Preparation of a Desktop Linux Cluster for DØ MonteCarlo Production
DØ Note # 4208

Horst Severini, Joel Snow
University of Oklahoma, Langston University
(Dated: July 29, 2003)

This note describes the steps necessary to setup an existing desktop linux cluster with Condor or any other batch scheduler installed in order to start producing DØ MonteCarlo (MC).

I. INTRODUCTION

Recently, the University of Oklahoma High Energy Physics (OUHEP) group and the Langston University High Energy Physics (LUHEP) group became part of the D0 Southern Analysis Region (SAR) [1], and have started generating D0 MC for this effort. Since it is becoming easier to generate D0 MC with automated software packages like MC_Runjob and McFarm, and Linux desktop clusters are common, we present in this note a method for getting D0 MC production running on a generic Linux desktop cluster. Most of the information in this note can be found on the OUHEP web pages for the DØ [2] and ATLAS [3] GRID efforts. In the Appendix some of the installation procedures are explained in greater detail for people who are interested in duplicating this configuration on their cluster.

The guiding philosophy in this configuration is that MC production be efficiently accomplished while not requiring a cluster dedicated to that end. The cluster should be able to accommodate a variety of batch jobs and not negatively impact interactive use. The OUHEP cluster in addition to D0 MC production is interactively used by faculty and students, and runs ATLAS grid and other batch jobs. (Even by theorists!)

II. GENERAL CLUSTER SETUP

In order to be able to run MC on a desktop cluster, it must be well organized and coupled, with a common user database, shared home disk and system and analysis areas. Shared file space is also very important for most batch queuing systems. This can be most easily accomplished with an NIS/YP (Network Information Service/Yellow Pages) server and an Autmounter (autofs), both of whose setup and configuration are explained in Appendices A and B. All of these services are provided via operating system packages. The OUHEP cluster uses Red Hat Linux 7 and the LUHEP cluster uses Debian GNU/Linux 3.

It is also very advantageous to have password-less entry from one cluster machine to another. This can be accomplished with ssh-keygen by creating public and private keys (with no passphrase) and appending the public key to the authorized_keys file in the .ssh subdirectory of the account with the shared home directory. For example:

```
ssh-keygen -t rsa      (press ENTER for password twice)
cat .ssh/id_rsa.pub >> .ssh/authorized_keys2
```

This is not only convenient, but also necessary for automated scripts which need to perform remote shell operations on all compute nodes in order to run or monitor MC production there. For example McFarm requires password-less entry.

III. BATCH SETUP

A batch queuing system is necessary to distribute submitted jobs to otherwise idle nodes. At both the OUHEP and LUHEP clusters we run Condor [4]. The OUHEP setup is described in detail in Appendix C.

In principle, any batch system should be able to accommodate MC production, as long as the MC production software supports it.

For some of the remote MC operations, the Globus [5] software package is also needed. It is convenient to install the Virtual Data Toolkit (VDT) [6], since that includes both Globus and Condor and various other Grid tools.

IV. DØ SOFTWARE SETUP

Once the cluster system software is setup properly, the next step is to install and configure DØ software. On the OUHEP and LUHEP clusters, this was done according to the installation instructions on the DØ Software Release Web pages [7, 10].

Besides the regular DØ software, special MC production software (MC_Runjob) [8] needs to be installed as well, to handle the actual MC production.

In order to process official MC requests, they have to be downloaded from FNAL via SAM [9], a data management tool, and then the generated MC uploaded to FNAL again.

Once the general DØ software setup is complete, the University of Texas at Arlington (UTA) McFarm software can be installed and run. There are detailed installation and running instructions on several web pages and documents linked from there [11, 12].

One thing worth noting is the OUHEP and LUHEP setup of the directory structure of the McFarm installation. Every worker node has to have a local scratch area where files are stored during MC production. All those directories have to be available from the central server to retrieve MC files after a job has finished.

Rather than pointing McFarm toward the local (but NFS cross mounted) areas, we made a global scratch area, from which we then fan out all the local scratch directories via soft links, so that each node will eventually only use its local scratch space, albeit via the global NFS automount path.

For the case of OUHEP the global scratch directory looks like this:

```
/scratch/users/mcfarm:
total 12
drwxr-xr-x    3 mcfarm       4096 May 20 11:59 .
drwxr-xr-x   18 root         4096 Apr 24 15:01 ..
lrwxrwxrwx    1 mcfarm         21 May 20 11:59 MB -> /raid/users/mcfarm/MB
lrwxrwxrwx    1 mcfarm         14 Apr  1 07:42 archive000_A -> scr000/archive
lrwxrwxrwx    1 mcfarm         14 Apr  1 07:43 archive001_A -> scr001/archive
lrwxrwxrwx    1 mcfarm         14 Apr 10 10:33 archive002_A -> scr002/archive
lrwxrwxrwx    1 mcfarm         14 Apr 10 10:35 archive003_A -> scr003/archive
lrwxrwxrwx    1 mcfarm         14 Apr  7 16:15 archive004_A -> scr004/archive
lrwxrwxrwx    1 mcfarm         14 Apr  7 16:18 archive005_A -> scr005/archive
lrwxrwxrwx    1 mcfarm         14 Apr  7 16:22 archive009_A -> scr009/archive
lrwxrwxrwx    1 mcfarm         14 Mar 19 13:14 cache000_A -> scr000/cache_A
lrwxrwxrwx    1 mcfarm         14 Mar 19 13:11 cache001_A -> scr001/cache_A
lrwxrwxrwx    1 mcfarm         14 Apr 10 10:33 cache002_A -> scr002/cache_A
lrwxrwxrwx    1 mcfarm         14 Apr 10 10:35 cache003_A -> scr003/cache_A
lrwxrwxrwx    1 mcfarm         14 Apr  7 16:15 cache004_A -> scr004/cache_A
lrwxrwxrwx    1 mcfarm         14 Apr  7 16:18 cache005_A -> scr005/cache_A
lrwxrwxrwx    1 mcfarm         14 Apr  7 16:22 cache009_A -> scr009/cache_A
lrwxrwxrwx    1 mcfarm         17 Mar 19 13:15 gather000 -> scr000/gath_queue
lrwxrwxrwx    1 mcfarm         17 Mar 19 13:13 gather001 -> scr001/gath_queue
lrwxrwxrwx    1 mcfarm         17 Apr 10 10:33 gather002 -> scr002/gath_queue
lrwxrwxrwx    1 mcfarm         17 Apr 10 10:35 gather003 -> scr003/gath_queue
lrwxrwxrwx    1 mcfarm         17 Apr  7 16:15 gather004 -> scr004/gath_queue
lrwxrwxrwx    1 mcfarm         17 Apr  7 16:18 gather005 -> scr005/gath_queue
lrwxrwxrwx    1 mcfarm         17 Apr  7 16:22 gather009 -> scr009/gath_queue
drwxrwxr-x   10 mcfarm       4096 Apr  8 14:24 scr000
lrwxrwxrwx    1 mcfarm         20 Mar 19 13:12 scr001 -> /home3/ouhep1/scr001
lrwxrwxrwx    1 mcfarm         20 Apr 10 10:33 scr002 -> /home2/ouhep2/scr002
lrwxrwxrwx    1 mcfarm         20 Apr 10 10:35 scr003 -> /home2/ouhep3/scr003
lrwxrwxrwx    1 mcfarm         20 Apr  7 16:15 scr004 -> /home2/ouhep4/scr004
lrwxrwxrwx    1 mcfarm         20 Apr  7 16:18 scr005 -> /home2/ouhep5/scr005
lrwxrwxrwx    1 mcfarm         18 Apr  7 16:22 scr009 -> /home2/hope/scr009
```

with each scrXXX directory containing

```
/scratch/users/mcfarm/scr000:
total 44
drwxrwxr-x   10 mcfarm       4096 Apr  8 14:24 .
drwxr-xr-x    3 mcfarm       4096 May 20 11:59 ..
drwxrwxr-x    3 mcfarm       4096 Apr  9 15:15 archive
drwxrwxr-x    3 mcfarm       8192 May 22 20:25 cache_A
drwxrwxr-x    2 mcfarm       4096 May 27 12:37 error_queue
drwxrwxr-x    2 mcfarm       4096 May 28 15:41 exec_queue
drwxrwxr-x    2 mcfarm       4096 May 28 15:49 gath_queue
lrwxrwxrwx    1 mcfarm         23 Mar 25 11:52 localbin -> /home/mcfarm/mcfarm/bin
drwxrwxr-x    2 mcfarm       4096 May 27 12:32 logs
drwxrwxr-x    3 mcfarm       4096 May 28 15:42 run
```

Here, /homeX/ouhepY refers to the local disk /myhomeX on worker node ouhepY. That way, new nodes can easily be added or removed with a simple script (which we wrote), since everything is in one place. One advantage of this global scratch space scheme is ease maintenance and administration.

V. MONITORING

The progress of the MC production at various production sites can be monitored on the UTA job status Web pages [13], and the health and load of various DØ MC production clusters is monitored with the Ganglia Cluster Monitoring Tool [14]. The installation instructions for the Ganglia client setup is described in Appendix D.

VI. USER EXPERIENCE AND UTILIZATION OF THE FARM

After running McFarm for several months on the OUHEP cluster, we can say that it runs reasonably stable and does not require a lot of intervention unless there is some kind of external problem, be it with the cluster because of a power outage or another software or hardware issue, with SAM because of storage or database issues, or with McFarm bugs that are discovered and need to be fixed.

While the initial setup takes a dedicated expert with lots of help from the McFarm authors, during smooth running the farm can easily be operated by a part time student, with expert help required only when encountering problems.

McFarm does not interfere with other batch submissions or interactive jobs, since the batch system handles the interplay between all jobs very well.

When fully loaded with MC requests, the farm operates with an average efficiency of about 95%, and small clusters like OUHEP and LUHEP with about 12 - 16 processors can produce roughly 20 - 30 thousand events per week when given full use of the resources.

With a good internal network, the maximum desktop cluster size that McFarm can handle should not be limited to small clusters, since the McFarm software has the capabilities to distribute the network load by using several of the nodes for administration and data transfer, thus eliminating a possible bottleneck on the job server. At this point, the authors predict a cluster size limit of about 200 processors.

VII. SUMMARY AND OUTLOOK

Setting up a cluster to do DØ MC production takes some time, but is getting easier and more automated. The configuration outlined in this note will allow a non-dedicated desktop cluster to function as a MC farm which can transparently store to and receive data from the central data store at Fermilab. Of particular note here is that the configuration is not bound to Red Hat Linux since it functions fine under Debian Linux. This broadens the possible installation base and provides alternative choices when considering cluster design.

For reference, this is the setup we have installed on the OUHEP and LUHEP clusters:

- Operating System: RedHat 7.2 (OUHEP), Debian 3.0 (LUHEP)
- System Software: NIS, NFS, autofs, OpenSSH

- Batch System: Condor
- Grid Software: Globus
- Fermi Software: UPS/UPD, SAM
- DØ Software: DØRunII releases, mc_runjob
- MC Manager: McFarm
- System Monitoring Software: Ganglia

The Web pages which contain all the relevant information about setup, configuration, and running the MC production software are still in the process of being updated. New versions of McFarm software and installation instructions will be posted regularly [15, 16].

For future versions of MC Farm software, it will be of great importance to be versatile enough to be installable on any type of linux cluster, with any kind of normal batch queuing system installed, especially without root access, and to be able to run it without permanent daemons or outside network access on any of the worker nodes.

This will open up a much greater number of shared clusters for MC production, and move toward a true Grid environment.

---

[1] http://www-hep.uta.edu/d0-sar/d0-sar.html
[2] http://www-hep.nhn.ou.edu/d0/grid/
[3] http://www-hep.nhn.ou.edu/atlas/grid/
[4] http://www.cs.wisc.edu/condor/
[5] http://www.globus.org/
[6] http://www.lsc-group.phys.uwm.edu/vdt/
[7] http://www-d0.fnal.gov/software/cmgt/cmgt.html
[8] http://www-clued0.fnal.gov/runjob/
[9] http://d0db.fnal.gov/sam/
[10] http://www-hep.uta.edu/∼d0race/Linux_D0software_intro.html
[11] http://www-hep.uta.edu/∼d0race/McFarm/McFarm.html
[12] http://hepfm000.uta.edu/
[13] http://hepfm000.uta.edu/job_status/
[14] http://hepfm000.uta.edu/ganglia-webfrontend-2.5.1/
[15] http://www-hep.uta.edu/∼d0race/McFarm/scripts/
[16] http://www-hep.nhn.ou.edu/d0/grid/docs/

Appendix: OUHEP Installation Notes

APPENDIX A: YP (NIS) SERVER

Installation of NIS (YP) server and clients:

First, need to install the required rpms:

ypbind
yp-tools

on all machines, and in addition

ypserv

on ouhep1.
ouhep1 is the NIS server, and all clients get their user info from it.

On the server:

In /var/yp/securenets (if it still exists -- I think newer NIS versions
don't have that anymore), change:

This line gives access to everybody. PLEASE ADJUST!
255.255.254.0   <your_IP_subnet>

In /var/yp/Makefile, change MINUID and MINGID to the minimum UID and GID
you'd like to be propagated by NIS. Here I set it to 300 and 100,
respectively, since we have a user with UID 300, and I wanted to be sure
that the default linux group users, GID=100) gets in.

And comment out the maps you don't want to be built. Here, I have

all: passwd group hosts rpc services netid protocols mail

Then generate NIS database:

export NISDOMAIN=<your_new_yp_domainname>
/usr/lib/yp/ypinit -m

On the client machines:

Remove all local users (which will be in the NIS maps)
from /etc/passwd and /etc/group
(with "userdel <username>), and then add

+::::::

at the end of /etc/passwd and /etc/group,
to allow NIS to add users internally.

On all machines (i.e., including the server):

In /etc/host.conf, add 'nis':

```
order hosts,bind,nis

In /etc/yp.conf, add:

ypserver ouhep1

In /etc/nsswitch.conf, change the following:

------
passwd:     compat
group:      compat
shadow:     compat

passwd_compat:     nis
group_compat:      nis
shadow_compat:     nis

hosts:      files dns nis

bootparams: nis [NOTFOUND=return] files

ethers:     nis [NOTFOUND=return] files
netmasks:   nis [NOTFOUND=return] files
networks:   nis [NOTFOUND=return] files
protocols:  nis [NOTFOUND=return] files
rpc:        nis [NOTFOUND=return] files
services:   nis [NOTFOUND=return] files

netgroup:   nis [NOTFOUND=return] nisplus

publickey:  nis [NOTFOUND=return] nisplus

automount:  files [NOTFOUND=return] nisplus
aliases:    nis [NOTFOUND=return] files nisplus
------

In /etc/sysconfig/network, add:

NISDOMAIN=<your_new_yp_domainname>


Then add services:

/sbin/chkconfig --add ypbind
/sbin/chkconfig ypbind on

On the server, also:

/sbin/chkconfig --add ypserv
/sbin/chkconfig --add yppasswdd
/sbin/chkconfig ypserv on
/sbin/chkconfig yppasswdd on


Then start first the server:

/etc/init.d/ypserv start
```

```
/etc/init.d/yppasswdd start
/etc/init.d/ypbind start

then the clients:

/etc/init.d/ypbind start


If that doesn't work, reboot all machines, that should do it.
```

APPENDIX B: NFS AUTOMOUNTER (AUTOFS)

```
Installation of the automounter (autofs):

In order for this to work, you need to have nfsd (/etc/init.d/nfs)
running and have all disks which you would like to automount
to other systems exported properly in /etc/exports .

rpm -Uvh autofs-*.rpm

In /etc/auto.master, add

/misc   /etc/auto.misc  --timeout=60
/home1  /etc/auto.home1 --timeout 60
/home2  /etc/auto.home2 --timeout 60
/home3  /etc/auto.home3 --timeout 60
/home4  /etc/auto.home4 --timeout 60
/home5  /etc/auto.home5 --timeout 60

In /etc/auto.misc, add

cd              -fstype=iso9660,ro,nosuid,nodev :/dev/cdrom
local                                   ouhep1:/usr/local
home                                    ouhep1:/home
scratch                                 ouhep0:/scratch
raid                                    ouhep0:/raid

This will mount /usr/local/, ..., which is physically located on ouhep1,
and serves the entire cluster, as /misc/local on all other machines,
when they request it. You then need a soft link on all nodes
(except on the server node, where the directory already exists):

ln -s /misc/local   /usr/local
ln -s /misc/home    /home
ln -s /misc/scratch /scratch
ln -s /misc/raid    /raid

In /etc/auto.home[1-5], add

*                                               &:/myhome[1-5]
^                                               ^       ^
|                                               |       |
hostname key                                 repeat    mount point
                                          hostname key    on <hostname>

This assumes that all machines <hostname> have local disks called
/myhome[1-5], and can be seen by all other machines as /home[1-5]/<hostname>/

Then all you have to do is start autofs:

/etc/init.d/autofs start

and it will create all required mount points (/misc, /home[1-5])
for you and will allow you to access /home[1-5]<hostname>/
from any machine, which will automount <hostname>:/myhome[1-5]
for as long as it is being accessed, and then unmount is again
after it has been unused for more than 60 s (the timeout).
```

APPENDIX C: CONDOR

```
# cd /atlas/tar/condor
# tar zxvf condor-6.4.7-linux-x86-glibc22.tar.gz
# cd condor-6.4.7
# ./condor_install

Would you like to do a full installation of Condor? [yes]
Are you planning to setup Condor on multiple machines? [yes]
Will all the machines share files via a file server? [yes]

What are the hostnames of the machines you wish to setup?
(Just type the hostnames, not the fully qualified names.
Put one machine per line.  When you are done, just hit enter.)
ouhep0
ouhep1
ouhep2
ouhep3
ouhep4
ouhep5
ouhep6
ouhep7
ouhep8
hope

Setting up Condor for the following machines:
ouhep0 ouhep1 ouhep2 ouhep3 ouhep4 ouhep5 ouhep6 ouhep7 ouhep8 hope

Have you installed a release directory already? [no]

Where would you like to install the Condor release directory?
[/usr/local/condor]
That directory doesn't exist, should I create it now? [yes]

If something goes wrong with Condor, who should get email about it?
[root@ouhep1.nhn.ou.edu] hs@nhn.ou.edu

What is the full path to a mail program that understands "-s" means
you want to specify a subject? [/bin/mail]

Do all of the machines in your pool from your domain ("nhn.ou.edu")
share a common filesystem? [no] yes

Do all of the users across all the machines in your domain have a unique
UID (in other words, do they all share a common passwd file)? [no] yes

In some cases, even if you have unique UIDs, you might not have all users
listed in the password file on each machine.
Is this the case at your site? [no]

Shall I create links in some other directory? [yes]

Where should I install these files?
[/usr/local/bin]

What is the full hostname of the central manager?
[ouhep1.nhn.ou.edu]
```

```
You have a "condor" user on this machine.  Is the home directory for
this account (/home/condor) shared among all machines in your pool?
[yes]

Do you want to put all the Condor directories for each machine in
subdirectories of /home/condor/hosts? [yes]

Do you want to specify a local partition for file locking? [yes]

Where should I put the lock files? [/var/lock/condor]

Do you want all the machine-specific config files for each host in one
directory? [yes]
What directory should I use? [/usr/local/condor/etc]

What name would you like to use for this pool?  This should be a
short description (20 characters or so) that describes your site.
For example, the name for the UW-Madison Computer Science Condor
Pool is: "UW-Madison CS".  This value is stored in your central
manager's local config file as "COLLECTOR_NAME", if you decide to
change it later.  (This shouldn't include any " marks).
OUHEP

Should I put in a soft link from /home/condor/condor_config to
/usr/local/condor/etc/condor_config [yes]

To start Condor on any machine, just execute:
/usr/local/condor/sbin/condor_master

Since this is your central manager, you should start Condor here first.


------------ End of condor_install script -------------

# /usr/local/condor/sbin/condor_init
# cd /home/condor/
# chown -R condor.condor condor_config hosts/
# rm /home/condor/hosts/ouhep1/condor_config.local

Change parts in /usr/local/condor/etc/condor_config to:

##  Where is the machine-specific local config file for each host?
#LOCAL_CONFIG_FILE = $(LOCAL_DIR)/condor_config.local
LOCAL_CONFIG_FILE = $(RELEASE_DIR)/etc/$(HOSTNAME).local

HOSTALLOW_READ = *.nhn.ou.edu, *.cs.wisc.edu

HOSTALLOW_WRITE = *.nhn.ou.edu

BackgroundLoad = 0.6
HighLoad = 0.8

UWCS_START              = $(CPU_Idle)
UWCS_SUSPEND            = $(CPU_Busy)
UWCS_CONTINUE           = $(CPU_Idle)

PREEMPT_VANILLA = False
```

```
UWCS_PREEMPTION_REQUIREMENTS = ( ($(StateTimer) > (1 * $(HOUR)) && RemoteUserPrio > SubmittorPrio * 1.2

JOB_RENICE_INCREMENT = 19

DEFAULT_RANK = TARGET.KFlops

Also, copy condor script from /usr/local/condor/etc/examples/condor.generic
to /usr/local/bin/condor and change

$default_pool="";           to

$default_pool="default";

and

%configlocation = (
);                          to

%configlocation = (
        "default",      "/usr/local/condor/etc/condor_config",
);

To startup on boot:
# cp -p /usr/local/condor/etc/examples/condor.boot /etc/rc.d/init.d/condor
# ln -s /etc/rc.d/init.d/condor /etc/rc.d/rc3.d/S95condor
# ln -s /etc/rc.d/init.d/condor /etc/rc.d/rc5.d/S95condor
# ln -s /etc/rc.d/init.d/condor /etc/rc.d/rc0.d/K04condor
# ln -s /etc/rc.d/init.d/condor /etc/rc.d/rc6.d/K04condor

Do full install of condor_compile script (on all machines):
# mv /usr/bin/ld /usr/bin/ld.real
# cp -p /usr/local/condor/lib/ld /usr/bin/ld
```

APPENDIX D: GANGLIA

```
Get rpms from

http://ganglia.sourceforge.net/ :
ganglia-monitor-core-gmetad-2.5.3-1.i386.rpm
ganglia-monitor-core-gmond-2.5.3-1.i386.rpm

http://www.rrdtool.com/ :
rrdtool-1.0.38-1.i386.rpm
rrdtool-devel-1.0.38-1.i386.rpm

On ouhep0:

rpm -Uvh ganglia-monitor-core-gmetad-2.5.3-1.i386.rpm ganglia-monitor-core-gmond-2.5.3-1.i386.rpm rrdtoo

On ouhep1-9:

rpm -Uvh ganglia-monitor-core-gmond-2.5.3-1.i386.rpm rrdtool-1.0.38-1.i386.rpm rrdtool-devel-1.0.38-1.i3

Configure /etc/gmetad.conf (on ouhep0):
change the following:

data_source "OUHEP" localhost
trusted_hosts hepfm000.uta.edu

Configure /etc/gmetad.conf (on ouhep0-9):

name  "OUHEP"
num_nodes  10

Then start (or restart):

/etc/init.d/gmetad start   (on ouhep0 only)
/etc/init.d/gmond start    (on ouhep0-9)
```