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

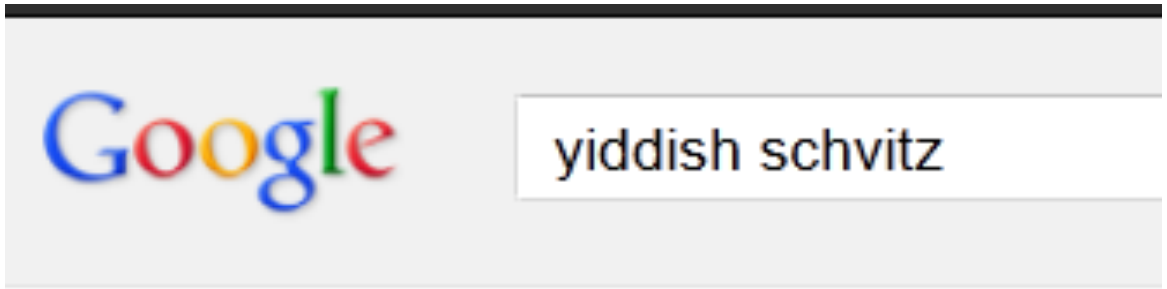
Condor Week 2012

Condor Project
Computer Sciences Department
University of Wisconsin-Madison

13 Years of Condor Week

Bar Mitzvah Edition of "What's New in Condor"

Hint:



Example:

Release Situation

- Development Series
  - Current: Condor v7.7.6 (release candidate for v7.8.0)
  - Series v7.7.x now dead, v7.9.x in four weeks.
- Stable Series
  - "Any Day": Condor v7.8.0
  - v7.6.7 will *likely* be the last v7.6.x released
  - Last Year: Condor v7.6.0 (April 19th 2011)
- 14 Condor releases since last Condor Week

Dropped Some Old OSes

- 7.8.x dropped these ports from 7.6.x:
  - RHEL 3, RHEL 4
  - MacOS 10.4

Official Ports for v7.8

- Binary packages available for
  - Windows (x86)
  - Debian 5 (x86, x86_64)
  - Debian 6 (x86_64)
  - RHEL 5 (x86, x86_64)
  - RHEL 6 (x86_64)
  - MacOS 10.7 (x86_64)
- Of course source code as well
- Continue to push into distro repositories

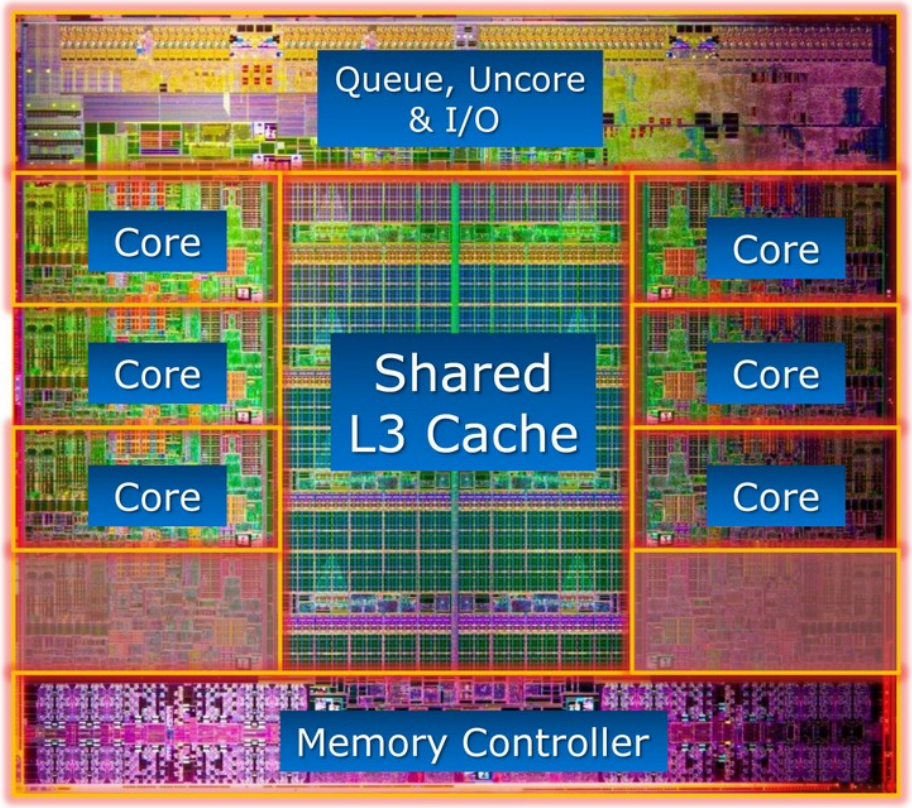
New goodies in v7.6

- Scalability enhan[illegible].)
- F[illegible]
- G[illegible]ems
- [illegible]ng Groups
- [illegible]tion on Vista/Win7
- [illegible]t "KBDD" in `Daemon_List`
- Sizeable amount of "Snow Leopard" work…

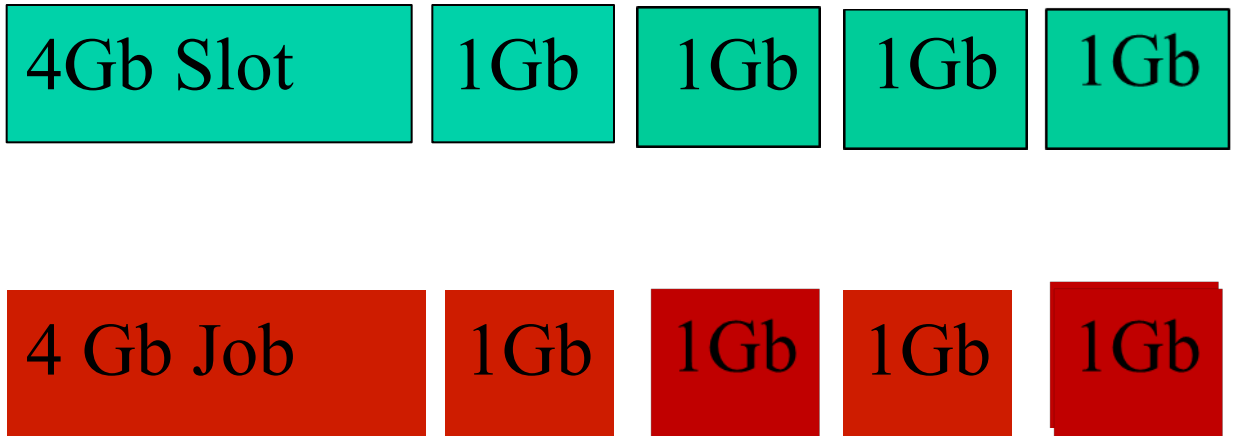
New goodies with v7.8

- Scheduling:
  - Partitionable Slot improvements
  - Drain management
  - Statistics
- Improved slot isolation and monitoring
- IPv6
- Diet! (Shared Libs)
- Better machine descriptions
- Absent Ads
- …

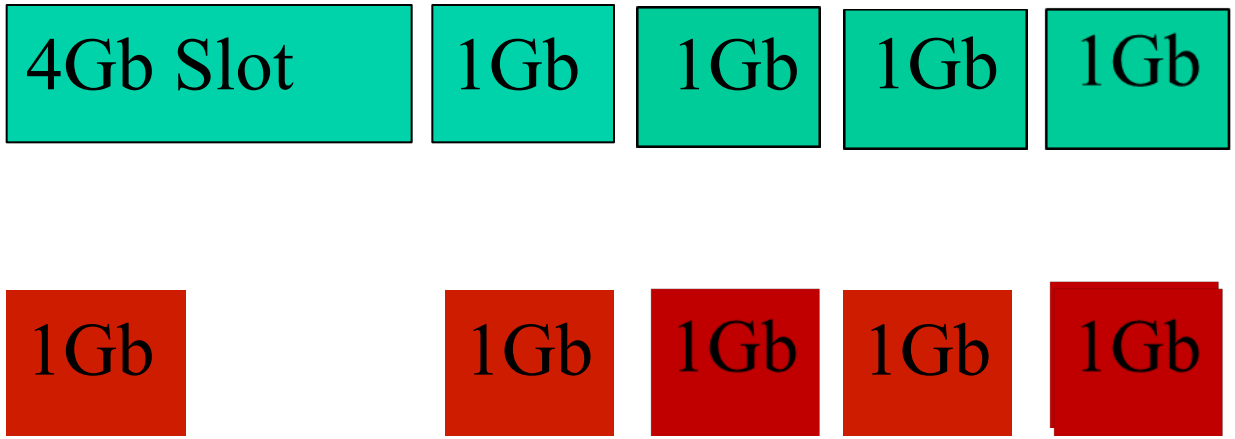
What's the Problem?



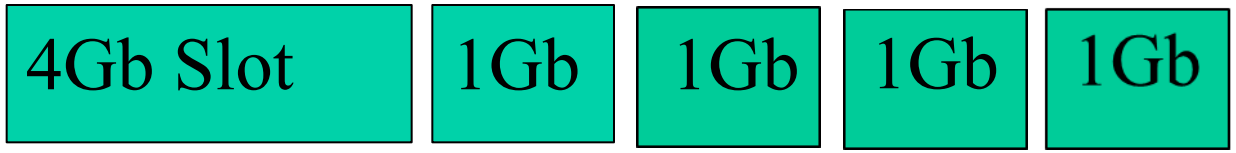
8 Gb machine partitioned into 5 static slots

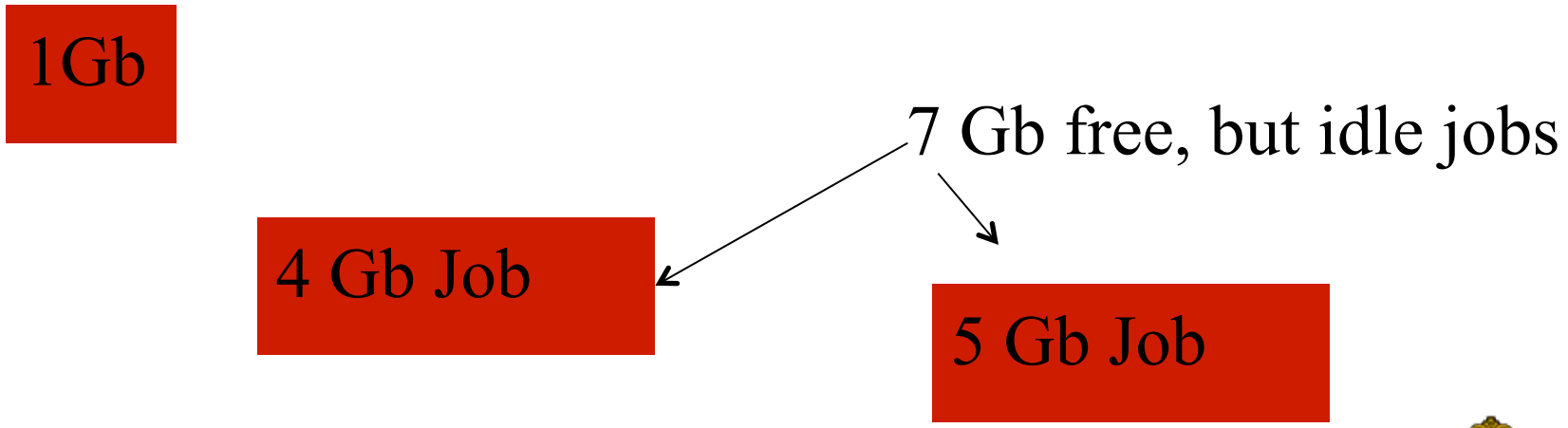


8 Gb machine partitioned into 5 static slots



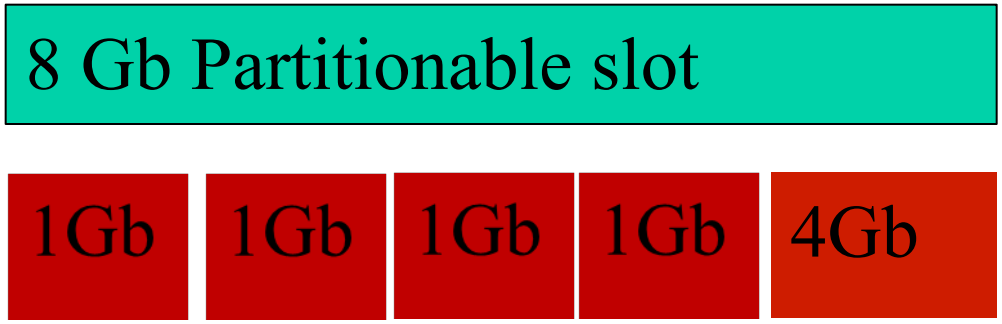
8 Gb machine partitioned into 5 static slots





The big idea

- One "partionable" slot
- From which "dynamic" slots are made
- When dynamic slot exit, merged back into "partionable"
- Split happens at claim time


8 Gb Partitionable slot
1Gb
1Gb
1Gb
1Gb
4Gb


8 Gb Partitionable slot
1Gb
1Gb
1Gb
5Gb

How to configure

```
NUM_SLOTS = 1
NUM_SLOTS_TYPE_1 = 1
SLOT_TYPE_1 =  cpus=100%
SLOT_TYPE_1_PARTITIONABLE = true
```

…and to submit

```
Request_Memory = 1024
queue
```

All this was in 7.2

- But there were downsides in v7.2...
  - Slow – only one dynamic slot created per negotiation cycle
  - Parallel universe w/ partitionable slots was a little meshuggah*
  - Dedicated slots users broken
  - Selection of dynamic slots sizes tricky
  - Fragmentation leads to starvation

Solutions in v7.8

- Slow matching → Schedd creates dynamic slots by claiming leftovers, no matchmaker micromanagement.
- Broken for parallel universe → Fixed.
- Dedicated slots users broken → Fixed.
- Selection of dynamic slots sizes tricky → Quantize @ Startd (Knob for That ™)
- Fragmentation leads to starvation → Added first class draining support and a defragmentation daemon

Statistics

Todd / Greg: We have all these thoughts to improve scheduling…
Macher* : First tell me quantitatively how well the current scheduling policy performs today and how you'll measure change.
Todd / Greg: We'll have to get back to you….
Macher : I thought so…

- Effort to Collect and Expose statistics previously buried in the Condor daemons
- Counters, sliding windows, and histograms on job mix, run times, data transfer times, goodput, badput, …

IPv6

- It works! To try it, just add to config:

  ENABLE_IPV6 = TRUE

  NETWORK_INTERFACE = 2607:f388:1086:0:21b:24ff:fedf:b520

- Buy' ngop!* … but there are limitations:
  - IPv4 *or* IPv6, not both
  - No Windows support
  - Security policies can't refer to IPv6 addresses Hostnames still work fine
  - Can only use one network interface

Improved Slot Management

- Control Group (cgroups) provides better process tree tracking and metrics (RHEL6+)
  - Imagesize, ResidentSetSize, PssSetSize → MemoryUsage
- Per-slot file system mounts
  - Example: Each slots gets its own view of /tmp via knob MOUNT_UNDER_SCRATCH
- Run jobs in a chroot jail
  - NAMED_CHROOT knob, jobs sets RequestedChroot

Heterogeneous Help

Sometimes 'OpSys' = LINUX isn't descriptive enough

- OpSysAndVer ("RedHat5")
- OpSysLongName ("Red Hat Enterpri…")
- OpSysMajorVersion ("5")
- OpSysName ("Lion")
- OpSysShortName ("RedHat")
- OpSysVer (501)

Some hidden gems…

- condor_ssh_to_job supports X11 forwarding via the new -X option.
- New grid types: SGE, Globus 5.x
- New ClassAd functions: pow(), quantize(), splitUserName(), splitSlotName()
- EC2 grid support updated: Query API, ssh key flexibility, uses Amazon 2phase commit
- HGQ: autoregroup, condor_userprio tool group options (-grouprollup, -grouporder)
- Condor can kvetch* about disappearing machines

Absent Ads

- Condor can remember what machines "should" be in a pool.
- If a machine ad goes kaput* without being invalidated, it can be stored to disk as "absent," instead of forgotten.
- Useful for heterogeneous pools.

condor_status -absent

```
Name                                OpSys      Arch       Went Absent Will Forget

slot1@exec-9.batlab.org             LINUX      X86_64      5/1  17:04  5/31 17:04
slot2@exec-9.batlab.org             LINUX      X86_64      5/1  17:04  5/31 17:04
slot3@exec-9.batlab.org             LINUX      X86_64      5/1  17:04  5/31 17:04
slot4@exec-9.batlab.org             LINUX      X86_64      5/1  17:04  5/31 17:04
slot1@exec-14.batlab.org            WINDOWS    X86_64      5/1  17:04  5/31 17:04
                     Total Owner Claimed Unclaimed Matched Preempting Backfill

        X86_64/LINUX    12     0       0        12       0          0        0
      X86_64/WINDOWS     1     0       0         1       0          0        0

               Total    13     0       0        13       0          0        0
```

Configuring Absent Ads

- ABSENT_REQUIREMENTS
  - Filter which ads you want Condor to remember
- COLLECTOR_PERSISTENT_AD_LOG
  - (used to be OFFLINE_LOG) File on disk to store persistent data – absent ads survive a collector restart/upgrade
- ABSENT_EXPIRE_ADS_AFTER
  - Defaults to 30 days

DAGMench* advances

- New KeepClaimIdle attribute and **DAGMAN_HOLD_CLAIM_TIME**
- **FINAL** node
- Always run the POST script, unless explicitly asked not to
- **PRE_SKIP** to allow PRE scripts to short-circuit the rest of the node (node succeeds)

DAGMench advances, cont

- Propagation of DAG priorities to children, sub-DAGs and job priorities
- DAGMan halt file: less drastic way to control DAGMan
- **DAGMAN_USE_STRICT** flag (like -Wall)
- Rescue DAGs are "partial" dags (edits to original DAG work for rescue)

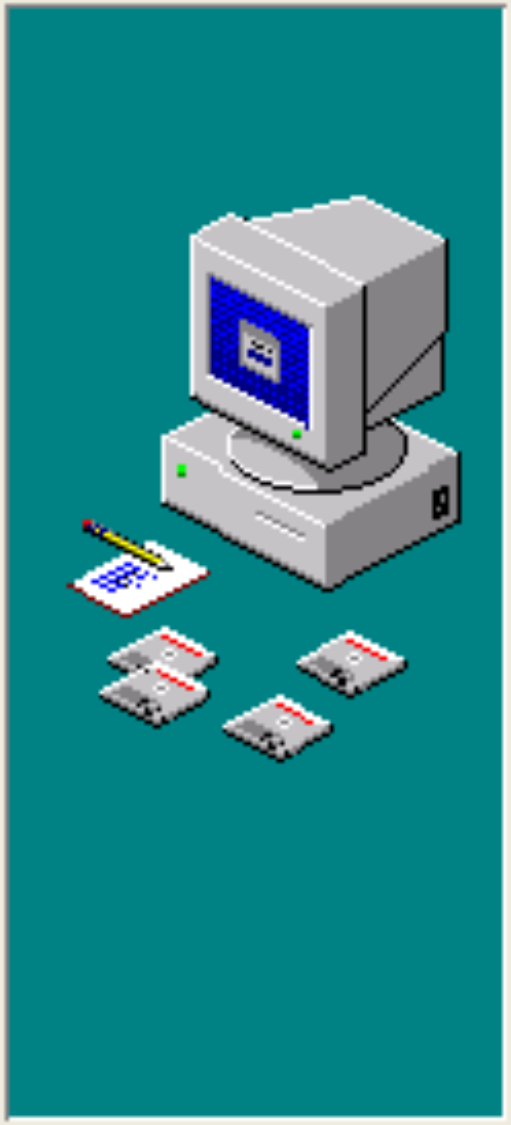
Always ongoing…

- Performance
  - Batch commands to Cream grid type
  - Speed up matchmaking for machines with many slots O(30+)

  [GLOW negotiation cycle went from 25 minutes to 4 minutes]
- Packaging
  - Debian forced us to Shared Libraries

  [from ~140 GB to ~15 GB !]


FOR THE LAST TIME MYRON, YOU WILL NOT DO YOUR HAFTARAH TO A RAP BEAT !!!!
Hunau © J 2008



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

Who do…

- Talk to commu
- Priorit categ
- Plan (D Docum
- Implen

https://condo ew?tn=2961

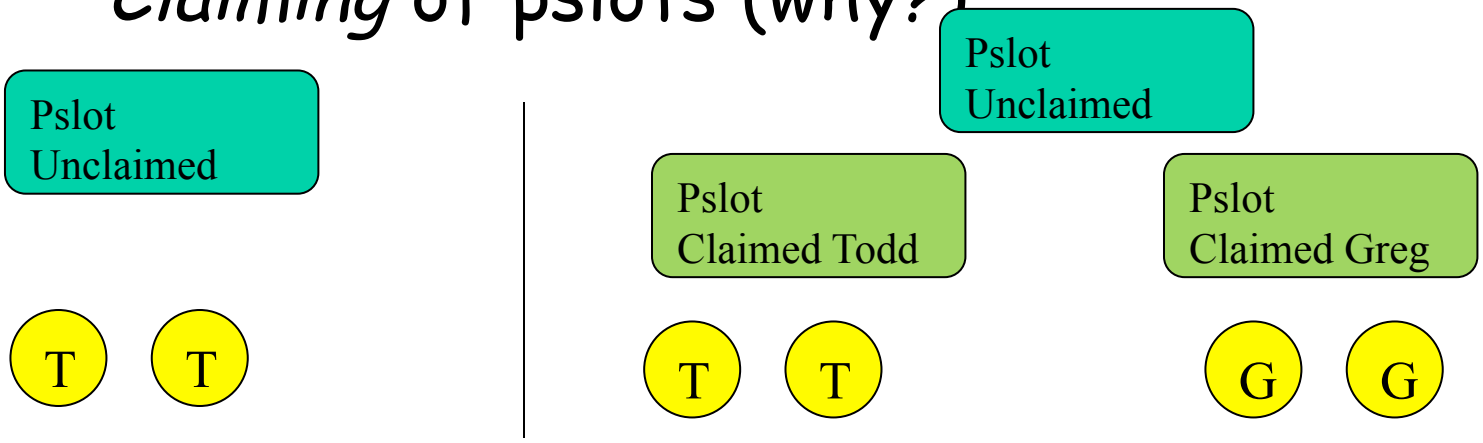
Work on GlideIn Infrastructure

(Dynamically deploy Condor on the fly)

- Large number of jobs on OSG are Condor GlideIns
- Pilot Streams instead of Pilot Jobs
  - Today: Factory submits 1,000 jobs, and must keep submitting more as they exit
  - Tomorrow: Factory says "Maintain 1,000 jobs", and jobs resubmit themselves at the site
  - Reduce load at factory and front end, moves towards resource provisioning
  - Job Instances, Condor-C scaling, more security
- What happened tools

Resource Management on Execute Side

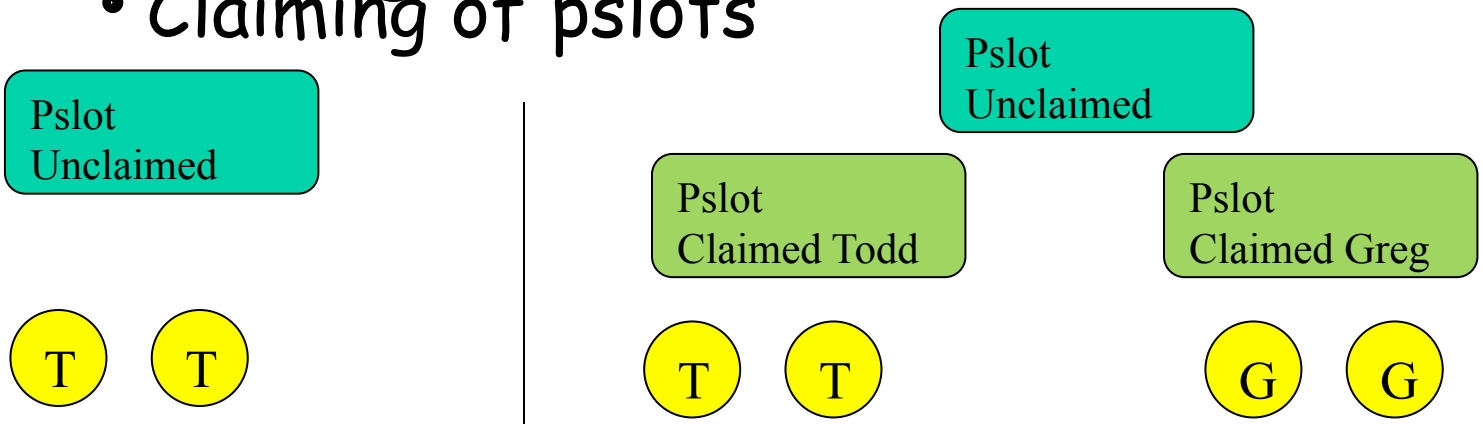
- Continue work on partitionable slots
  - Make tools more pslot smart (ex: -run)
    - Bi-directional condor_q -analyze
  - Work-fetch and pslots all fermished*
  - *Claiming* of pslots (why?)



Resource Management on Execute Side

- Continu… …ble slots
  - Make … …(ex: -run)
    - Bi-di… …e
  - Work-… …rmished*
  - Claiming of pslots

Schedd orchestrates both creation and destruction of dslots



Resource Management on Execute Side, cont.

- GPUs
  - Support "out of the box" for CUDA : discovery, monitor, provision, isolate, validate
- Continue Job Sandboxing work
  - On Linux: Continue to leverage cgroups esp RAM usage isolation, network isolation via network namespaces
  - Use JobObjects, IO manager on Windows

Resource Management on Submit Side

- Local Universe Jobs managed by a co-located Startd
- Sandbox Movement
  - Offload sandbox movement from the submit machine
  - Leverage HTTP caching in a more used-friendly manner
- Optimize Shadow usage

Oy vey!*
Last but not least…

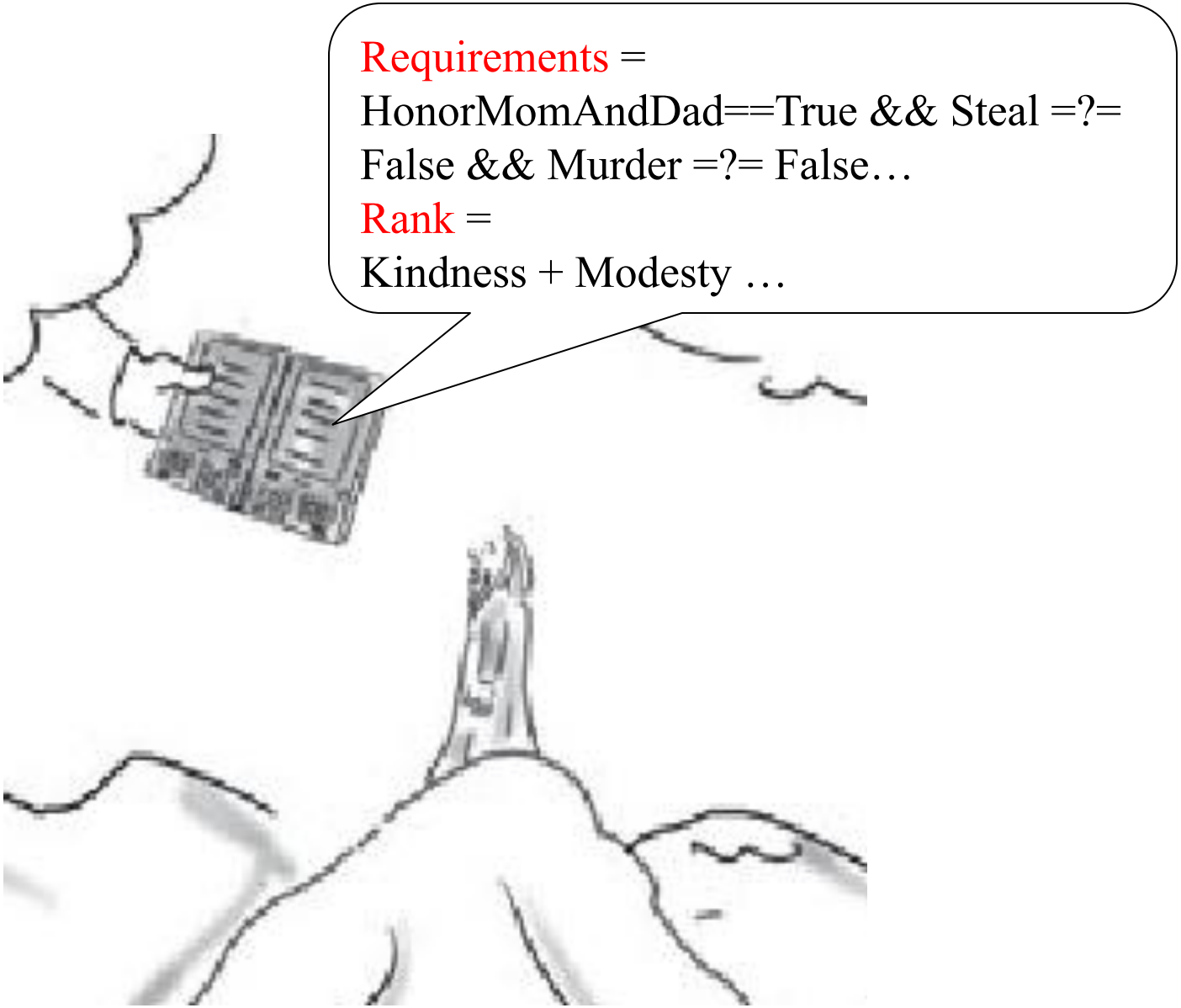
- ClassAd Scalability: (1) Memory, (2) Performance
- Heard earlier about: Bosco Work, UCS work
- Ckpt in Vanilla Universe
- Overlap transfer of sandbox results with launch of the next job

lengel

- With Cond
  schedd sta

Thank you!

Keep the community chatter going on condor-users!


Requirements =
HonorMomAndDad==True && Steal =?=
False && Murder =?= False…
Rank =
Kindness + Modesty …