



# HTCondor: The community and services

Matthew Farrellee  
Software Engineer

1 May 2013

# Some history

- Seventh consecutive HTCondor Week presentation from Red Hat
- Foundation, back in the mid-aughts
  - Red Hat & HTCondor Project sign strategic agreement
    - HTCondor 7.0 released with source
    - Red Hat Madison office opened
- Collaborative development, since the foundation
  - Joint features and software hardening (QE & Sec)
- Features, past present and future
  - VMu, concurrency limits w/ groups, EC2, partitionable slots, git, statistics, deltacloud, hierarchical group quotas, cmake, machine local limits, classad and collector performance, consumption policy...



# What is MRG?

- Red Hat Enterprise MRG - Messaging, Realtime, Grid
  - Messaging - high-performance, reliable, AMQP messaging
  - Realtime - high-performance, deterministic, low-latency
  - Grid - high-performance and high-throughput computing
- Released version 2.2 and 2.3 over last year
  - Aviary ISV APIs, EC2 enhancements, increased performance, integrated configuration, enhanced management, Wallaby shell, Plumage, Cluster Suite (HA) integration, Wallaby - Cumin integration, LDAP & Kerberos, HTML5 UI, negotiation and accounting enhancements, Aviary HA, Mongo datastore, cgroups
- MRG 2.3 is based on Condor 7.8 (a rebase, significant!)
- Bleeding edge packages available in Fedora



# Open source status

- Apache License, Version 2.0
  - <http://research.cs.wisc.edu/htcondor/license.html>
- Ticketing system and community wiki
  - <https://htcondor-wiki.cs.wisc.edu/>
- Source code repository
  - <https://github.com/htcondor/htcondor>
- Downstream packaging in Fedora, Debian, Red Hat Enterprise Linux, MacPorts
- Accepting contributions small and large



# Open source status: Tickets

- This year (26 Apr 2013)
  - 3,582 total (+636)
  - 1,522 defects (+288)
  - 1,359 enhancements (+250)
  - Over 1600 defects or enhancements resolved
- Last year (29 Apr 2012)
  - 2,946 total (+827)
  - 1,234 defects (+305)
  - 1,109 enhancements (+512)
  - Over 1000 defects or enhancements resolved
- Individual and organization data is difficult to collect
- By some number of individuals
- Someone want to write a report that joins ticket assignee with email and aggregates?
- From some number of organizations
- Individual and organization data is difficult to collect



# Open source status: Commits

- This year (26 Apr 2013)
  - 2,326 in last year
  - About 6 per day for 365 days
  - Only master branch
- By 28 individuals (down 5 from previous)
- From CHTC, Red Hat, U of Nebraska-Lincoln, U of Chicago, UCSD (same, swap SORS for UCSD)
- Last year (29 Apr 2012)
  - 2,971 in last year
  - About 8 per day for 366 days
  - Only master branch, more on topics
- By 33 individuals (down 4 from previous)
- From UW, Red Hat, U of Nebraska-Lincoln, U of Chicago, SORS (down 1 from previous)



# Open source status: Committers

- This year (to 26 Apr 2013)
  - 561 Karen Miller
  - 360 Jaime Frey
  - 243 John (TJ) Knoeller
  - 135 Greg Thain
  - 106 Nathan W. Panike
  - 105 Timothy St. Clair
  - 94 Zach Miller
  - 87 Peter MacKinnon
  - 85 Dan Bradley
  - 71 Todd L Miller
  - Scot, Derek, Brian, Matt, Z, Kent, Sam, Erik, Alan, Marco, Todd, Rob, Igor, Matyas, Guilherme, Florian, Dan, Jon
- Last year (to 29 Apr 2012)
  - 444 Greg Thain
  - 317 Karen Miller
  - 316 John (TJ) Knoeller
  - 278 Jaime Frey
  - 226 Dan Bradley
  - 210 Matthew Farrellee
  - 131 Timothy St. Clair
  - 128 Nathan W. Panike
  - 126 Peter MacKinnon
  - 114 Erik Erlandson
  - 114 Alan De Smet



## Question and plan from last year

- Question: How can we stably improve the developer community?
  - Answer: The condor way - through an organic and phased approach
- Plan: Develop a strategy and implement portions by HTCondor Week 2013
  - Phase 0 – gather input from community
  - Phase 1 – get house in order and low hanging fruit
  - Phase 2 – expand based on experiences





## Phase 0 – Gather input

- Gathered general and targeted community input
- Top questions: What features are coming? How can I become a committer? How do I get my contribution in? What can I work on?
- Top suggestions: Maintain a public roadmap, can be approximate. Provide regular releases. Make repository public. Give me a test suite. Give me credit.



# Phase 1 – Get house in order & low hanging fruit

- Getting house in order
  - Predictability - establish monthly release cadence
  - Transparency - tracking contents of releases via tickets
    - <https://htcondor-wiki.cs.wisc.edu/index.cgi/wiki?p=ReleaseHistory>
  - Engage - define processes and expectations
- Low hanging fruit
  - Transparency - no more private repository
  - Inform - established technology blog feed
    - <http://htcondor.github.io/planet/>
  - Communicate - transition to htcondor-devel and #distcomp
    - Public list traffic increasing, private decreasing, still small
    - #distcomp active w/ discussion, project notification and user questions



## Phase 1.5 – Areas to improve

- Test suite (test suite represents 1.8% of code changes)
- Tickets need attention
  - Many in limbo or remain open but released or not update as progress is made
- Reviews are not in standard community process (26 Apr 2013 - 30 outstanding reviews)
  - <https://htcondor-wiki.cs.wisc.edu/index.cgi/rptview?rn=104>
- Only consistent community notification is when a release happens
- Missed a few monthly releases



## Phase 1.5 – A project roadmap

- Nearly everyone providing input asked for a project roadmap
  - For user & developer planning, for understanding, for expectations and for engagement
- What's a sufficient roadmap?
- Proposal
  - Assuming time based point releases (x.y.Z) (1 per month) culminating in a series release (x.Y) (1 per year)
  - A roadmap is two things
    - High level themes for the series
    - Bug & enhancement tickets assigned to the current and future point releases



# Consider - HTCondor as a Service Manager

- Managing services is an ongoing problem in IT
- Many tools exist to perform management in a static, or slow moving fashion
- IT is changing, becoming more dynamic and self-service
- Having ways to handle scheduling, accounting, policy, isolation, SLA, monitoring, coordination, management is increasing in priority
- Sounds a lot like what HTCondor provides, so apply it!
- Not a new idea (original: glide-in), but not formalized
- Demo: Dynamically create Apache Hadoop clusters
- Currently, services wrapped in controls scripts that publish static properties & dynamic statistics, and handle life-cycle



**I still want to hear your opinions on  
HTCondor's developer community.**

**[matt@redhat.com](mailto:matt@redhat.com)**

**<http://spinningmatt.wordpress.com>**

**Also: Still hiring**

