Release Management and New Capabilities in Chirp

Patrick Donnelly, University of Notre Dame
CCL Workshop, June 2012
Release Wrangler
What is CCTools?

Collection of Distributed Computing Tools

- AllPairs/Wavefront
- Chirp/Parrot
- Makeflow/WorkQueue
- Sand
Previous Release Methodology

- Tagged /trunk every few months.
- Contained features and bug fixes.
- Version number bumps mostly arbitrary.
- 3.4.2 --> MAJOR.MINOR.RELEASE
New CCTools Release Philosophy

- Versions are now MAJOR.MINOR.REVISION
- REVISION changes may not break API/ABI compatibility --> Bug Fix
## Latest builds from: trunk

- cctools-2116-i386-darwin-8.11.tar.gz
- cctools-2106-i686-redhat5.tar.gz
- cctools-2116-i686-redhat6.tar.gz
- cctools-2116-source.tar.gz
- cctools-2116-x86_64-redhat5.tar.gz
- cctools-2116-x86_64-redhat6.tar.gz

## Revisions

<table>
<thead>
<tr>
<th>Id</th>
<th>Author</th>
<th>Message</th>
<th>Branch</th>
<th>Commit Time</th>
<th>Tarballs</th>
<th>Logs</th>
</tr>
</thead>
</table>
| 2116| dpandir| a. Fix typo. b. Add reference to Python API.      | trunk  | Thu, Jun 07, 2012 04:31 PM | • i386-darwin-8.11  
    • 1666-redhat5  
    • 1666-redhat6  
    • source  
    • x86_64-redhat5  
    • x86_64-redhat6 | summary  
    • condor.submit  
    • condor.log  
    • 1666-darwin-8.11  
    • build.summary  
    • build.log  
    • 1666-redhat5  
    • build.summary  
    • build.log  
    • source  
    • build.summary  
    • build.log  
    • x86_64-redhat5  
    • build.summary  
    • build.log  
    • x86_64-redhat6  
    • build.summary  
    • build.log |
| 2115| dpandir| Update with the -j option that submits workers as a job array. | trunk  | Thu, Jun 07, 2012 03:27 PM | • i386-darwin-8.11  
    • 1666-redhat5  
    • 1666-redhat6  
    • source  
    • x86_64-redhat5 | summary  
    • condor.submit  
    • condor.log  
    • 1386-darwin-8.11  
    • build.summary  
    • build.log  
    • 1666-redhat5  
    • build.summary  
    • build.log  
    • source  
    • build.summary  
    • build.log  
    • x86_64-redhat5  
    • build.summary  
    • build.log  
    • x86_64-redhat6  
    • build.summary  
    • build.log |
Patrick Donnelly

**Patrick Donnelly**  Apr 17, 2012  -  Public

My reign of terror as Release Wrangler is about to begin!

Thanks to +Michael Albrecht, +Dinesh Rajan, and +Peter Bui for this awesome Birthday Gift!
Downloads

The cctools package contains Parrot, Chirp, Makeflow, Work Queue, SAND, and other software. This software is Copyright (C) 2004-2011 The University of Notre Dame and licensed via the GNU General Public License V2.

Report bugs or other problems to our mailing list.

Users at Notre Dame click here for the local installation.

Download Stable Version 3.5.0

cctools-3.5.0-src.tar.gz
Source Code
Red Hat Enterprise Linux 6, x86 CPU (64 bit)
cctools-3.5.0-x86_64-redhat6.tar.gz
Red Hat Enterprise Linux 5, x86 CPU (64 bit)
cctools-3.5.0-x86_64-redhat5.tar.gz
Red Hat Enterprise Linux 5, x86 CPU (32 bit)
cctools-3.5.0-i386-redhat5.tar.gz
MacOS 10.4 "Tiger", x86 CPU
cctools-3.5.0-i686-windows.tar.gz
Windows 7/Vista/XP, x86 CPU (requires Cygwin installation)

Download Current Unstable Version:

cctools-current-src.tar.gz
Source Code
Red Hat Enterprise Linux 5, x86 CPU (64 bit)
cctools-current-x86_64-redhat5.tar.gz
Red Hat Enterprise Linux 5, x86 CPU (32 bit)
cctools-current-i686-redhat5.tar.gz

Software Engineering Tools

- CCTools Autobuild System
- CCTools SVN Repository Browser
- Download Older Versions
Outreach

Patrick Donnelly

Patrick Donnelly  8:07 AM - Public
The Cooperative Computing Lab is pleased to announce the release of version 3.5.0 of the Cooperative Computing Tools, including Parrot, Chirp, Makeflow, WorkQueue, SAND, All-Pairs, and other software.

LISTSERV 15.5 - CCTOOLS Archives »
View: Next Message | Previous Message Next in Topic | Previous in Topic Next by Same Author | Previous by Same Author Chronologically | Most Recent First - Proportional Font | Monospaced Font. Options...
Makeflow is a workflow engine for executing large complex applications on clusters, clouds, and grids. It can be used to drive several different distributed computing systems, including Condor, SGE, and the included Work Queue system. It does not require a distributed filesystem, so you can use it to harness whatever collection of machines you have available. It is typically used for scaling up data-intensive scientific applications to hundreds or thousands of cores.
New Capabilities in Chirp...
Chirp

- Distributed File System for use on a Grid
- Exports file system on host
- User-level filesystem
- Authentication mechanisms
  - Grid Security Infrastructure
  - Kerberos
  - Hostnames
  - Unix
  - Tickets (new!)
- Secure Authorization through Access Control Lists

**Chirp Network Protocol**

**Client**
- User Application
- Parrot
- libchirp

**Server**
- Chirp Server
- Unix Filesystem
Chirp with Backend Storage

- Supports a filesystem abstraction layer.
- Currently Available:
  - Hadoop
  - Locally mounted (Unix)
  - Chirp (Chirp mounting Chirp; how perverse)
- Why?
Hadoop Distributed Filesystem

- Java open source implementation of the concepts in the Google File System.
- Offers very large file storage on the order of terabytes.
- Replicated file storage.
- Active Storage and Map-Reduce.
- Streaming data access.
Problems with using Hadoop on a Grid

- Java API/C API
- Carry around JVM +- FUSE
- No authenticated access
- Tight coupling of JVM with Hadoop versions
Authentication in Chirp

- Globus
- Kerberos
- Hostname/IP
- Unix (local)
- Tickets (new!)
What are Tickets for?

Worker

Storage Access Node

Job Submitter

Job Ticket

Job
Input Data
Executable

/code read
/data read write

/code MyApp.exe /data/input1.txt > /data/output1.txt
What we want in a Ticket

- Available **secure** authentication mechanisms rely heavily on user interaction.
- **Desire:**
  - "Token" passed along with Job for authentication.
  - Temporary lifetime
  - Simple to setup
  - Reasonable security; minimal risk
Authentication Complications

- Users and machines do not have dedicated credentials with which to sign a token.
- Chirp servers and workers are transient.
Ticket Authentication

- Using Public Key Encryption, setup a ticket which is a private key credential for user Jobs.
- Authenticate similarly to SSH.
- Authenticated client *registers* a ticket for their current subject.
- Server maintains a list of registered tickets (public keys). Checks ACL with ticket ACL mask.
Simple Ticket Setup Steps

1. Client creates a ticket (Public/Private key pair);
2. Client registers the ticket with a storage node (send Public key);
3. Client sets capabilities of the ticket (sets ACL masks).
An Access Control List

For a directory, storage node maintains a list of tuples (ACL):

\(<\text{Subject, Rights}>\)

\textbf{Subject} is a tuple:

\(<\text{Means of Authentication, User}>\)

\textbf{Rights} is a list of primitive access methods (read, write, etc.)
Access Control List Masks

Resolution of directory authorization: Logical AND of ticket ACL mask and the rights of the subject from storage-node ACL.

Key point: A ticket cannot exceed the rights of the subject's current rights. Conversely, the ticket is still limited by the ACL mask the user assigns.
Typical Example

```
pdonnel3@cclws13 ~$ chirp -a unix disc01.crc.nd.edu:9090 ticket_create -o my.ticket / /pdonnel3/data rw1
connected to disc01.crc.nd.edu:9090 as unix:pdonnel3
Generated ticket my.ticket.
ticket 'my.ticket': successfully created with 1024 bits.
ticket 'my.ticket': successfully registered.
ticket 'my.ticket': directory '/' aclmask = 'l'.
ticket 'my.ticket': directory '/pdonnel3/data' aclmask = 'rw1'.
pdonnel3@cclws13 ~$
```

```
pdonnel3@cclws13 ~$ chirp -a unix disc01.crc.nd.edu:9090 listacl /pdonnel3
connected to disc01.crc.nd.edu:9090 as unix:pdonnel3
unix:pdonnel3 rwlda
hostname:*.*.nd.edu r1
* r1
pdonnel3@cclws13 ~$
```
pdonnel3@cclws13 ~$ chirp -a ticket -i my.ticket disc01.crc.nd.edu:9090 ls /pdonnel3
connected to disc01.crc.nd.edu:9090 as unix:pdonnel3
bash
congress-records
data
derp
foo.txt
foobar.txt
fooo.txt
foooo.txt
lots
pdonnel3@cclws13 ~$
chirp -d auth -a ticket -i my.ticket disc01.crc.nd.edu:9090
mkdir /pdonnel3/data/foo

2012/06/12 07:47:22.05 [20973] chirp: auth: ticket: registered
2012/06/12 07:47:22.05 [20973] chirp: auth: requesting 'ticket' authentication
2012/06/12 07:47:22.79 [20973] chirp: auth: server agrees to try 'ticket'
2012/06/12 07:47:22.80 [20973] chirp: auth: trying ticket 3272bd89b53c62df9ef8df89e7f2d4f737
2012/06/12 07:47:22.83 [20973] chirp: auth: receiving challenge of 64 bytes
2012/06/12 07:47:22.89 [20973] chirp: auth: succeeded challenge for 3272bd89b53c62df9ef8df89e7f2d4f737
2012/06/12 07:47:22.89 [20973] chirp: auth: successfully authenticated
2012/06/12 07:47:22.89 [20973] chirp: auth: reading back auth info from server
2012/06/12 07:47:22.89 [20973] chirp: auth: server thinks I am ticket:3272bd89b53c62df9ef8df89e7f2d4f737
connected to disc01.crc.nd.edu:9090 as unix:pdonnel3
pdonnel3@cclws13 ~$ chirp -a ticket -i my.ticket disc01.crc.nd.edu:9090
rmdir /pdonnel3/data/foo
connected to disc01.crc.nd.edu:9090 as unix:pdonnel3
couldn't rmdir: Permission denied
pdonnel3@cclws13 ~$
Future Work?

Integrate Chirp URLs and Ticket Authentication with Makeflow/WorkQueue
Questions?

Website: http://www.cse.nd.edu/~ccl
Autobuild: http://www.cse.nd.edu/~ccl/software/autobuild
Chirp: http://www.cse.nd.edu/~ccl/software/chirp

Patrick Donnelly: pdonnel3@nd.edu