What's new in HTCondor? What's coming?

European HTCondor Workshop
Sept 25, 2019 – Ispra, Italy

Todd Tannenbaum
Center for High Throughput Computing
Department of Computer Sciences
University of Wisconsin-Madison

CENTER FOR HIGH THROUGHPUT COMPUTING

HTCondor


th
ANNIVERSARY


visor
Alle
Adressen
Aufgaben
City Time
Einstellen
Graffiti
HotSync
Kalender
Kalender+
Kosten
Mail
Merkzettel
Rechner

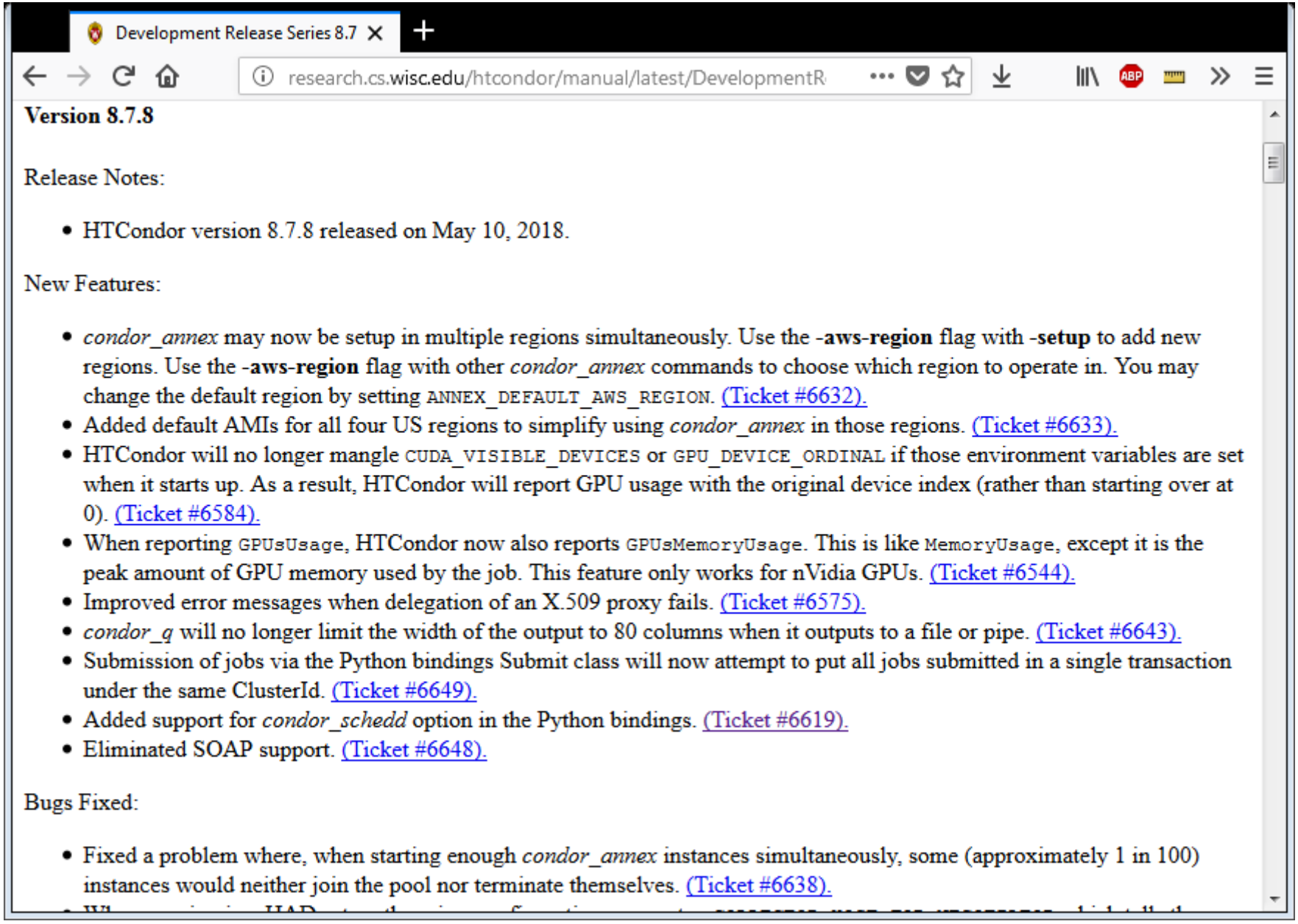
So what year was the first Condor Week meeting?

There are 2 hard problems in computer science: caching, naming, and *off-by-1 bugs*!

Release Series

- Stable Series (*bug fixes only*)
  - HTCondor v8.8.x - introduced Jan 2019

    (Currently at v8.8.4)
- Development Series (*should be 'new features' series)*
  - HTCondor v8.9.x (Currently at v8.9.2)
- Since HTCondor Week 2018…
  - Public Releases: 10
  - Documented enhancements: ~81
  - Documented bug fixes: ~78
- Detailed Version History in the Manual
  - http://htcondor.org/manual/latest/VersionHistoryandReleaseNotes.html



Version 8.7.8

Release Notes:

- HTCondor version 8.7.8 released on May 10, 2018.

New Features:

- *condor_annex* may now be setup in multiple regions simultaneously. Use the **-aws-region** flag with **-setup** to add new regions. Use the **-aws-region** flag with other *condor_annex* commands to choose which region to operate in. You may change the default region by setting `ANNEX_DEFAULT_AWS_REGION`. (Ticket #6632).
- Added default AMIs for all four US regions to simplify using *condor_annex* in those regions. (Ticket #6633).
- HTCondor will no longer mangle `CUDA_VISIBLE_DEVICES` or `GPU_DEVICE_ORDINAL` if those environment variables are set when it starts up. As a result, HTCondor will report GPU usage with the original device index (rather than starting over at 0). (Ticket #6584).
- When reporting `GPUsUsage`, HTCondor now also reports `GPUsMemoryUsage`. This is like `MemoryUsage`, except it is the peak amount of GPU memory used by the job. This feature only works for nVidia GPUs. (Ticket #6544).
- Improved error messages when delegation of an X.509 proxy fails. (Ticket #6575).
- *condor_q* will no longer limit the width of the output to 80 columns when it outputs to a file or pipe. (Ticket #6643).
- Submission of jobs via the Python bindings Submit class will now attempt to put all jobs submitted in a single transaction under the same ClusterId. (Ticket #6649).
- Added support for *condor_schedd* option in the Python bindings. (Ticket #6619).
- Eliminated SOAP support. (Ticket #6648).

Bugs Fixed:

- Fixed a problem where, when starting enough *condor_annex* instances simultaneously, some (approximately 1 in 100) instances would neither join the pool nor terminate themselves. (Ticket #6638).

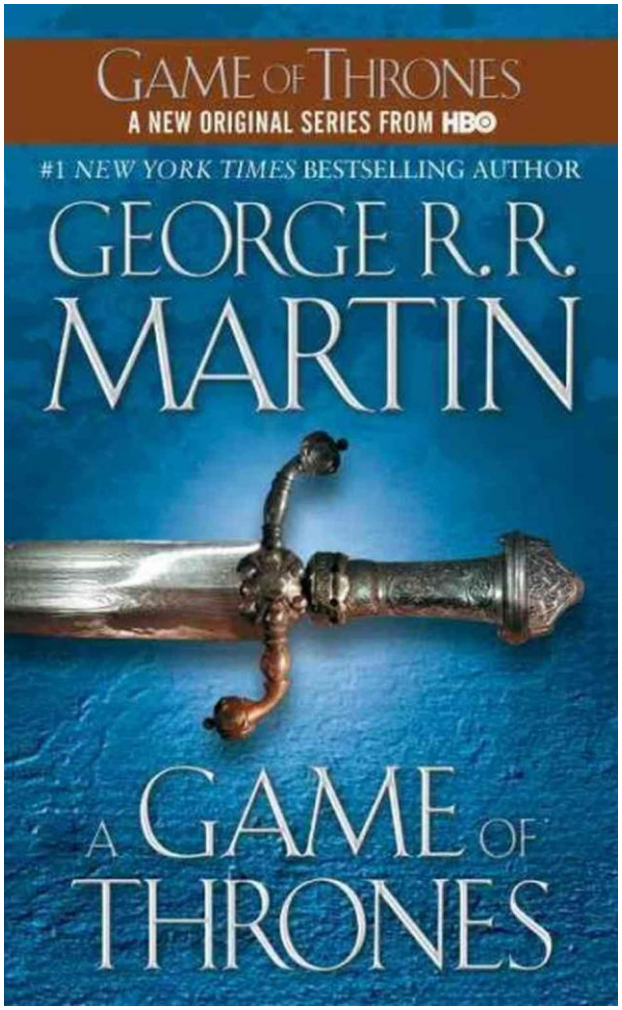
Enhancements in HTCondor v8.4

- Scalability and stability
  - Goal: 200k slots in one pool, 10 schedds managing 400k jobs
- Introduced Docker Job Universe
- IPv6 support
- Tool improvements, esp condor_submit
- Encrypted Job Execute Directory
- Periodic application-layer checkpoint support in Vanilla Universe
- Submit requirements
- New RPM / DEB packaging
- Systemd / SELinux compatibility

Enhancements in HTCondor v8.6

- Enabled and configured by default: use single TCP port, cgroups, mixed IPv6 + IPv4, kernel tuning
- Made some common tasks easier
- Schedd Job Transforms
- Docker Universe enhancements: usage updates, volume mounts, conditionally drop capabilities
- Singularity Support

What's new in v8.8 and/or cooking for v8.9 and beyond?



"You know nothing, Job Snow!"

Winter is coming for…

- SOAP API
  - Long live House Python!
- RHEL/Centos 6 Support
- Quill
- "Standard" Universe
  - Instead self-checkpoint vanilla job support [1]
- Non-standard packaging
  - UW packages and Centos/Debian packages can now be the same
- HTCondor Manual

[1] https://htcondor-wiki.cs.wisc.edu/index.cgi/wiki?p=HowToRunSelfCheckpointingJobs

Manual is now a cheat-sheet "Page790.pdf"

http://htcondor.readthedocs.io

https://htcondor.readthedocs.io/en/latest/users-manual/submitting-a-job.html

Search docs

CONTENTS

Overview

Users' Manual

Welcome to HTCondor

Introduction

Matchmaking with ClassAds

Running a Job: the Steps To Take

Submitting a Job

Sample submit description files

Read the Docs v: latest

Versions

latest stable

Downloads

PDF HTML Epub

On Read the Docs

Project Home Builds Downloads

On GitHub

View

Search

Search docs

Hosted by Read the Docs · Privacy Policy

Example 3

```
queue input,arguments from (
  file1, -a -b 26
  file2, -c -d 92
)
```

Using the `from` form of the options, each of the two variables specified is given a value from the list of items. For this example the **queue** command expands to

```
input = file1
arguments = -a -b 26
queue
input = file2
arguments = -c -d 92
queue
```

Variables in the Submit Description File

There are automatic variables for use within the submit description file.

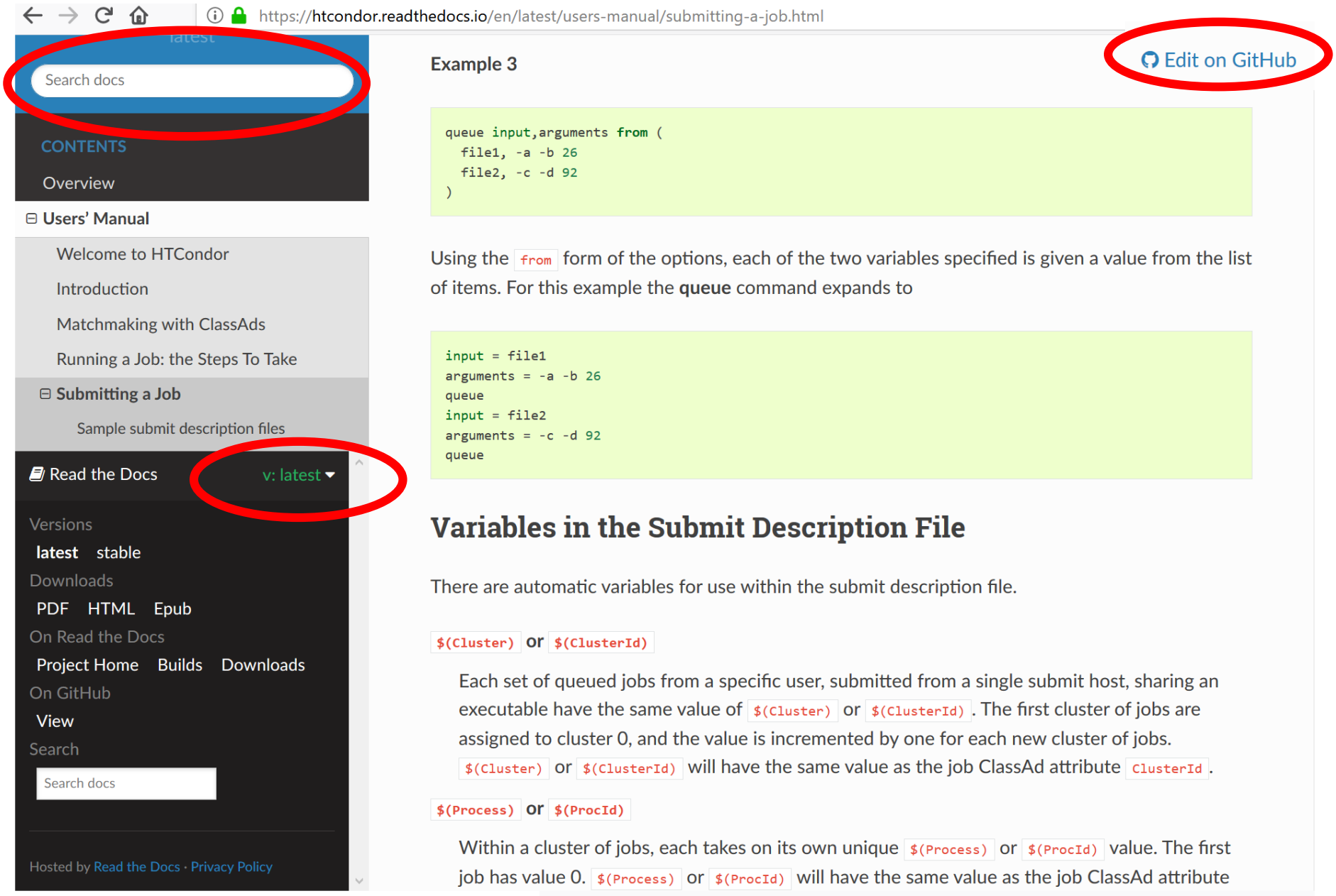
`$(Cluster)` or `$(ClusterId)`

Each set of queued jobs from a specific user, submitted from a single submit host, sharing an executable have the same value of `$(Cluster)` or `$(ClusterId)`. The first cluster of jobs are assigned to cluster 0, and the value is incremented by one for each new cluster of jobs. `$(Cluster)` or `$(ClusterId)` will have the same value as the job ClassAd attribute `ClusterId`.

`$(Process)` or `$(ProcId)`

Within a cluster of jobs, each takes on its own unique `$(Process)` or `$(ProcId)` value. The first job has value 0. `$(Process)` or `$(ProcId)` will have the same value as the job ClassAd attribute

http://htcondor.readthedocs.io



HTCondor Singularity Integration

- What is Singularity?

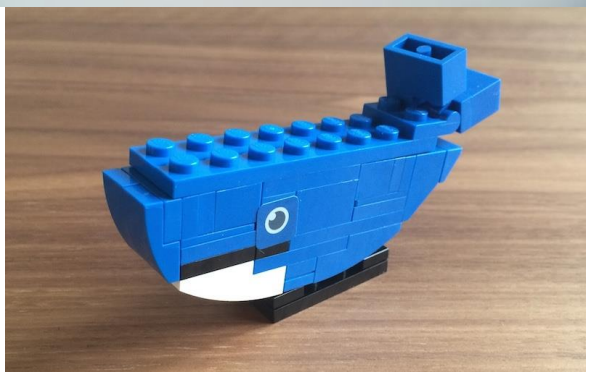
  Like Docker but…

  - No root owned daemon process, just a setuid
  - No setuid required (as of very latest RHEL7)
  - Easy access to host resources incl GPU, network, file systems

- HTCondor allows admin to define a policy (with access to job and machine attributes) to control
  - Singularity image to use
  - Volume (bind) mounts
  - Location where HTCondor transfers files

Docker Job Enhancements

- Docker jobs get usage updates (i.e. network usage) reported in job classad
- Admin can add additional volumes
- Conditionally drop capabilities
- Condor Chirp support
- Support for condor_ssh_to_job
  - For both Docker and Singularity
- Soft-kill (SIGTERM) of Docker jobs upon removal, preemption



Not just Batch - Interactive Sessions

- Two scenarios for interactive sessions
  - Interactive session alongside a batch job
    - condor_ssh_to_job: Debugging job, monitoring job
  - Interactive session alone (no batch job)
    - condor_submit –i : Juptyer notebooks, schedule shell access
    - p.s. Jupyter Hub batchspawner supports HTCondor
- Can tell the schedd to run a specified job immediately! Interactive sessions, test jobs
  - *condor_now <job_id_to_run> <job_ids_to_kill>*
  - No waiting for negotiation, scheduling

Python

- Bring HTC into Python environments incl Jupyter
- HTCondor Bindings (`import htcondor`) are steeped in the HTCondor ecosystem
  - Exposed to concepts like Schedds, Collectors, ClassAds, jobs, transactions to the Schedd, etc
- Released our **HTMap package**
  - No HTCondor concepts to learn, just extensions of familiar Python functionality. Inspired by BNL!

htcondor package

```
import htcondor

# Describe jobs
sub = htcondor.Submit('''
    executable = my_program.exe
    output = 'run$(ProcId).out'
    ''')

# Submit jobs
schedd = htcondor.Schedd()
with schedd.transaction() as txn:
  clusterid = sub.queue(txn,count = 10)

# Wait for jobs
import time
while len(schedd.query(
   constraint='ClusterId=='+str(clusterid),
   attr_list=['ProcId'])):
      time.sleep(1)
```


THERE IS
NO SPOON
Job!


th
ANNIVERSARY

htmap package

```
import htmap

# Describe work
def double(x):
    return 2 * x

# Do work
doubled = htmap.map(double,range(10))

# Use results!
print(list(doubled))
# [0, 2, 4, 6, 8, 10, 12, 14, 16, 18]
```

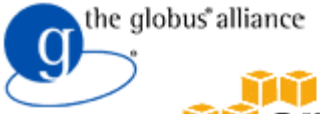
See https://github.com/htcondor/htmap

HTCondor "Annex"

- Instantiate an HTCondor Annex to dynamically add additional execute slots for jobs submitted at your site
  - Get status on an Annex
  - Control which jobs (or users, or groups) can use an Annex
- Want to launch an Annex on
  - Clouds
    - Via cloud APIs
  - HPC Centers / Supercomputers
    - Via edge services (i.e. HTCondor-CE)

Grid Universe

- Reliable, durable submission of a job to a remote scheduler
- Popular way to send pilot jobs (used by glideinWMS), key component of HTCondor-CE
- Supports many “back end” types:
  - HTCondor
  - PBS
  - LSF
  - Grid Engine
  - Google Compute Engine
  - Amazon AWS
  - OpenStack
  - Cream
  - NorduGrid ARC
  - BOINC
  - Globus: GT2, GT5
  - UNICORE



V8.8 Added Grid Universe support for Azure, SLURM, Cobalt, soon k8s

- Speak to Microsoft Azure
- Speak native SLURM protocol
- Speak to Cobalt Scheduler
- Actively working on Kubernetes!

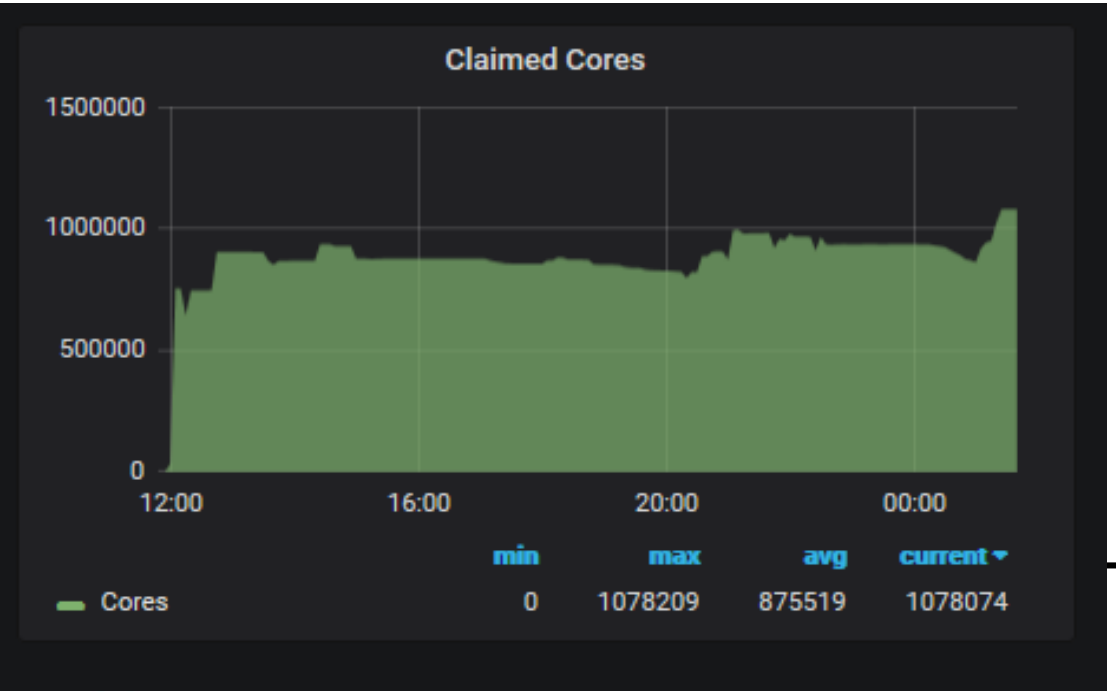
Jaime: Grid Jedi

Also HTCondor-CE "native" package

- HTCondor-CE started as an OSG package
- European Grid Infrastructure (EGI) wants to adopt HTCondor-CE without all the OSG dependencies….
- Moving HTCondor-CE upstream from OSG to HTCondor Project (plus adding EGI/WLCG accounting support)

http://news.fnal.gov/2018/07/fermilab-computing-experts-bolster-nova-evidence-1-million-cores-consumed/

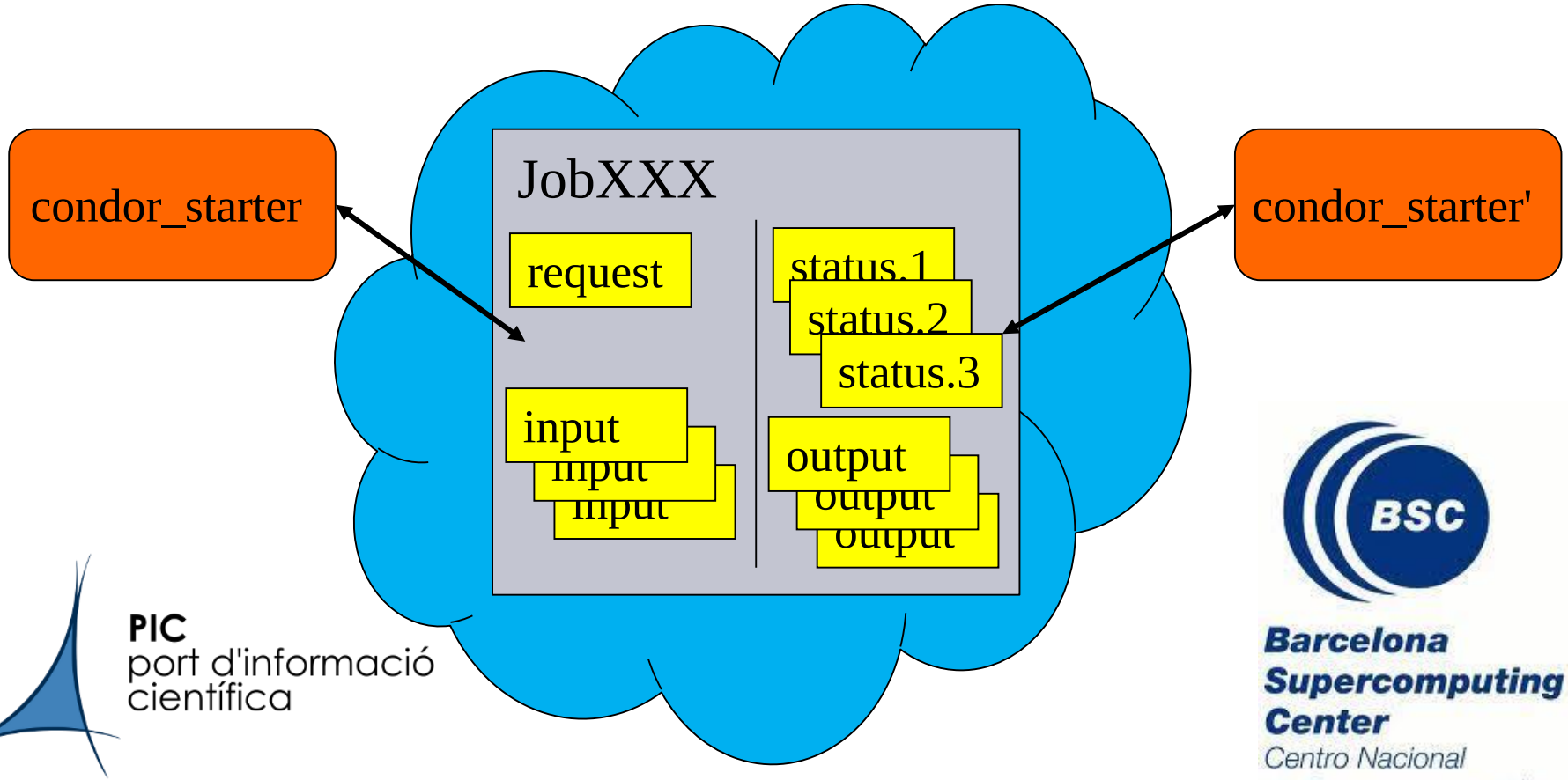
CPU cores!



FNAL HEPCloud
NOvA Run
(via Annex at NERSC)

HTCondor

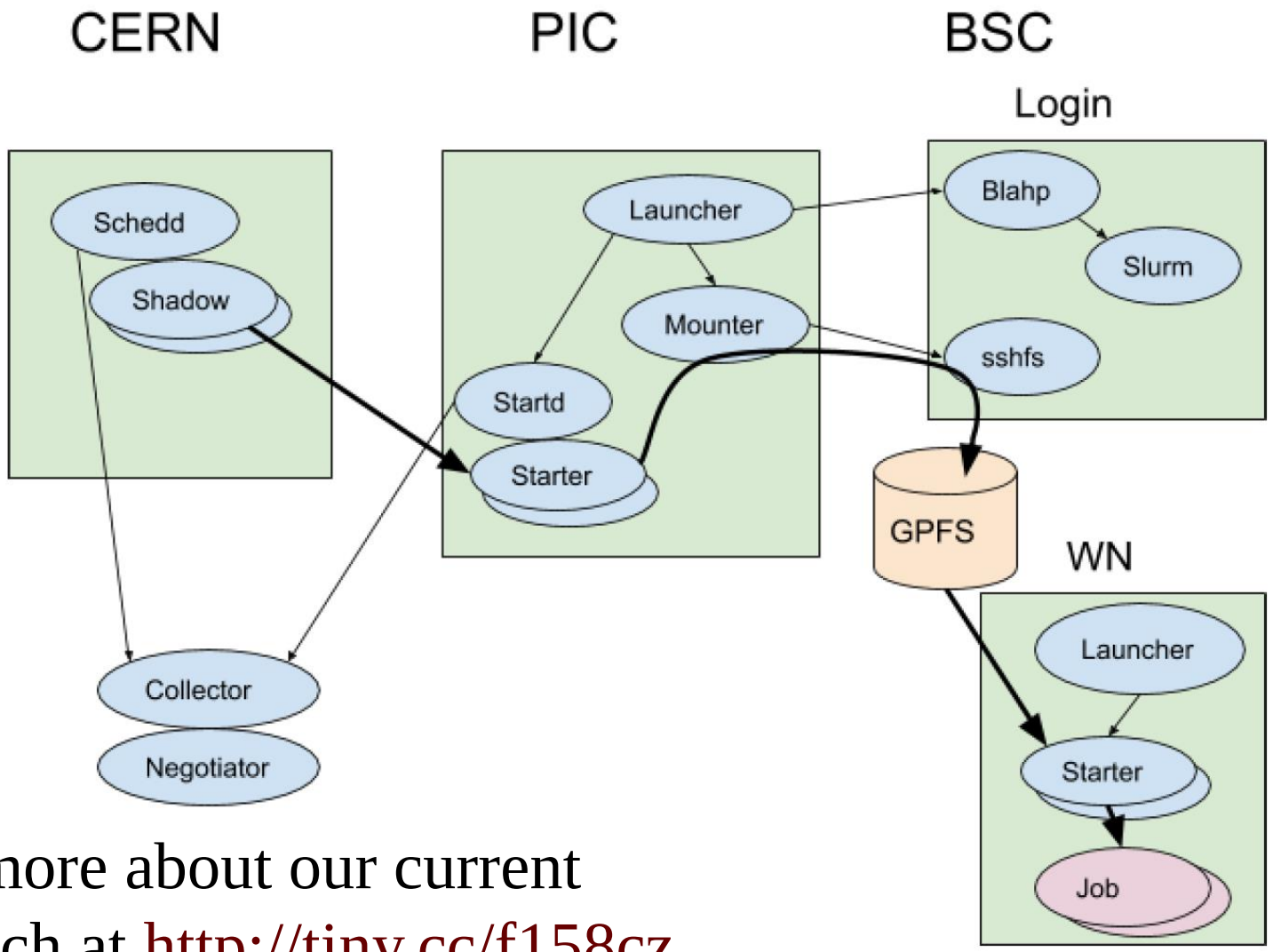
No internet access to HPC edge service? File-based communication between execute nodes



PIC
port d'informació
científica

BSC
Barcelona
Supercomputing
Center
Centro Nacional
de Supercomputación

Using MareNostrum at BSC



Read more about our current approach at http://tiny.cc/f158cz

- HTC … ect GPU … (CU…
- *New …*
  - Mo… …tion
  - Mo… …on

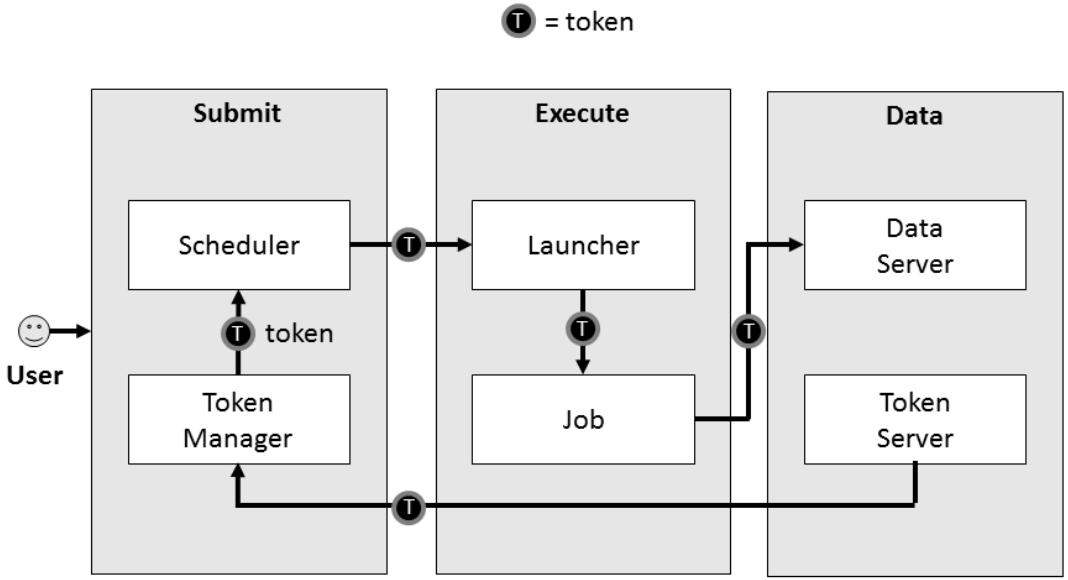
Nvidia's GeForce 256 was marketed in Oct 1999 as "the world's first GPU"

- Specify GPU memory?
- Volta hardware-assisted Mutli-Process S… (MPS)?
- Working with LIGO on requirements

Security

- Version of HTCondor available that has Federal Processing Standard (**FIPS**) Compliance
  - Currently as a separate download
  - AES has hardware support in most Intel CPUs, *so investigating at just doing TLS 1.3 all the time by default*
- TLS all the time may motivate us to drop UDP communications in HTCondor
  - Anyone care if UDP support disappears?

Security: From identity certs to authorization tokens



SCITOKENS

- HTCondor has long supported GSI certs
- Then added Kerberos/AFS tokens for CERN, DESY
- Now adding standardized token support
  - SciTokens (http://scitokens.org)
  - OAuth 2.0 Workflow → Box, Google Drive, AWS S3, …

Token Authentication Method

- Several Authentication Methods
  - File system (FS), SSL, pool password….
- Adding a new "tokens" method
  - Administrator can run a command-line tool to create a token to authenticate a new submit node or execute node
  - Users can run a command-line tool to create a token to authenticate as themselves
- "Promiscuous mode" support

File Transfer Improvements

USB "Thumb Drive" first introduced in Yr 2000 (8MB for $50)

File Transfer Improvements

- **Error messages *greatly* improved**: URL-based transfers can now provide sane, human-readable error messages when they fail (instead of just an exit code). Available in 8.8 series.
- URLs for output: Individual **output files can be URLs**, allowing stdout to be sent to the submit host and large output data sent elsewhere. Available in 8.9.1.
- **Smarter retries**. Including retries triggered by low throughput. Available in 8.9.2.
- Via both job attributes and entries in the job's event log, **HTCondor tells you the time when file transfers are queued, when transfers started, and when transfers completed**.
- If you use HTCondor to manage credentials, **we include file transfer plugins for Box.com, Google Drive, and MS One Drive cloud storage** for both input files and output files, and credentials can also be used with HTTP URL-based transfers. Available in 8.9.4.
- File transfers are now sorted by the submit host and URLs are transferred last. This means you can ensure some inputs (such as your S3 credentials!) are at the worker node before URL transfers are invoked. And **all transfers to/from the same endpoint happen over the same TCP connection**. Available in 8.9.1.
- **Performance improvements**. No network turn-around between files. Available v8.9.2
- Have an interesting use case? **Jobs can now supply their own file transfer plugins** — great for development! Available in 8.9.2.

```
executable = myprogram.exe

transfer_input_files = box://htcondor/myinput.dat

use_oauth_services = box

queue
```

Scalability Enhancements

- Central manager now manages queries
  - Queries (ie condor_status calls) are queued; priority is given to operational queries
- More performance metrics (e.g. in collector, DAGMan)
- *In v8.8 late materialization* of jobs in the schedd to enable submission of very large sets of jobs
  - Submit / remove millions of jobs in < 1 sec
  - More jobs materialized once number of idle jobs drops below a threshold (like DAGMan throttling)

Late materialization

This submit file will stop adding jobs into the queue once 50 jobs are idle:

```
executable = foo.exe
arguments = -run $(ProcessId)
materialize_max_idle = 50
queue 1000000
```

From Job *Clusters* to Job *Sets*

- Job "clusters" (even with late materialization) mostly behave as expected
  - Can remove all jobs in a cluster
  - Can edit all jobs in a cluster
- But some operations are missing
  - Append jobs to a set (in a subsequent submission)
  - Move an entire set of jobs from one schedd to another
  - Job set **aggregates** (for use in polices?)

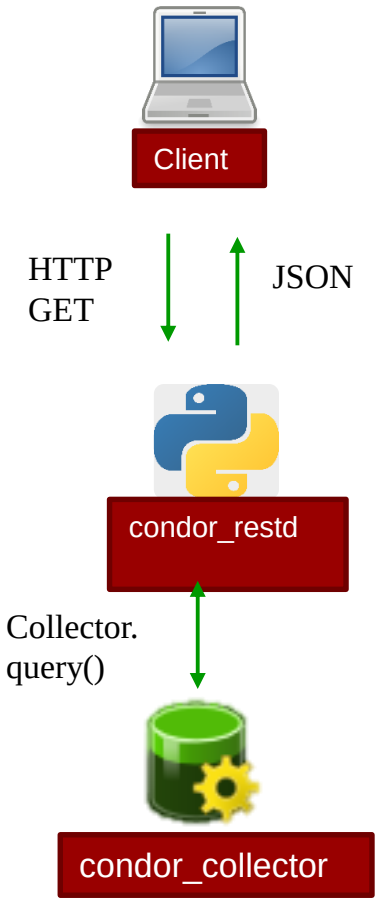
From Job *Clusters* to Job *Sets*, cont

- Users want to think about a set of jobs as it relates to their mental model, e.g.
  - Set of jobs analyzing genome 52
  - Set of jobs doing analysis on image captures from date xxx
- Initial support for sets in v8.9:
  - User supplies a set name upon submission
  - All jobs with the same name are in the same set
  - Aggregate statistics on set written to History file when last job in a set completes
  - More set operations to come…

REST API

- Python (Flask) webapp for querying HTCondor jobs, machines, and config
- Runs alongside HTCondor daemons
- Listens to HTTP queries responds with JSON

REST API, cont




```
$ curl "http://localhost:9680/v1/status\
?query=startd\
&projection=cpus,memory\
&constraint=memory>1024"
```

```
[
    {
        "name": "slot4@siren.cs.wisc.edu",
        "type": "Machine",
        "classad": {
            "cpus": 1,
            "memory": 1813
        }
    },
…
]
```

REST API, cont

- Swagger/OpenAPI spec to generate bindings for Java, Go, etc.
- Evolving, but see what we've got so far at
  - https://github.com/htcondor/htcondor-restd

- Potential Future improvements
  - Allow changes (job submission/removal, config editing)
  - Add auth
  - Improve scalability
  - Run under shared port

DAGMan: Optimizations

- **DAGMan memory diet**! Dedup node string data, edges are vectors instead of lists, etc:
  - In one DAG, these reduced the memory footprint from 50 GB to 4 GB
- Can **now submit jobs directly** (and faster!) without forking condor_submit
- Introduced **join nodes**, which dramatically reduce the number of edges in dense DAGs. In one particularly dense DAG with ~300,000 edges, join nodes resulted in the following improvements:
  - ~660M edges reduced to ~1.5M edges.
  - condor_dagman memory footprint dropped from 90 GB to 1 GB
  - Parsing time reduced from 1 hour to 20 seconds

Workflows: Provisioner Nodes

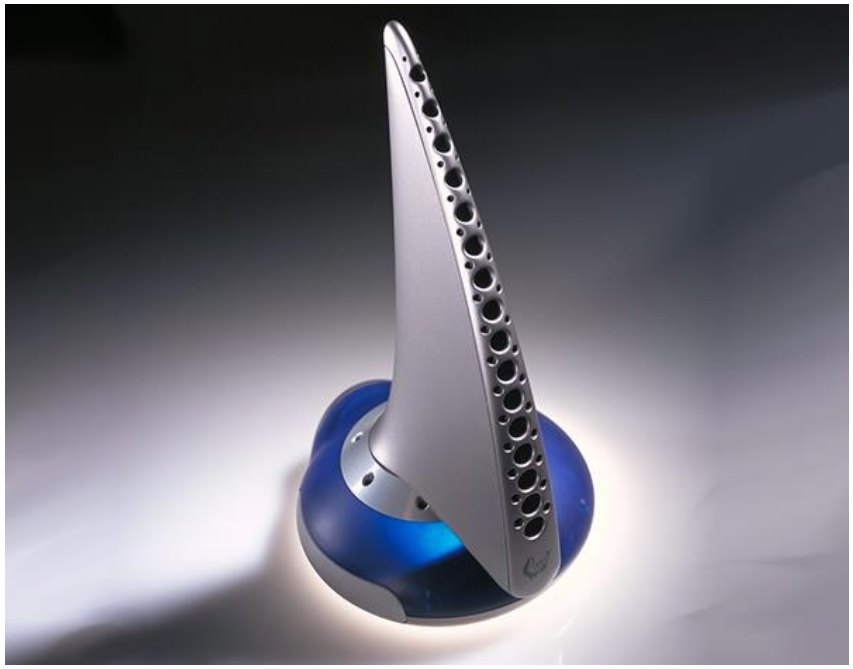
- Working to implement **provisioner nodes**
  - Special node that runs for the duration of a workflow
- Responsible for provisioning compute resources on remote clusters (Amazon EC2, Microsoft Azure, etc.)
- Important: also responsible for **deprovisioning** resources after they are no longer needed.
  - These resources cost money.
  - If we fail to deprovision them, this can incur large costs.
  - Recovery from failures is a first class citizen.

Workflows: What's Coming Next

- Sets, Sets, Sets!
  - New syntax in the DAGMan language to describe sets of jobs
- Defining ranges in DAG declarations
  - New syntax to declare ranges of objects
  - No more 10,000 JOB statements to declare 10,000 jobs whose only difference is a numeric suffic.
- condor_dagedit
  - Ability to edit certain properties of in-progress DAGs (MaxJobs, MaxIdle, etc.)
- Dataflow mode
  - Ability to skip jobs that have already been run (like /usr/bin/make!)

"Minicondor" Package

- Minicondor package
  - `yum install minicondor`
  - `apt-get install minihtcondor`
- "Personal Condor" = single machine, single user (daemons run as a regular user)
- "Minicondor" = single machine, multi-user (daemons run as root)



The "iSmell"

Thank you!