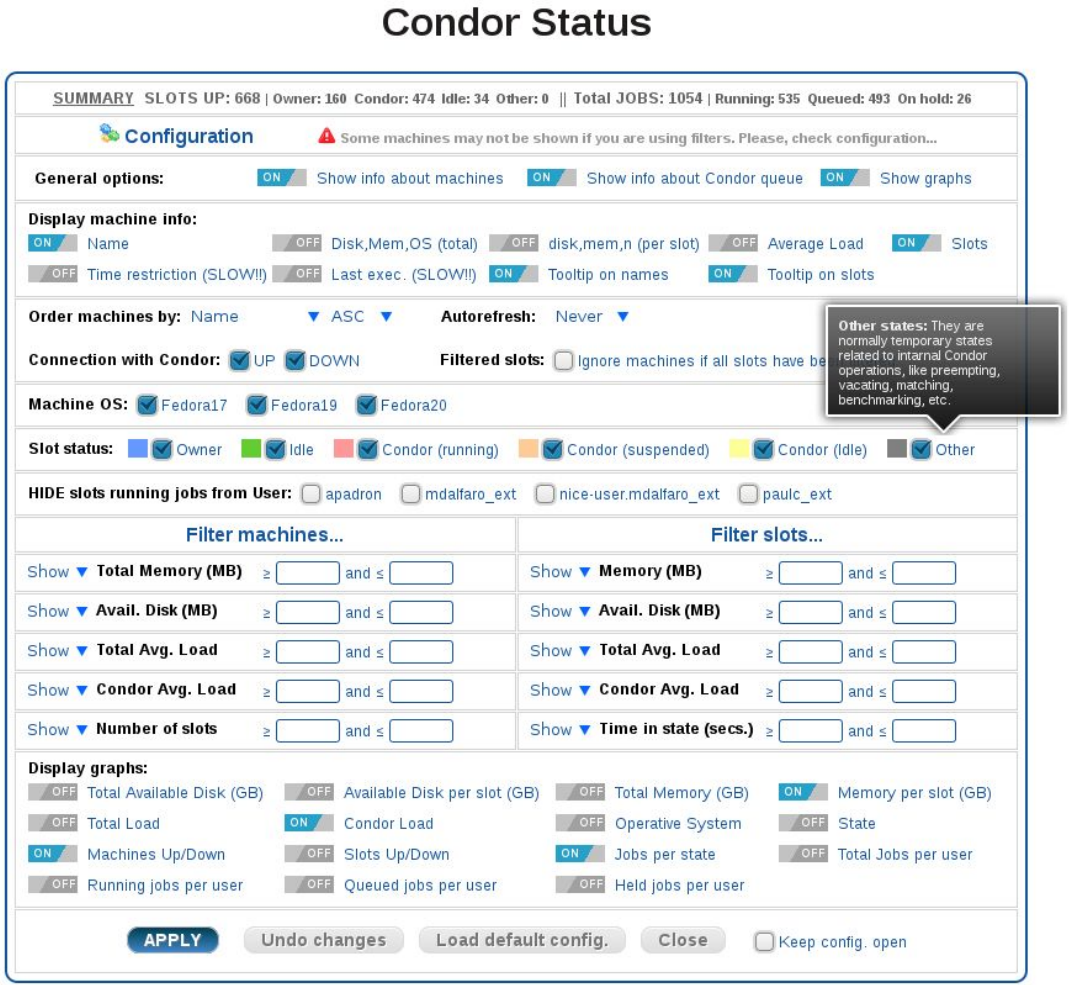
renta.ll.iac.es (slots: 4)	Perc.	Total (all days & slots)		Average per day		Avg.(d/slot)
Max. available time (30 days)	100.00%	2880.00 h.	120.00 d.	720.00 h.	4.00 d.	24 h.
User / Idle time	41.46%	1193.98 h.	49.75 d.	39.80 h.	1.66 d.	9.95 h.
Condor total time	**58.54%**	**1686.02 h.**	70.25 d.	**56.20 h.**	2.34 d.	**14.05 h.**
Condor (Run. Jobs: 98.34%)	57.57%	1658.10 h.	69.09 d.	55.27 h.	2.30 d.	13.82 h.
Condor (Susp. Jobs: 1.66%)	0.97%	27.92 h.	1.16 d.	6.98 h.	0.29 d.	0.23 h.



ConGUSTo (status)

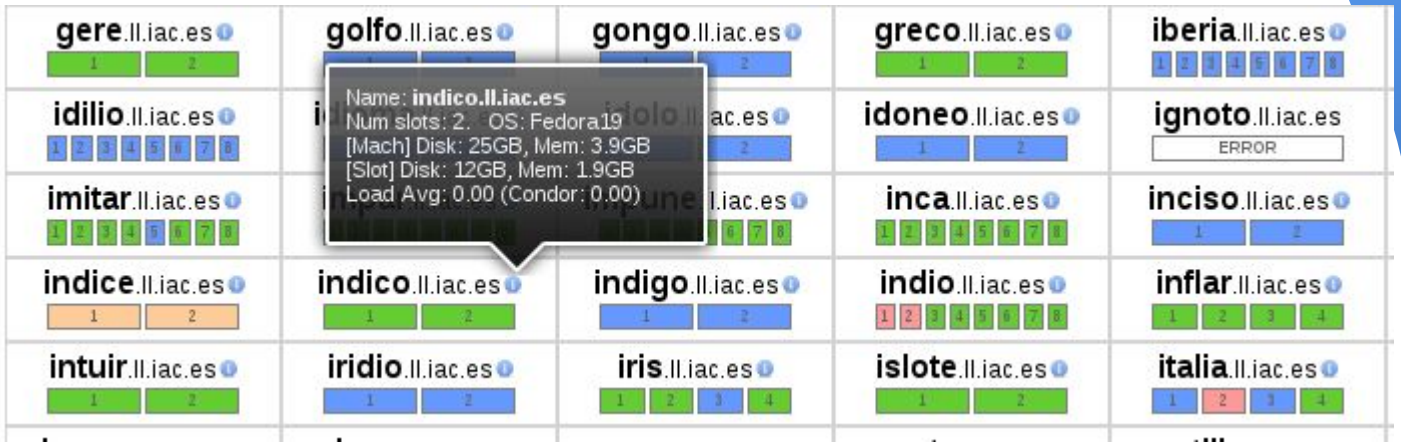


ConGUSTo (configuration)

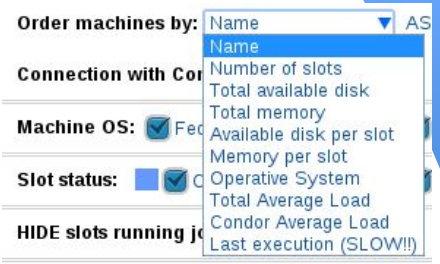


ConGUSTo (General and display options)

ConGUSTo (tooltips)



ConGUSTo (Filters)



Order machines by: Name ▼ ASC ▼ **Alert if free Disk <** 5 GB ▼ **Autorefresh:** Never ▼

Connection with Condor: ☑ UP ☑ DOWN **Filtered slots:** ☐ Ignore machines if all slots have been filtered

Machine OS: ☑ Fedora12 ☑ Fedora19 ☑ Fedora20

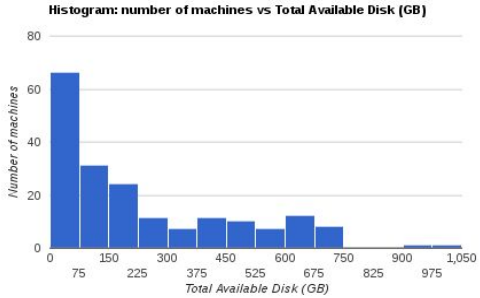
Slot status: ☑ Owner ☑ Idle ☑ Condor (running) ☑ Condor (suspended) ☑ Condor (Idle) ☑ Other

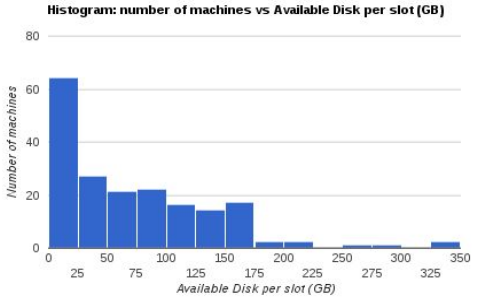
HIDE slots running jobs from User: ☐ tsh

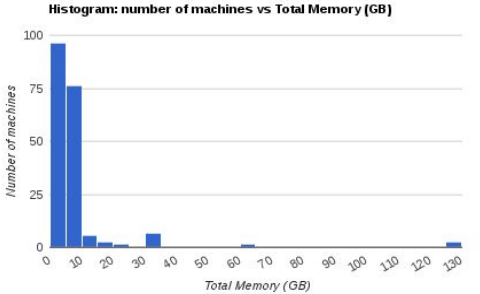
Filter machines...	Filter slots...
Show ▼ **Total Memory (MB)** ≥ and ≤	Show ▼ **Memory (MB)** ≥ and ≤
Show ▼ **Free Disk (GB)** ≥ and ≤	Show ▼ **Free Disk (GB)** ≥ and ≤
Show ▼ **Total Avg. Load** ≥ and ≤	Show ▼ **Total Avg. Load** ≥ and ≤
Show ▼ **Condor Avg. Load** ≥ and ≤	Show ▼ **Condor Avg. Load** ≥ and ≤
Show ▼ **Number of slots** ≥ and ≤	Show ▼ **Time in state (secs.)** ≥ and ≤

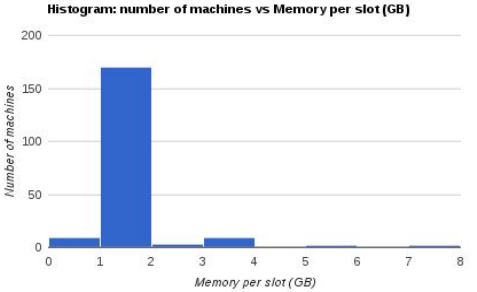
ConGUSTo (Graphs)

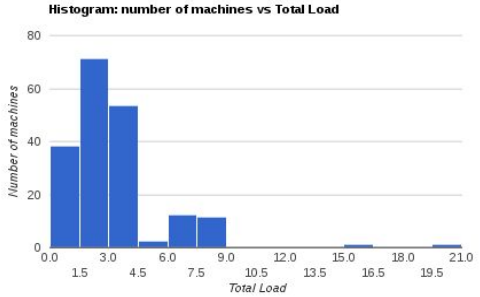
ConGUSTo (Graphs)

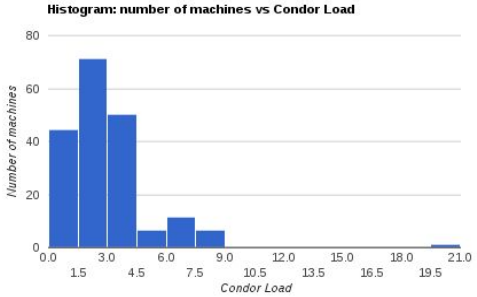






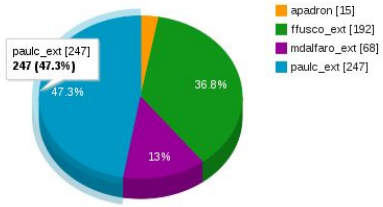




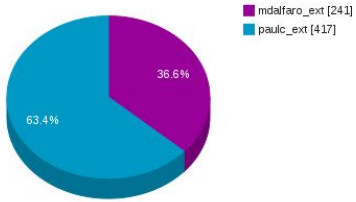


ConGUSTo (Graphs)

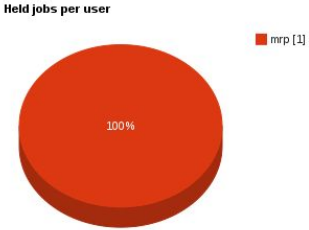
Running jobs per user



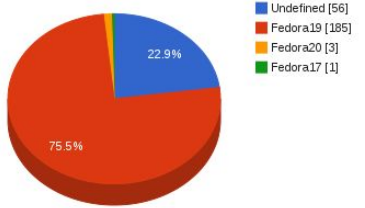
Queued jobs per user



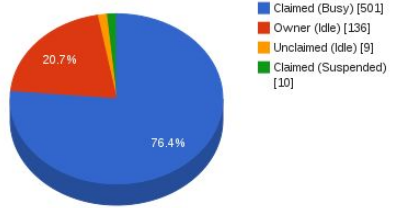
Held jobs per user



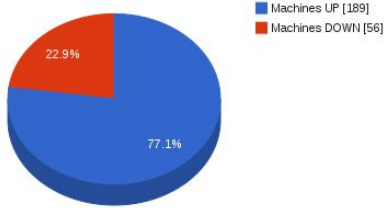
Operative System



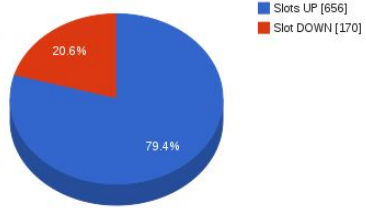
State



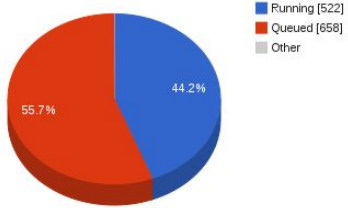
Machines Up/Down



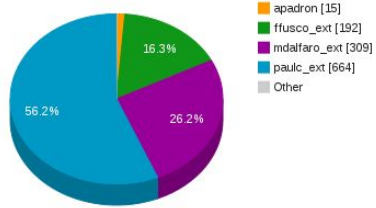
Slots Up/Down



Jobs per state



Total Jobs per user



ConGUSTo (queue and pool status)

Results after filters (Filters may affect in some graphs)
MACHINES: 245 (Up: 189, Down: 56) of 245.
SLOTS: 826 (Up: 656, Down: 170) of 826.

Condor queue (condor_status -submitters)

User	Total	Running	Queued	On hold
apastor	0	0	0	0
mrp	1	0	0	1
apadron	15	15	0	0
ffusco_ext	192	192	0	0
mdalfaro_ext	309	68	241	0
paulc_ext	664	247	417	0
TOTALS	1181	522	658	1

ConGUSTo

ConGUSTo was presented at the HTCondor Pool Administrators Workshop (CERN. Geneva, Switzerland. December 8th-11th, 2014)

Some Institutions that have obtained ConGUSTo:

- Aston Institute of Photonic Technologies, UK
- National Research Council, Canada
- BrainLab AG, Germany
- Institut des sciences chimiques de Rennes (Univ. Rennes), France
- Commonwealth Scientific Industrial and Research Organisation, Australia
- Medical University of Vienna, Austria.
- LIneA (Laboratório Interinstitucional: Observatório Nacional, Laboratório Nacional de Computação Científica, e Rede Nacional de Ensino e Pesquisa), Brazil https://soigan.linea.gov.br/congusto/

HTCondor @ IAC

Thanks!

Instituto de Astrofísica de Canarias

Tenerife - La Palma, Canary Islands. Spain

Antonio Dorta (adorta@iac.es)