User Introduction to the CCLRC Condor Pool

John Kewley

November 18, 2004

1 Introduction

Condor is a middleware product that enables computer resources that would not otherwise be utilised to be linked together to form resource "pool" enabling additional computations to be performed. Condor is a mature research project from the University of Wisconsin-Madison under Miron Livny.

The goal of the Condor Project is (to quote from Condor site): *"To develop, implement, deploy, and evaluate mechanisms and policies that support High Throughput Computing (HTC) on large collections of distributively owned computing resources. Guided by both the technological and sociological challenges of such a computing environment, the Condor Team has been building software tools that enable scientists and engineers to increase their computing throughput"*

This document describes the basics you will need to know for utilising the CCLRC Condor Pool. Since there is already a substantial user guide available from the Condor website which contains fairly detailed descriptions of the various mechanisms, this document will constrain itself to describing the way things are setup in our Pool and provide the essential information you would need to know when offering your machine to the Pool and/or utilising it for your computations.

The latest Condor manual is available at `http://www.cs.wisc.edu/condor/manual/v6.6`

2 The Basic Rules

Resource sharing If you want to use others' resources, you must provide some of your own resources to the Pool for the use of others.

High Availability Resources that are turned off every night are not very useful. The resources "loaned" to the pool should have high availability.

Every little helps If seldom used resources are always turned on, then let's add them to the pool so others can make use of them. We'll worry about the "Well who would want to use it?" questions later. If, of course, the machine is hardly ever turned on, then there is unlikely much to gain by adding it to the Pool.

Non-reliance Since many of these resources are personal workstations, there should be no insistence that certain resources are always available. If the owner wants to reboot his machine, then that is fine, jobs will be returned to the job queue for rescheduling.

3 Impact on resources

In what way will Condor interfere with the normal running of your resource?

Intrusion : As Condor jobs run at the lowest possible priority there should be minimal noticeable impact. The idea is that the jobs make use of otherwise-unused cpu cycles. If it can be shown that even this impact is/might be unacceptable (mission critical machines for instance), it is possible to configure Condor so that jobs only run outside these hours, or when the machines are idle.

Security : Since the machines will be inside the site firewall, only internal users will be able to run jobs.

Windows : Condor can be installed by the machine "owner"; normal Administrator privilege is all that is needed. Remote jobs running on that machine will be run as a special user "condor". Assuming a default DL setup, this gives permissions to do very little, indeed far less than were someone to physically logon on that workstation. Care must be taken, however, that any personal/sensitive files are protected (for example, by default, there is no protection on files in `E:` on our Windows 2000 and Windows XP machines).

Linux : The installation process will setup a user `condor` and install everything into `/opt/condor`. `root` access is required for this initial installation.

Once condor is installed, several condor dæmons will be run from `root` (automatically from `rc.d`), but with effective UID of `condor`.

Jobs will run as user `nobody`, so only world accessible files and directories can be accessed. Condor does a significant amount of work to prevent security hazards, but loopholes are known to exist. Even with access solely as user nobody, a job could (maliciously or carelessly) fill up /tmp (which is world writable) and/or gain read access to world readable files.

4 Ideal Application types

Applications most suited to submission under Condor in the CCLRC pool are jobs such as parameter sweep/searches which fire off a large number of small(ish) jobs, as opposed to a few long-running jobs. Jobs of the order of minutes and hours rather than days would fit in best with the pool we have here.

5 Installation on Linux

TBA - at present the Pool administrator will add your machines to the pool for you.

6 Installation on Windows

TBA - at present the Pool administrator will add your machines to the pool for you.

7 Universes

Various types of job are supported by Condor. Due to the differing operating profiles of these jobs, they are separated into "universes", so that for any job spec, you specify the universe in which the job should run.

Initially only "vanilla" jobs will be supported:

> The vanilla universe can be used to run other jobs that cannot be relinked against condor. This includes scripting languages (perl, tcl, etc) to binary only packages that cannot be recompiled. Jobs cannot be checkpointed nor restarted from an intermediate point; if a job dies, the machine it is on dies, or condor needs to move it for some reason, the job will be restarted from the beginning.

We will considering adding "standard" universe support once it is supported by the Condor Windows port:

> The standard universe accepts jobs that have been compiled using condor compile. These jobs will proxy most I/O calls back through the host that submitted the job. This allows for jobs to be scheduled on any machine of the same architecture regardless of the underlying filesystems that are available to it. Using this universe will also allow your job to be checkpointed, moved to another machine and restarted.
>
> Features:
>
> - executables must be preprepared by compiling with condor compile
> - jobs can have checkpointing, although this will be disabled initially
> - remote procedure calls happen at submitting node
> - jobs can be prempted and either restarted at a future time, or migrated to another machine.
>
> Restrictions :
>
> - multi-process jobs are not allowed (or at least produce undefined behaviour if used.
> - no ipcs calls

- mmap cannot be used
- restricted use of file locks
- no alarm calls (eg sleep)
- no kernel-level threads, user-level are allowed

8 Daemons / Services / Shadow Processes

The condor daemons (Windows services) are described in detail in the Condor User Guide, a brief summary is given here for reference.

- condor master runs on every Condor node
- condor startd runs on every machine on which condor jobs can run (compute nodes)
- condor schedd runs on any node on which jobs may be submitted
- condor starter (ephemeral) - one for each condor job on each machine that it runs on
- condor shadow (ephemeral) - one copy for each condor job. Runs on the scheduler node on which the job was scheduled.