What's new in Condor? What's coming?

Condor Week 2011

Condor Project
Computer Sciences Department
University of Wisconsin-Madison

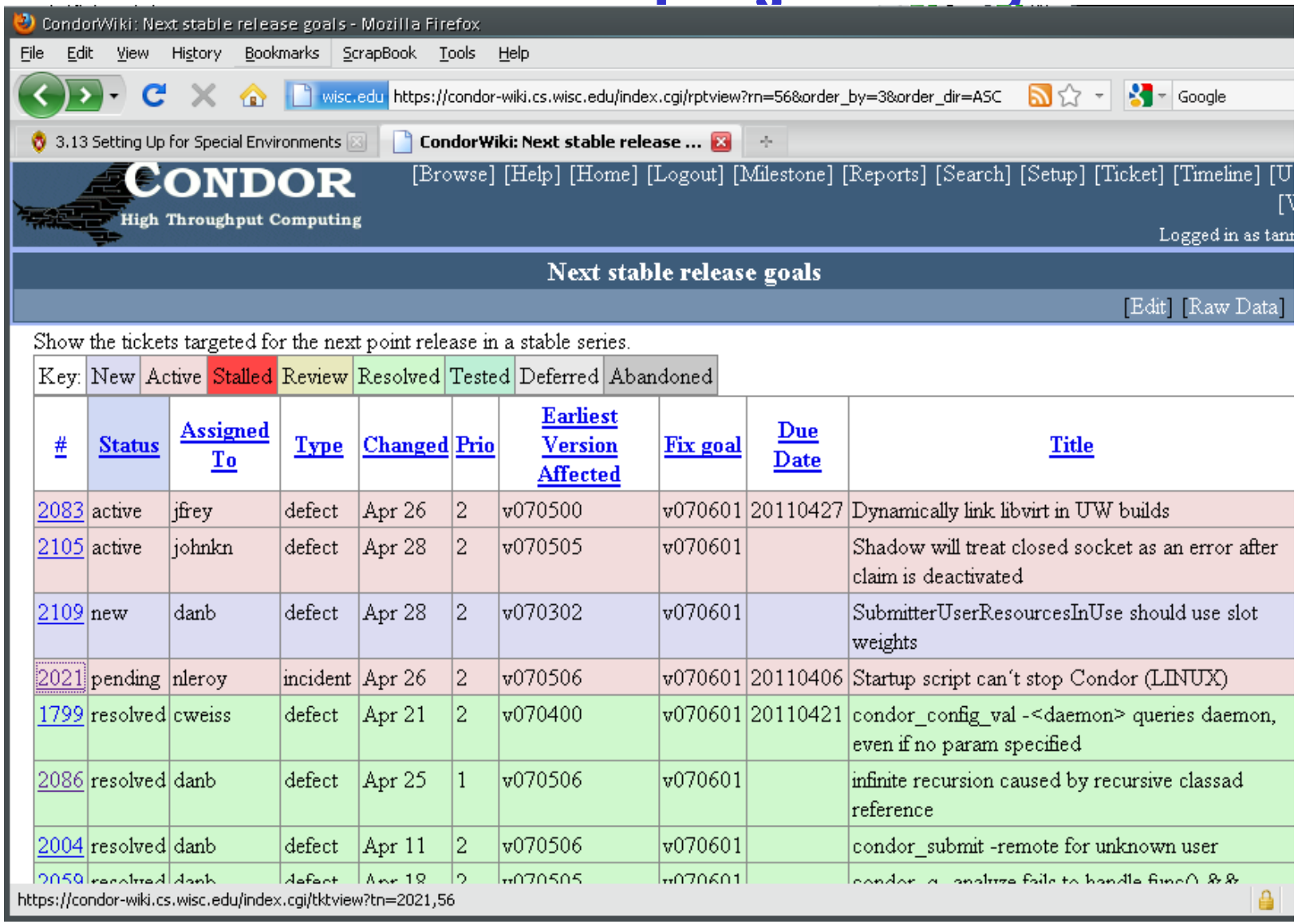
The Condor Torpedo of TRUTH Talk

Condor Project
Computer Sciences Department
University of Wisconsin-Madison

Release Situation

- Stable Series
  - Current: Condor v7.6.0 (April 19th 2011)
  - Last Year: Condor v7.4.2 (April 6th 2010)
  - Condor v7.6.1 scheduled for end of May
- Development Series
  - Current: Condor v7.7.0 (coming June 2011)
    - Why 8 weeks? Pipelining
  - Last Year : Condor v7.5.1 (March 2nd 2010)

"Work Plan for Upcoming Releases"
wiki.condorproject.org



#	Status	Assigned To	Type	Changed	Prio	Earliest Version Affected	Fix goal	Due Date	Title
2083	active	jfrey	defect	Apr 26	2	v070500	v070601	20110427	Dynamically link libvirt in UW builds
2105	active	johnkn	defect	Apr 28	2	v070505	v070601		Shadow will treat closed socket as an error after claim is deactivated
2109	new	danb	defect	Apr 28	2	v070302	v070601		SubmitterUserResourcesInUse should use slot weights
2021	pending	nleroy	incident	Apr 26	2	v070506	v070601	20110406	Startup script can't stop Condor (LINUX)
1799	resolved	cweiss	defect	Apr 21	2	v070400	v070601	20110421	condor_config_val -<daemon> queries daemon, even if no param specified
2086	resolved	danb	defect	Apr 25	1	v070506	v070601		infinite recursion caused by recursive classad reference
2004	resolved	danb	defect	Apr 11	2	v070506	v070601		condor_submit -remote for unknown user

Ports

- 7.6.0 dropped these ports from 7.4.4:
  - aix5.2-aix
  - PPC-sles9
  - PPC-yd50
  - ia64-rhel3
  - x86-debian40
  - solaris29-Sparc

Ports in v7.6

- Binaries on the web now
  - winnt50-x86.(msi|zip)
  - x86_64_deb_5.0
  - x86_64_rhap_5
  - x86_64_rhas_3
  - x86_deb_5.0
  - x86_macos_10.4
  - x86_rhap_5
  - x86_rhas_3
- Of course source code as well!
- Ports coming in v7.6 series
  - RHEL 6 x86 / x86_64
  - Debian 6 x86 / x86_64

Other choices

- Improved Packaging
  - www.cs.wisc.edu/condor/yum
  - www.cs.wisc.edu/condor/debian
- Go native!
  - Fedora, Red Hat, Ubuntu
  - Debian coming?
- Go Rocks w/ Condor Roll
- Build from source
- VDT (client side)

No Tarballs!

New goodies in v7.4

- Scalability, stabi[illegible]ty
- C[illegible]CB
- G[illegible]
- C[illegible]ion
- [illegible]_job
- [illegible]inity

New goodies in v7.6

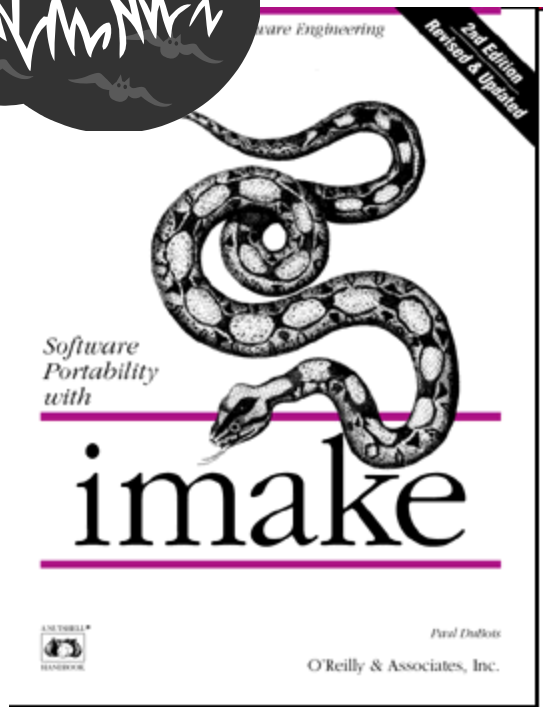
- Scalability enhancements (always…)
- File Transfer enhancements
- Grid/Cloud Universe enhancements
- Hierarchical Accounting Groups
- Keyboard detection on Vista/Win7
  - Just put "KBDD" in `Daemon_List`
- Sizeable amount of "Snow Leopard" work…

Condor "Snow Leopard"

Some Snow-Leopard Work

- Easier/faster to build
- Scratch some long-running itches, carry some long-running efforts over the finish line, such as…

Extreme-Makeover – Build system edition!



Thanks Tim, the nightmares finally stopped

CMake
Cross-platform Make

Packaging via CPack, MSI via WiX

From CondorWeek 2003:

- New version of ClassAds into Condor
  - Conditionals !!
    - if/then/else
  - Aggregates (lists, nested classads)
  - Built-in functions
    - String operations, pattern matching, time operators, unit conversions
  - Clean implementations in C++ and Java
  - ClassAd collections
- This may become v6.8.0

Is this *TODD* ?!?!

New ClassAds are now in Condor!

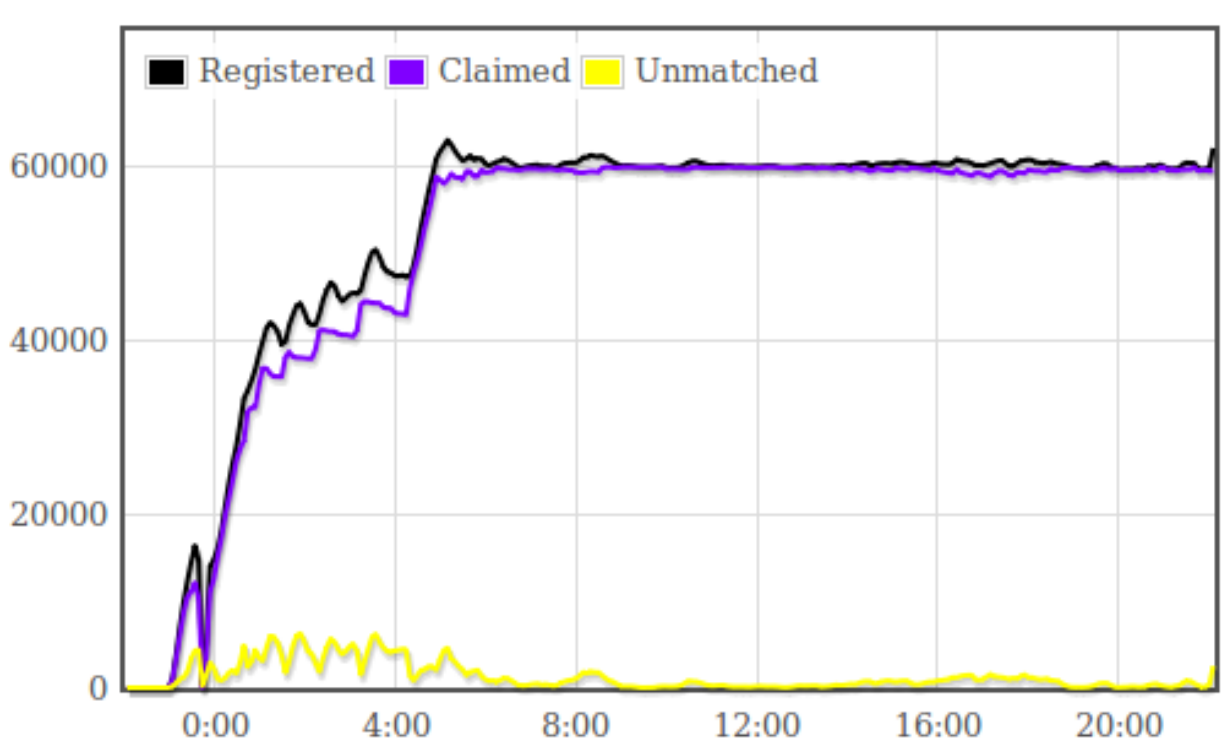
- Library in v7.5 / v7.6
  - Nothing user visible changes (we hope)… well, almost nothing
  - Logan's DAG Priority ?

  +JobPrio = DAGManJobId*100 + NumRestarts
- Developers are ready & eager to take leverage new ads in next dev series - but are users ready? Are YOU ready?

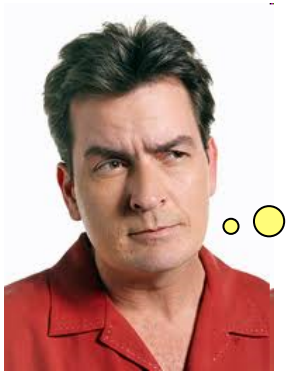
Scalability

- Everywhere
  - DaemonCore optimizations
  - Compiler optimizations enabled
- Schedd
  - Shadow recycling
  - Reduced fsync frequency (was a disk killer!)
  - Daemoncore optimizations
  - Reduced protocol overhead between shadows and the schedd
  - Asynchronous matchmaking(schedd would be unresponsive for O(10s))
- Collector
  - Moving to new classads - richer semantics and about 20% faster
  - Optimized classad insertion and removal
- Grid
  - Cream batching

of Cores managed by one Condor schedd: 60k



CHEP 2010 Paper
(slides at http://tinyurl.com/3nrawrw)



"Condor FASTBALL!"

An update on the scalability limits of the Condor batch system

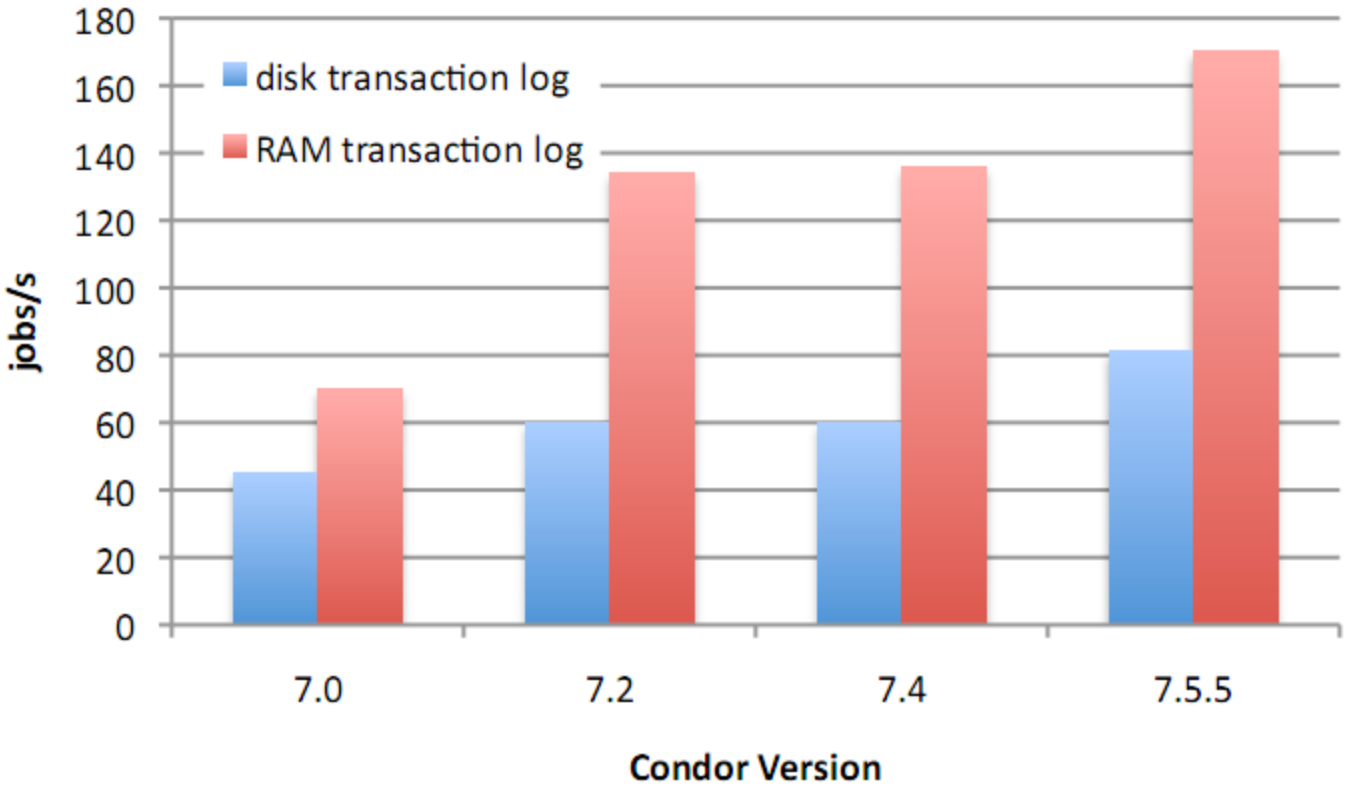
D Bradley[1], T St Clair[1], M Farrellee[1], Z Guo[1], M Livny[1], I Sfiligoi[2], T Tannenbaum[1]

[1]University of Wisconsin, Madison, WI, USA
[2]University of California, San Diego, La Jolla, CA, USA

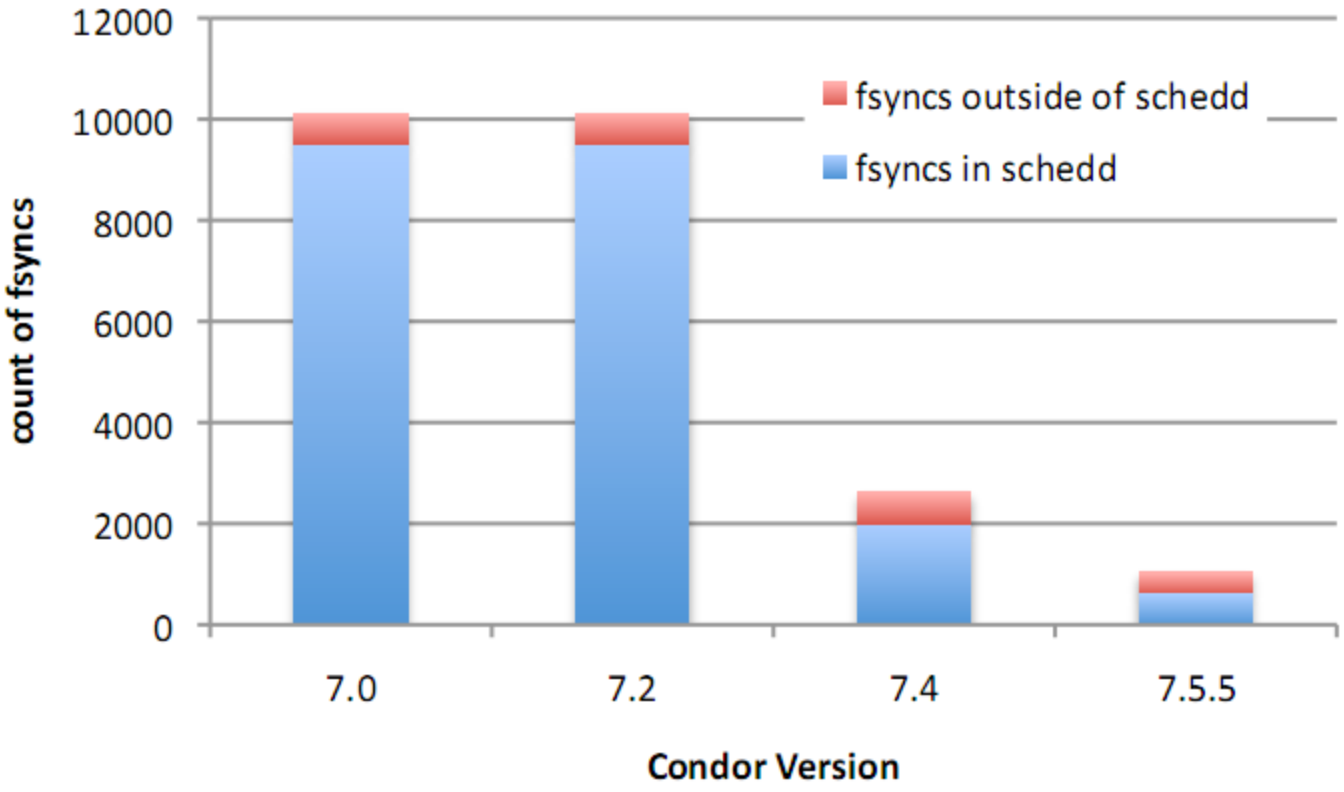
E-mail: dan@hep.wisc.edu

One Schedd: Maximum sustained job completion rate

(120k subsecond jobs, 'ideal' disk conditions)

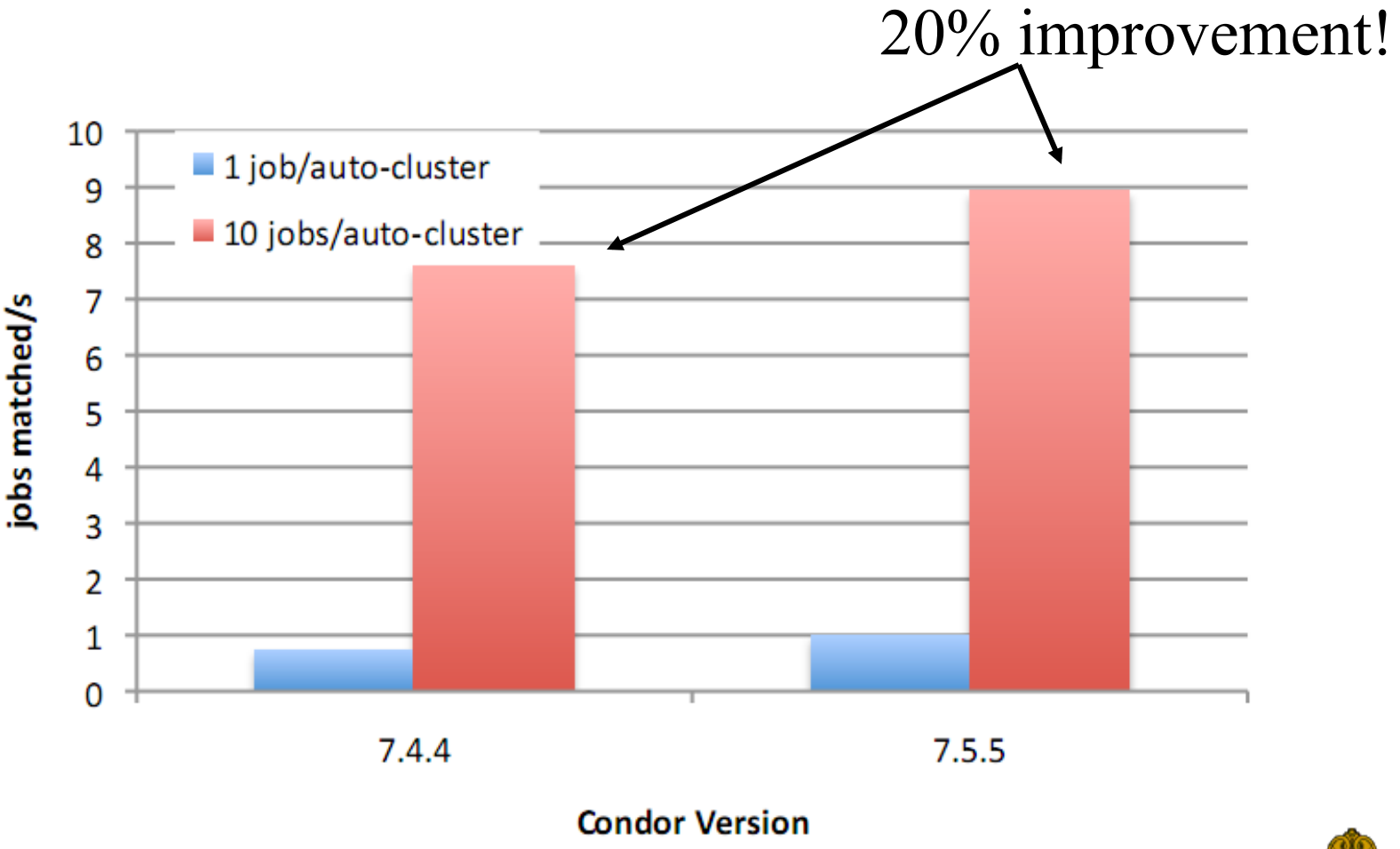


Fsync Reduction
(sample workflow, 100 jobs, realistic disk demands: event logs, $$() expansions)



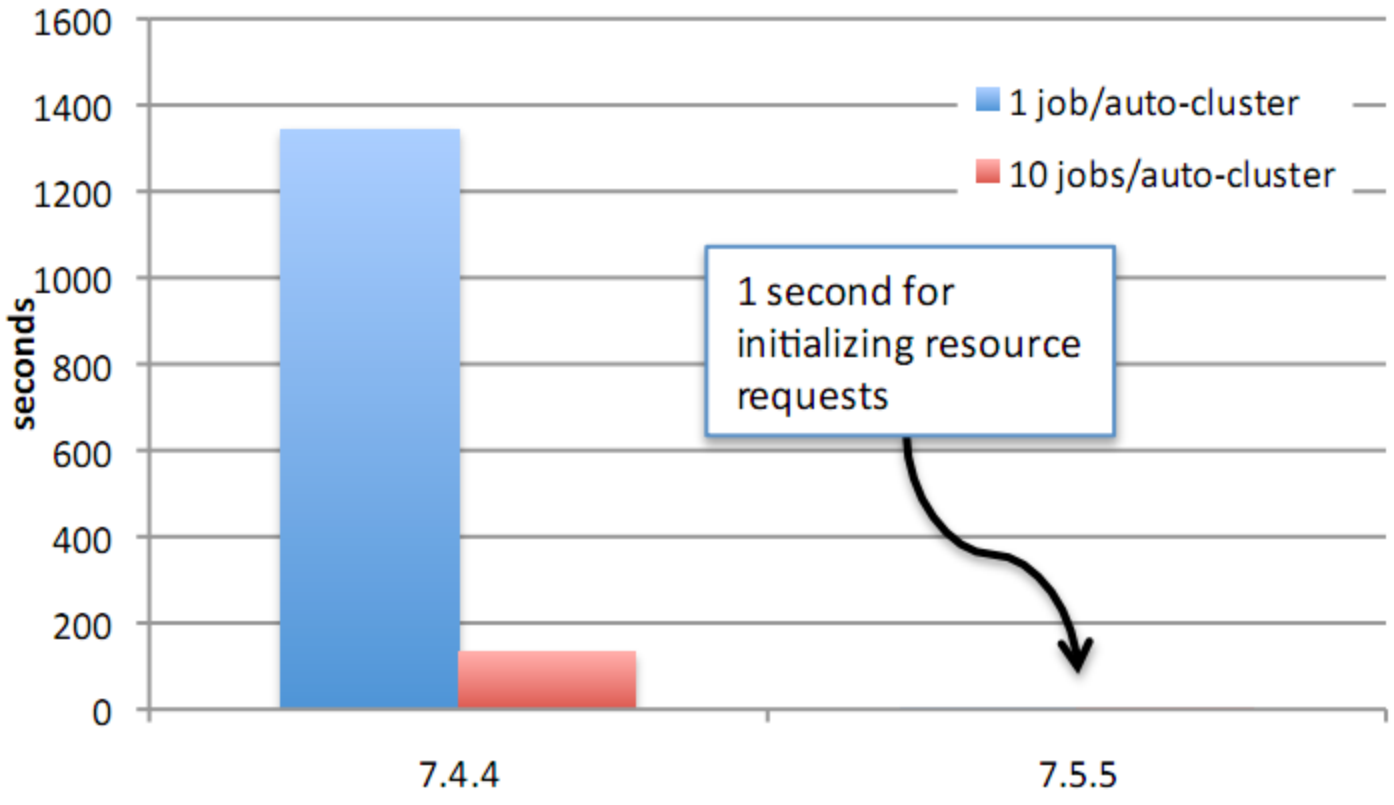
Avg # of jobs matched per second

(1K jobs matched to 10K slot pool)



Time spent in the schedd

(same test… 1K jobs matched to 10K slot pool)
Schedd responsiveness improvement very noticeable, does other work



Condor File Transfer Hooks

- By default moves files between submit and execute hosts (shadow and starter).
- File Transfer Hooks - can have URLs grab files from anywhere
  - Globus Online, HDFS, all the usual like http:// …
- New – works for both stage-in and stage-out
- New - now works in VM Universe!
  - file:// URLs can be used to allow VM disk image files to be pre-staged on execute nodes

Contribution Modules

https://condor-wiki.cs.wisc.edu/index.cgi/wiki?p=ContribModules

- Quill, Stork, Aviary, DBQ, Remote Condor, QMF Management

Directory transfer

- Transfer of directories is now supported
- Just list them along with other files to be transferred

transfer_input_files = input.txt,scripts,libs

transfer_output_files = output.txt,images

Directory transfer

- Two modes (just like rsync)

1. Transfer directory + contents
   transfer_input_files = input_dir
2. Transfer only contents
   transfer_input_files = input_dir/

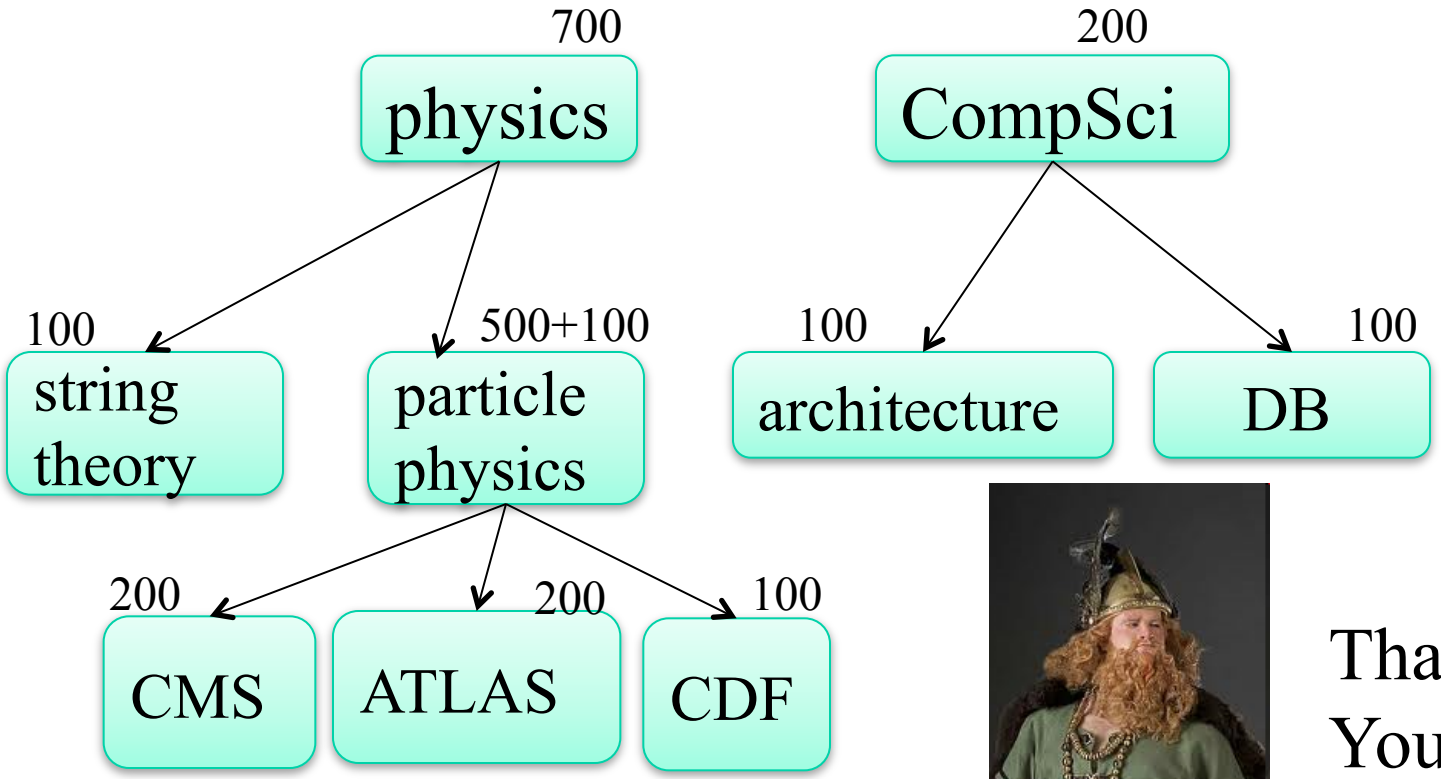
Directory transfer

- When output is auto-detected (i.e. transfer_output_files not specified), directories are *ignored*.
- This behavior may change in future versions.

Directory transfer

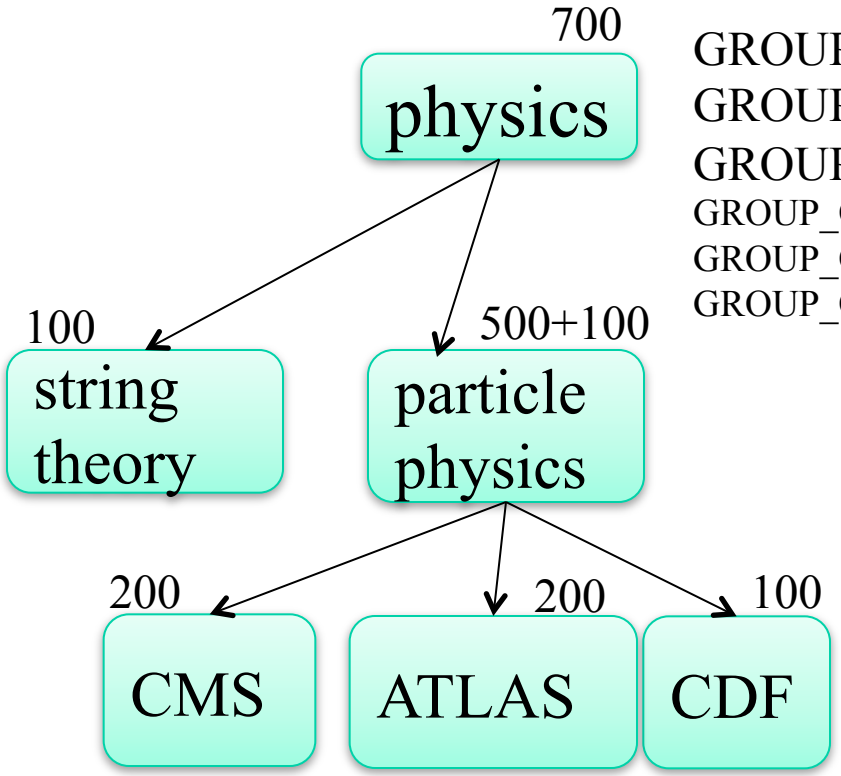
- Works in Condor-C and non-grid universes.
- Does not currently work for Globus jobs.

Hierarchical Group Quotas



Thank You Erik!

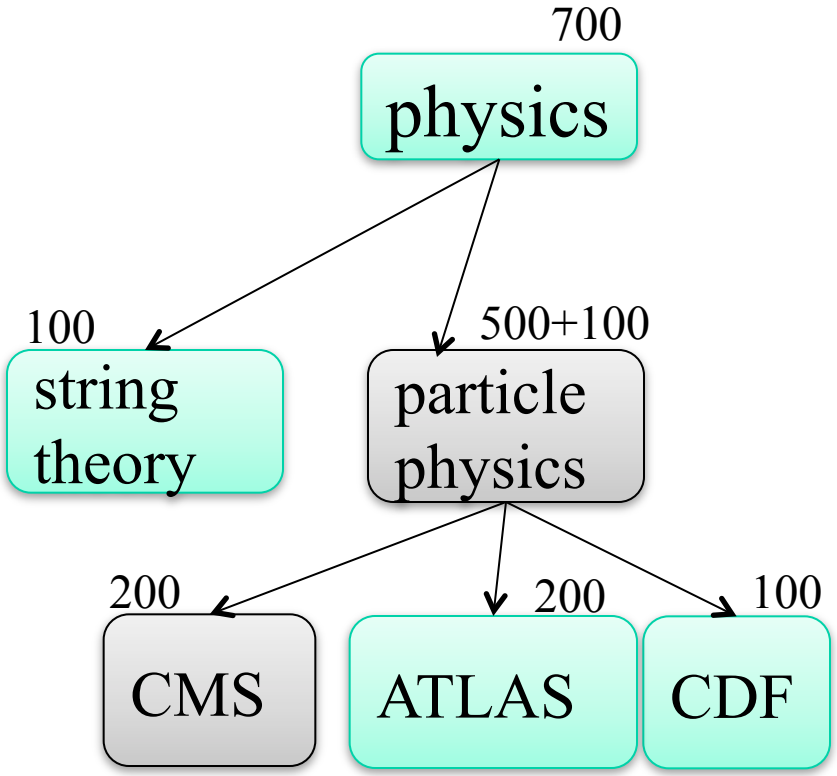
Hierarchical Group Quotas




```
GROUP_QUOTA_physics = 700
GROUP_QUOTA_physics.string_theory = 100
GROUP_QUOTA_physics.particle_physics = 600
GROUP_QUOTA_physics.particle_physics.CMS = 200
GROUP_QUOTA_physics.particle_physics.ATLAS = 200
GROUP_QUOTA_physics.particle_physics.CDF = 100
```

group.sub-group.sub-sub-group…

Hierarchical Group Quotas

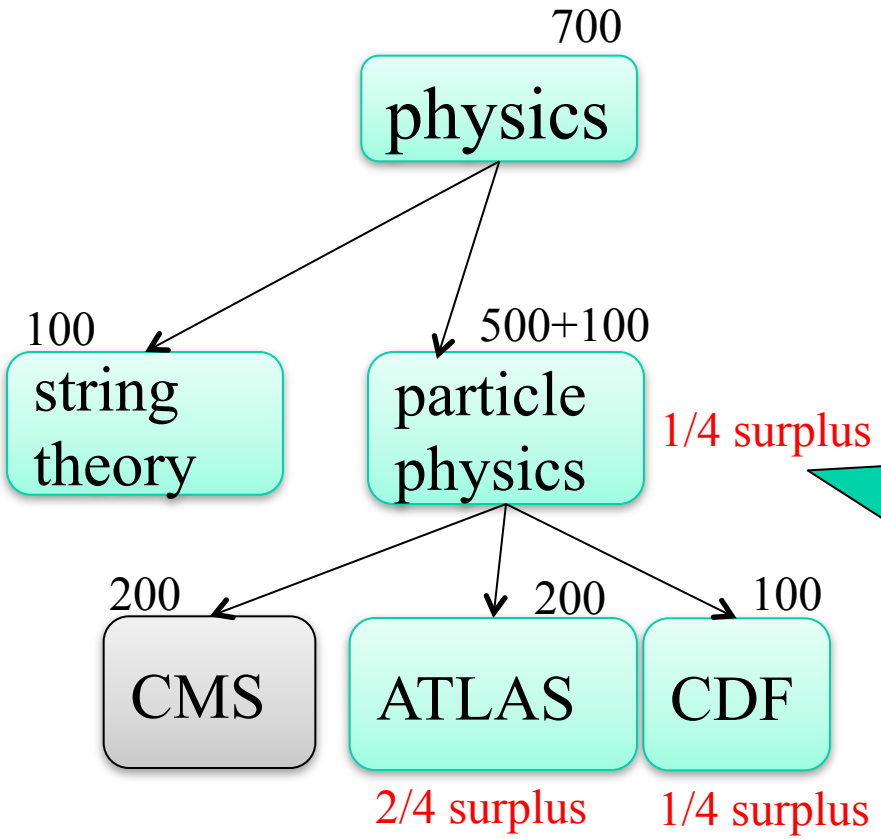


Groups configured to accept surplus will share it in proportion to their quota.

Here, unused particle physics surplus is shared by ATLAS and CDF.

```
GROUP_ACCEPT_SURPLUS_physics.particle_physics.ATLAS = true
GROUP_ACCEPT_SURPLUS_physics.particle_physics.CDF = true
```

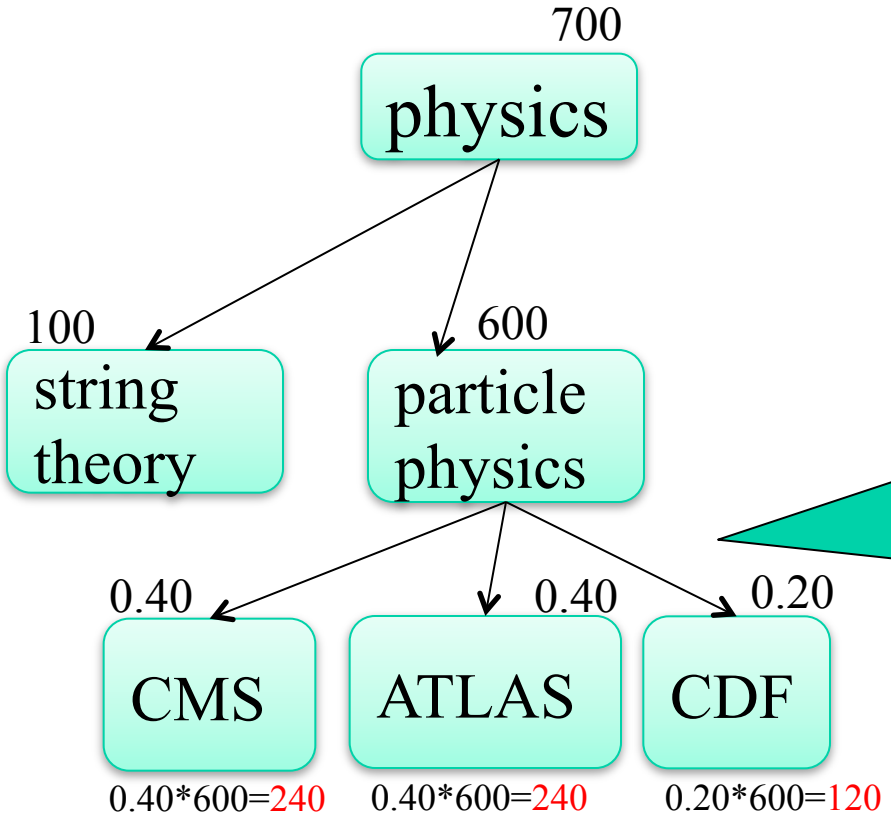
Hierarchical Group Quotas



Job submitters may belong to a parent group in the hierarchy.

Here, general particle physics submitters share surplus with ATLAS and CDF.

Hierarchical Group Quotas

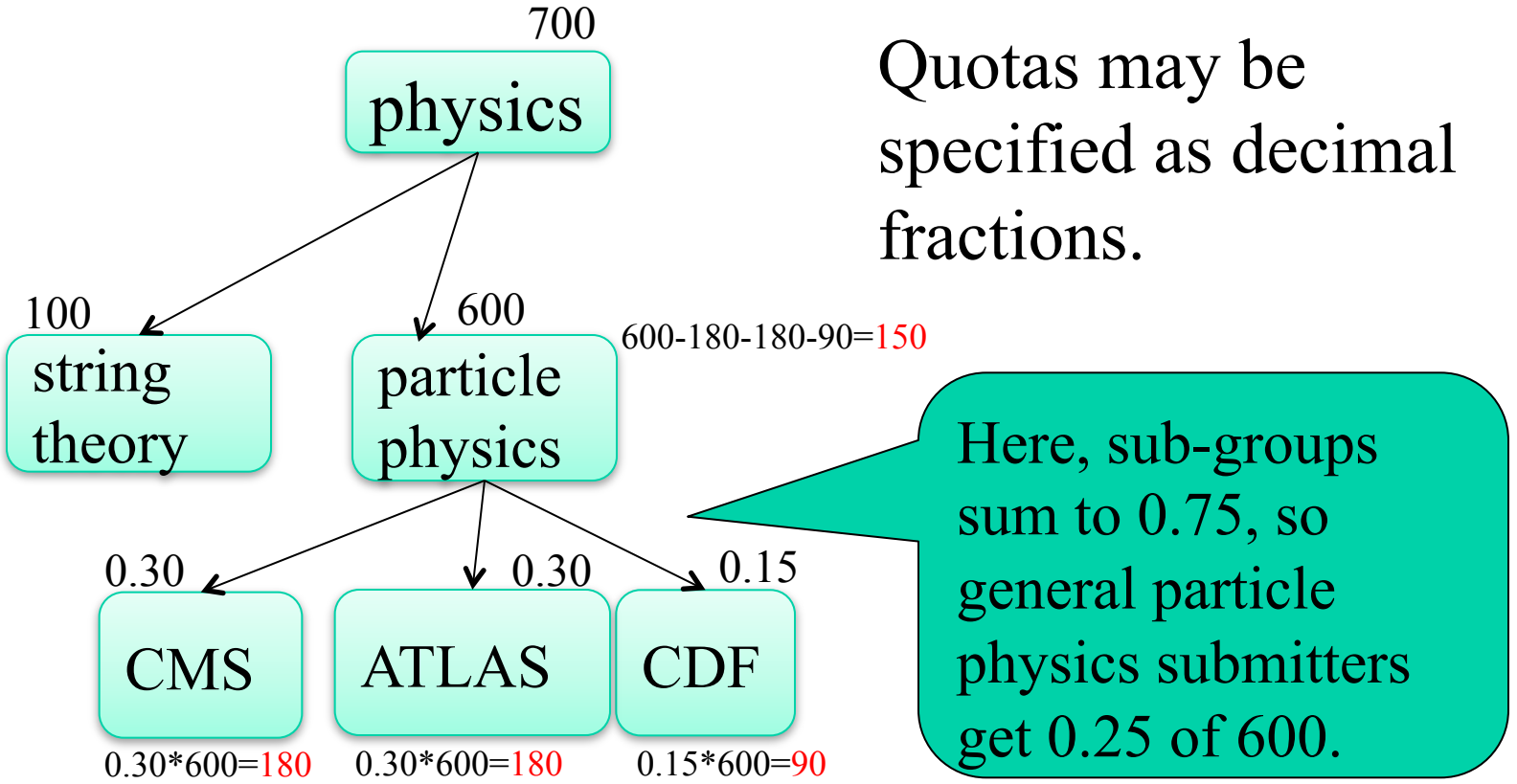


Quotas may be specified as decimal fractions.

Here, sub-groups sum to 1.0, so general particle physics submitters get nothing.

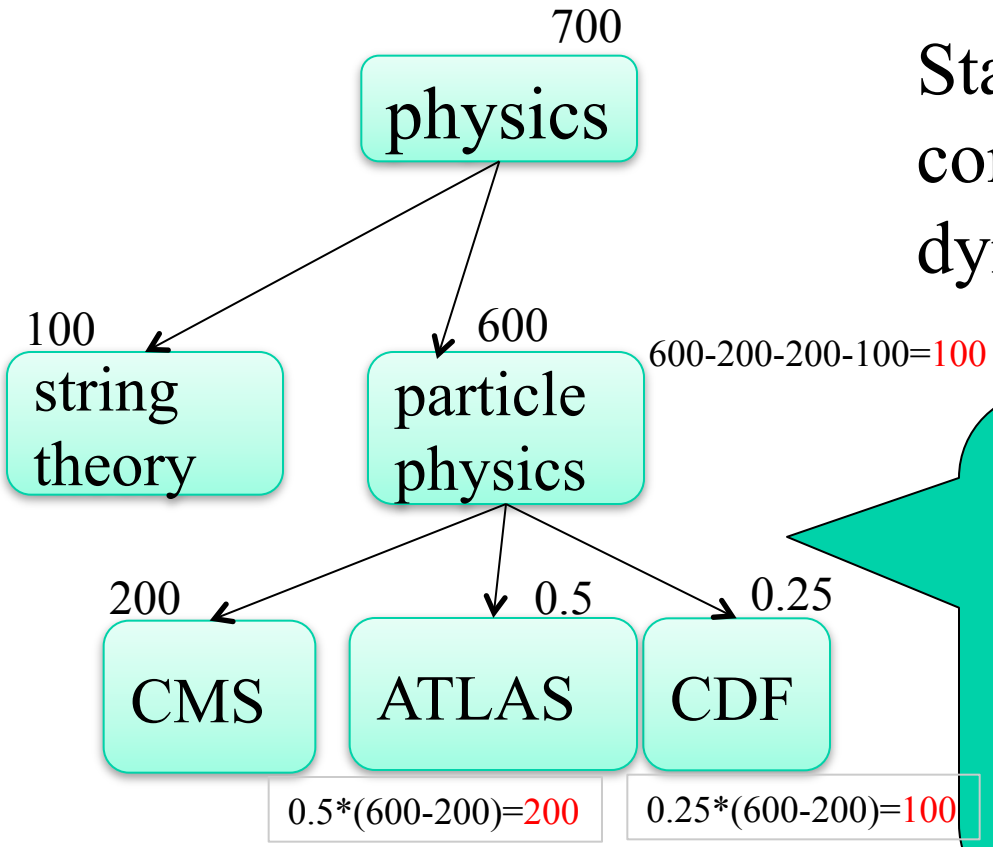
GROUP_QUOTA_DYNAMIC_physics.particle_physics.CMS=0.4

Hierarchical Group Quotas



Quotas may be specified as decimal fractions.

Hierarchical Group Quotas



Static quotas may be combined with dynamic quotas.

Here, ATLAS and CDF have dynamic quotas that apply to what is left over after the CMS static quota is subtracted.

Changed behavior: AUTOREGROUP

- GROUP_AUTOREGROUP is not identical to behavior prior to 7.5.6
- Now it is equivalent to GROUP_ACCEPT_SURPLUS
- Sharing between users in different groups determined by group quotas, *not* user priorities.

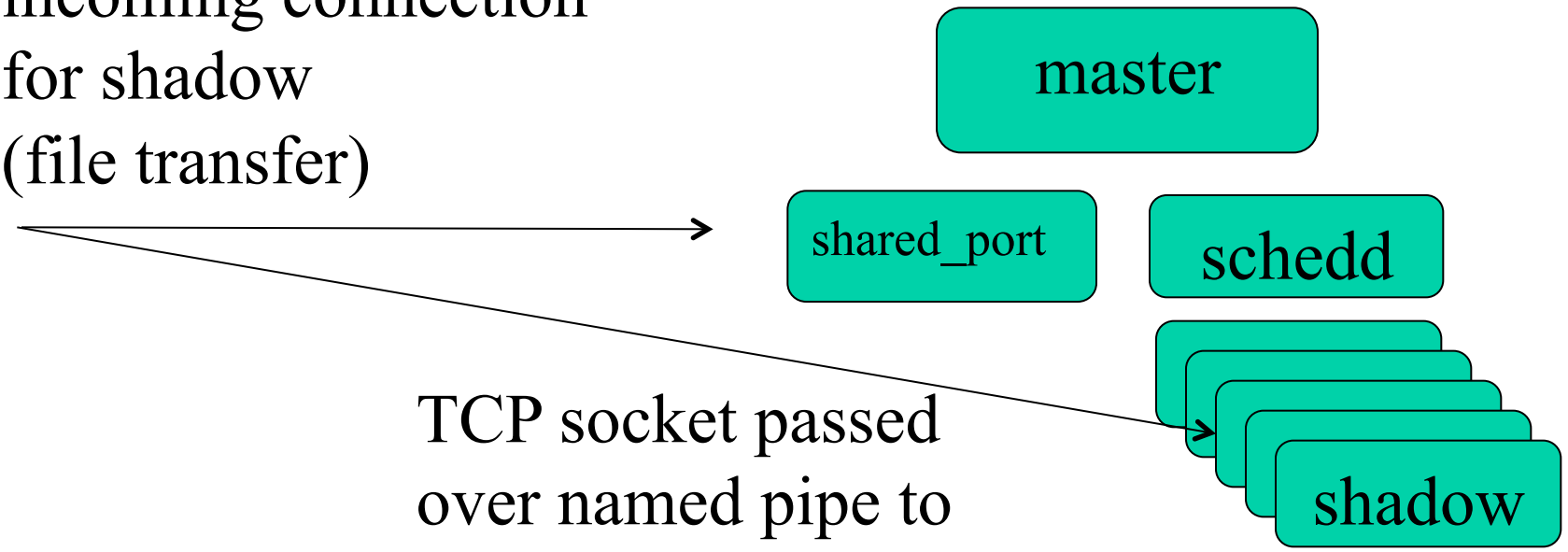
Network Port Usage

- Condor v7.4 needs a lot of open network ports for incoming connections
  - Schedd: 5 + 5*NumRunningJobs
  - Startd: 5 + 5*NumSlots
- Not a pleasant firewall situation.
- CCB can make the schedd or the startd (but not both) turn these into outgoing ports instead of incoming

Have Condor listen on just *one* port per machine

How it works

incoming connection
for shadow
(file transfer)



TCP socket passed
over named pipe to
intended recipient

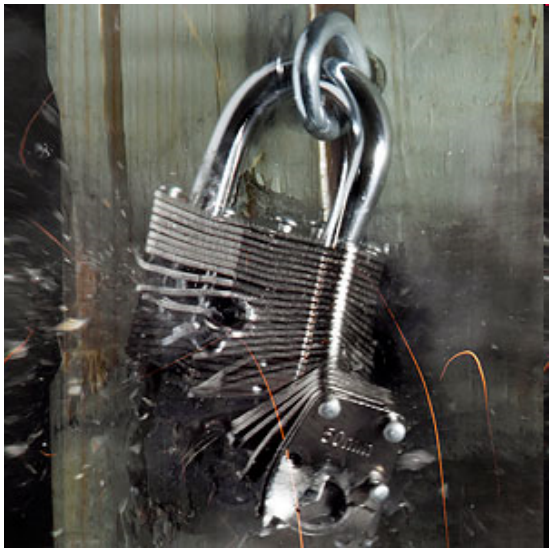
condor_shared_port

- All daemons on a machine can share one incoming port
  - Simplifies firewall or port forwarding config
  - Improves scalability
  - Running now on both Unix and Windows

USE_SHARED_PORT = True

DAEMON_LIST = … SHARED_PORT

Death to Locks!



- LOCK_DEBUG_LOG_TO_APPEND
  - Defaults to False on Unix
  - Relies on Posix O_APPEND semantics
  - Big gain w/ many running jobs
  - Only will lock on rotation
- CREATE_LOCKS_ON_LOCAL_DISK
  - Defaults to True
  - Lock files for event logs to /tmp
  - Relief for event logs in user home directories (aka on NFS)
  - Next step: only lock on rotation

Access to Cloud Services via Condor

- Cloud service handled as a job scheduler type in grid universe
- Condor can speak two cloud protocols
  - EC2
  - Deltacloud
- These cover a broad spectrum of cloud services

Cloud Protocols

- Amazon's EC2 becoming a lingua franca in cloud world
- Many cloud services speak EC2
  - Nimbus
  - Eucalyptus
  - OpenStack
  - OpenNebula

EC2 Dialects

- EC2 has two dialects: Soap and Query
  - Condor speaks Soap today
- Many cloud services only speak Query
  - So Condor will speak Query soon… currently being tested, should be released in summer

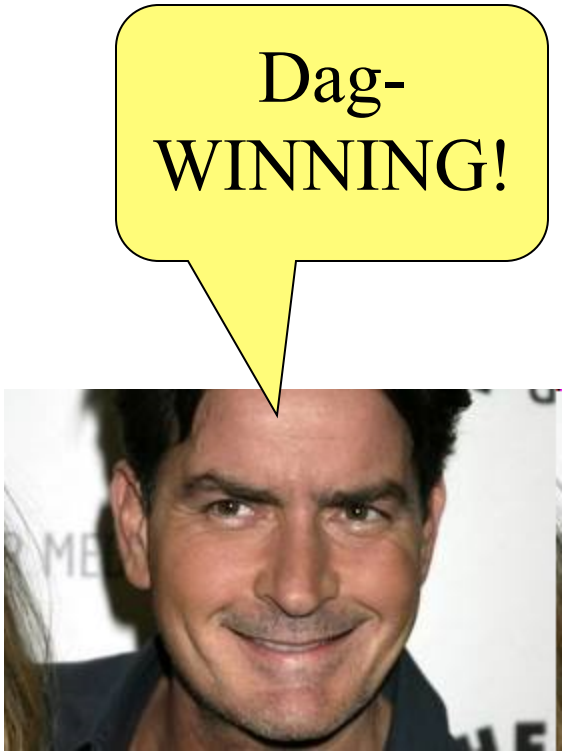
Deltacloud

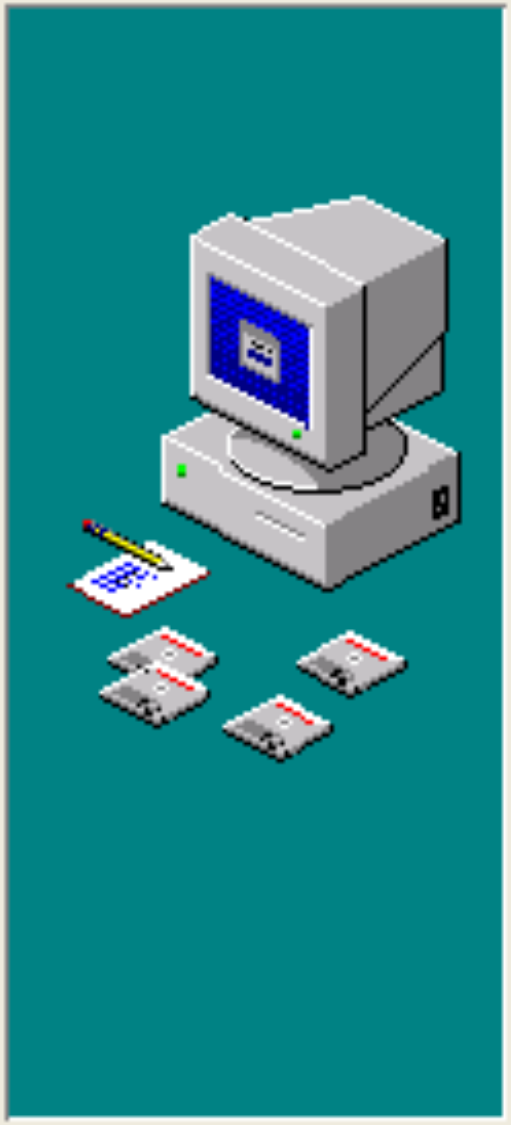
- Project sponsored by Red Hat and Apache
- Has its own simple protocol
- Translates requests into protocol of target cloud service
  - Many protocols supported
  - Wider reach than EC2 protocol (e.g. GoGrid, Rackable, RHEV, …)

Igor's Talk Last Year…

DAGMan

- Condor_hold/condor_remove of DAGMan job now works correctly (7.5.6).
- $MAX_RETRIES for script argument (7.5.6).
- A bunch of new config macros to match existing condor_submit_dag command-line arguments (7.5.6).
- Condor_hold/condor_remove of DAGMan job now works correctly (7.5.6).
- Jobstate.log file (7.5.5).
- Node status file (7.5.4).
- The new configuration variable DAGMAN_MAX_JOB_HOLDS specifies the maximum number of times a DAG node job is allowed to go on hold (7.5.4).
- Category throttles in splices can be overridden by higher levels in the DAG splicing structure (7.5.3).
- Lazy creation of submit files for nested DAGs (7.5.2).





Please read the following license agreement. Use the scrollbar to read the rest of the agreement.

Terms of License

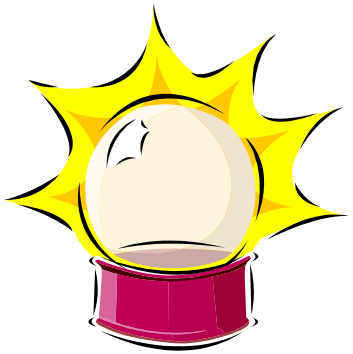
Any and all dates in these slides are relative from a date hereby *unspecified* in the event of a likely situation involving a frequent condition. Viewing, use, reproduction, display, modification and redistribution of these slides, with or without modification, in source and binary forms, is permitted *only after a deposit by said user into PayPal accounts registered to Todd Tannenbaum*

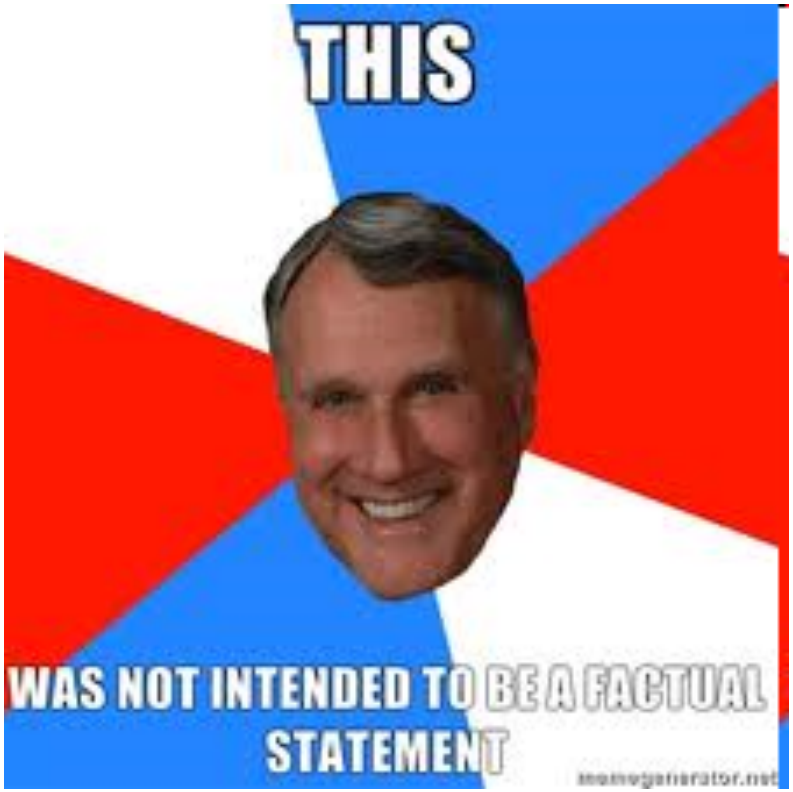
….

Do you accept all the terms of the preceding license agreement? If so, click on the Yes push button. If you select No, setup will close.

< Back | Yes | No

Crystal Ball Legend





Very Likely

Likely

Fingers Crossed

What to do, what to do…

- Talk to collaborations
- Prioritize, categorize
- Pick
- Plan
- Implement

considerable time on this item. Blue items are strong candidates for moving forward; red items are critical and/or already promised.

Scheduling

Easy (days)

7b dynamic license management (needed for xrootd overflowing, not sure what the timeframe is... 6 months?) – [CMS:4,5]

Med (weeks)

1a Better statistics centrally gathered and presented on workload, scheduling effectiveness (aka what is really happening in B240?). [BaTLaB:+4, CMS:+1 - #1917, CHTC, LIGO, RH:+5, OSG:???]

1c Centralized slot provisioning [BaTLaB:+3, CHTC, maybe LIGO, RH:+3, CMS:1, OSG:TBD]

1f HTPC work, e.g. a lack of a coherent policy for balancing how much of the pool is devoted to single-core vs. full-machine jobs [CMS:+5!!!!!, OSG:TBD (Igor +5), LIGO - esp if it involves dynamic slots, RH:+3, BatLab:0]

1g* Checkpoint, preempt-resume - do something re standard universe [LIGO:+5, CHTC Engage - Cui, DePablo, LMCG, RH:+1, CMS:0, OSG:0]

Long (months)

1b* More -scheduling- at the schedd, not just matchmaking [RH:0, CMS:-2!!!, BaTLab:0, OSG:???]

1d GPUs [LIGO, OSG(nanohub):TBD , CHTC Engage - ChemE, Chemistry, RH:+5, BatLaB:0, CMS:0]

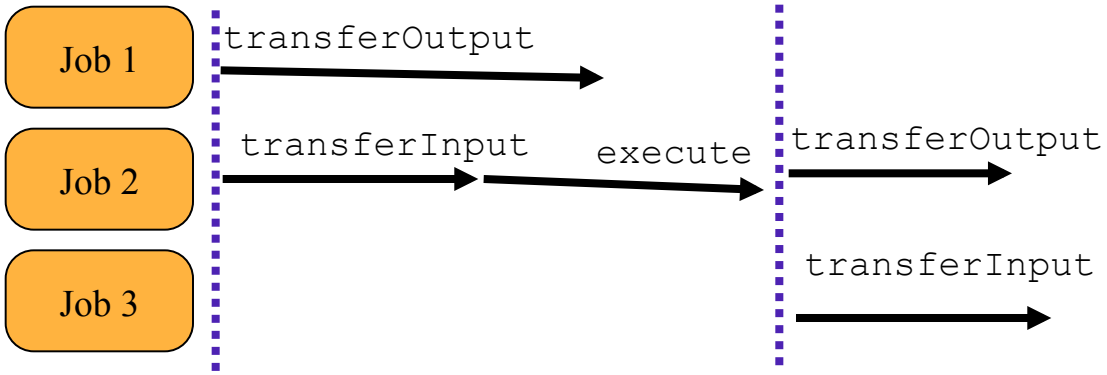
1e Power - power budgets [Purdue, Liverpool, Cisco/UCS?, Embraer, Excelon?, JC?, RH:+4 if budgets could spill over into other things as well, CMS:0, OSG:0, BaTLaB:1]

Brewing in the kitchen…

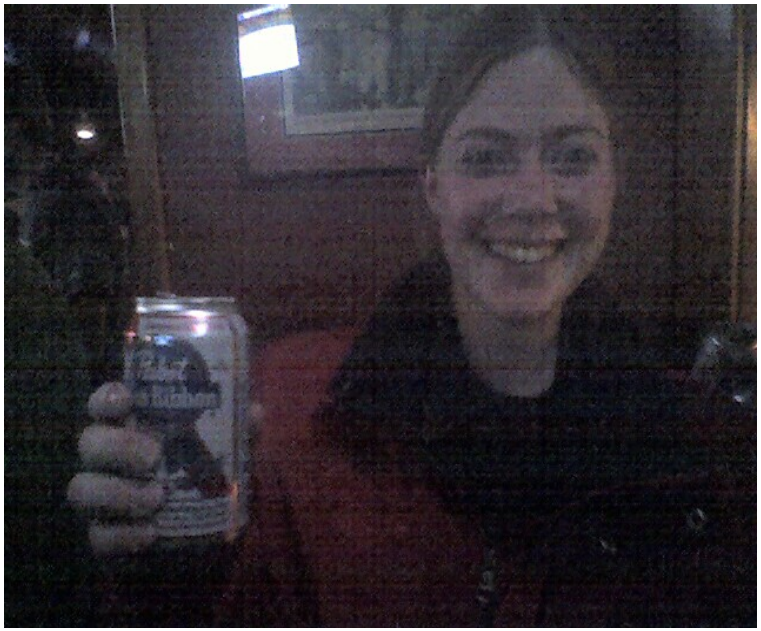
- Vanilla Universe adds ability for
  - Checkpoint – Condor *or* DMTCP (Gene!)
  - Remote I/O (Starter is Chirp Server)
- Refresh Integration w/ EC2
  - Restful ("Query") Interface
- Improved user accounting & isolation (thanks Brian B!)
- Asynch Transfer of Job Sandbox (thanks Parag!)

Asynchronous sandbox transfer

Interleave output sandbox transfer and next job's input transfer/execution within same claim



Next Step: Requires multiple concurrent claim activations



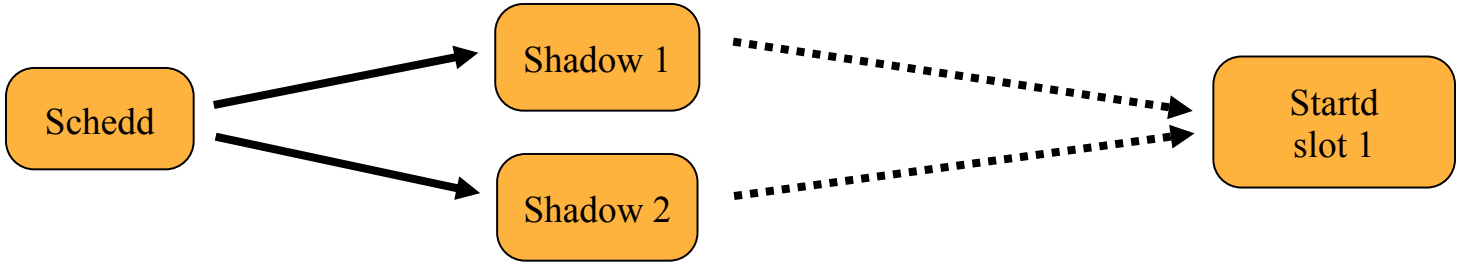
Next Step: Requires multiple concurrent claim activations

ClaimID ABCDE1234

1	2532
2	3362
3	4267
...	...

ClaimID ABCDE6246

1	2532
2	3362
3	4267
...	...



...

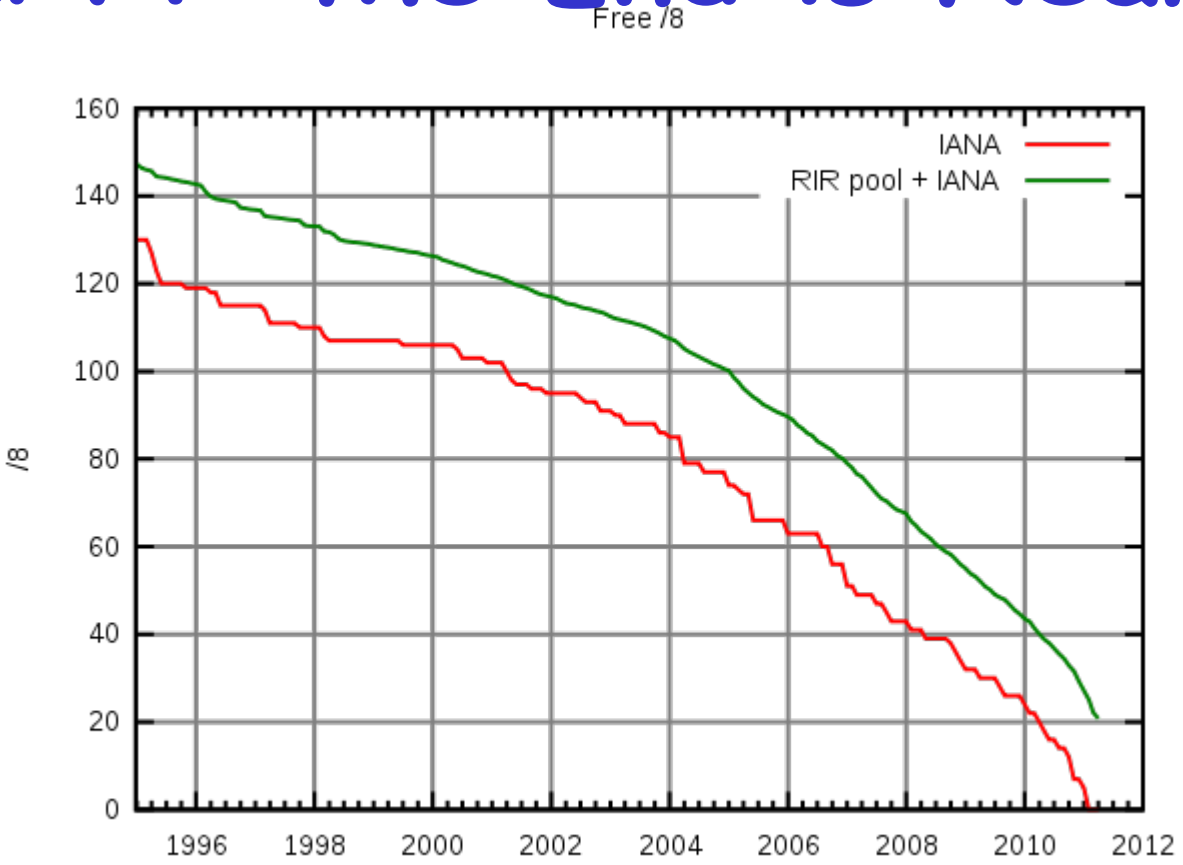
Brewing in the kitchen…

- Hang onto the claim for X seconds after job completes – great for DAGs
- Expose more workload statistics
  - Negotiator, Schedd now
  - Coming: transfer time, goodput, claim reuse, job buckets, …
- Gateway-less submission via SSH
- ExTENCI work (more)
- IPv6 support (more)

ExTENCI Project

- Automate and streamline…
  - Authoring of application-specific VM images
  - Distribution of images to many cloud services
  - Launching and management of VM instances as part of a glidein pool

IPv4: the End is Near!

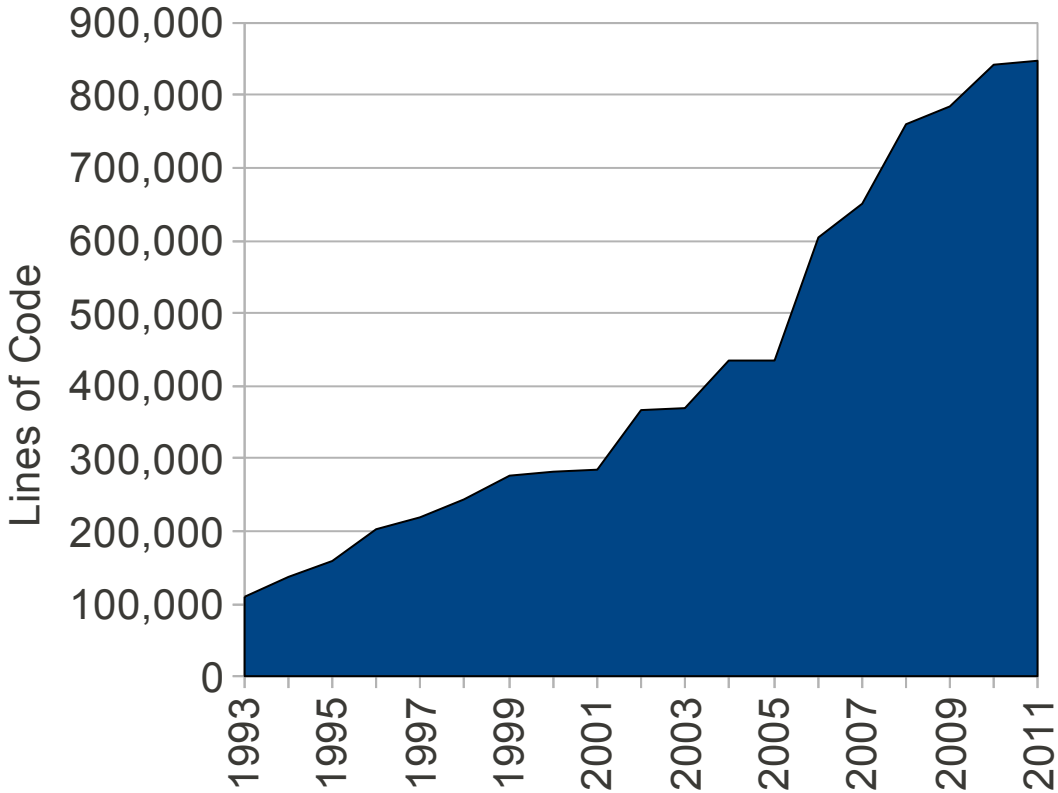


https://secure.wikimedia.org/wikipedia/en/wiki/File:Ipv4-exhaust.svg

The dangerous assumptions of IPv4

- You can fit an IP address in 4 bytes
- You can fit a human readable IP address in 15 bytes
- You only need one IP address for a host

Condor is big...



~840,000 lines of code!

- ... so we had a lot of work to do

DaemonCore helps

- (Almost) all of Condor shares a networking library: DaemonCore
- (Almost) all networking code isolated to a smaller section of code
- Still about 37,000 lines of code
- And tracking down special cases

Condor Connection Broker

- CCB broke many of the IPv4 assumptions
- Many of the old limits were removed for CCB
  - Might need multiple IP addresses for a host
  - A human readable IP address might be very long.

IPv6 - What else?

- Lots of reviewing code
- Lots of testing
- Lots of portability issues

IPv6 in 7.7.0

- Identify all places where IPv4 addresses are used
- Replace with generic address objects
- Replace IPv4 calls with IPv6 versions
- The new plumbing is in place, but no IPv6 toilets have been installed
  - Verify the new pipes don't leak!

IPv6 in 7.7.1ish

- Install the toilets!
- Initial support may be limited
  - IPv6 may default to disabled
  - May not support mixed IPv4/IPv6 pools
  - Want to test small changes

IPv6 in 7.8.0

- Full IPv6 support
- Mixed IPv4/IPv6 pools

- "Just Works"

More Scheduling Work

- Mechanism for balancing how much of the pool is devoted to single-core vs. full-machine jobs (or big –vs- small RAM, etc). Key Issues:
  - Will we make use of time estimates for job completion?
  - How will we choose a machine to drain?
  - How will drained machines change personality?
  - What needs to be monitored?
  - How do we decide when to initiate draining?
- What about Dynamic (repartionable) Slots?

Deeper into the Crystal Ball

- Effective transfer of large job input sandboxes
  - Leverage File Transfer plugins?
- GPU
  - Scheduling
  - Provisioning
- Negotiator Scaling
  - More Cores? Autocluser Startd? Cross-User caching? Sharding?

You all have Tige^h^h Condor-BLOOD!!!!

Thank you!

Keep the community chatter going on condor-users!