CHTC Student Handbook

Edition 0.1

Center for High Throughput Computing

May 27, 2016

Contents

1 Student Work 101

1.1 Tasks and Workflow

As a student worker for the CHTC you will be responsible for a large variety of tasks to help keep the CHTC Cluster up and running, but there a few basic things you will find yourself doing most often. Our primary job is to fix individual machines in the cluster that go down. Most often machines can be brought down by kernel panics, networking issues, or hardware issues. To fix a machine, we take a simple approach to diagnose the issue and come up with a solution.

1. Diagnose the problem. You might need to plug into the machine with a console, or SSH into it if it is still online. If a machine kernel panics it will often print a memory stack trace to the console and freeze up. You will need to physically reboot the machine to proceed.
2. Find a solution. Often times rebooting the machine will clear up minor problems such as kernel panics or networking issues.
    - Kernel Panic (crash) - reboot, check server for errors.
    - Networking issue - reboot machine or restart networking, then go to advanced networking troubleshooting if still needed.
    - Hardware issue - diagnose the hardware issue and fix it. (Replace a bad disk, make note of a bad RAID card, etc)
3. Fix the problem and confirm that condor is running jobs again.
    - his might mean rebooting the machine, rebuilding the machine, replacing a bad disk and rebuilding, or something else entirely. Fix it if possible. If not - make note of the issue in the ticket and await further instructions.

Some other tasks you might be assigned include, but are not limited to: archiving condor releases to DVD, building new execute machines, working with Dell/Cisco Tech Support to get replacement parts, moving servers, a nd other random tasks assigned by full time CHTC staff.

1.2 RT Ticket System

We use the Request Tracker (RT) Ticket System for managing work at the CHTC. crt.cs.wisc.edu is the web address to access our Request Tracker.This will be your home for managing work tickets. The basic flow is: You are assigned a ticket, click on the ticket name to open it. By default, your 10 highest priority, or newest, tickets will be listed on your home page. You can view all of your tickets by clicking the "10 Highest Priority Tickets I Own" link (counter-intuitive, I know).

Once you have selected a ticket, you can read all correspondence related to that particular issue. Under the "actions" button (top right) the most common things you will do are "reply" and "resolve." Whenever you make changes to a machine, make sure to log what you did in the ticket. When the machine is fixed or the task is completed, you can resolve the ticket.

1.3 Monitoring Nodes

You can monitor the cluster on monitor0.chtc.wisc.edu. This has links to Icinga, our host monitoring application, Grafana, and Ganglia. All of these are extremely powerful tools that can be used to monitor the status of machines in the cluster and diagnose problems with individual nodes.

1.4 How to Use This Guide

- Any line or section beginning with "$" is a command line operation. Enter it directly as it appears, without the preceding "$".
- Any bracketed item (eg. [System Name]) needs to be replaced with the corresponding entity. (eg. change [System Name] $\rightarrow$ e1000.chtc.wisc.edu).

- Following the step-by-step walkthroughs to complete most basic tasks and use this as a reference manual.
- The appendix includes a complete list of node locations, useful for location a machine for physical changes (hard reboot, HDD swap, etc).

2 Rebuilding a Node

Rebuilding nodes is one of the primary responsibilities of student CHTC employees. Here's a step-by-step walkthrough for you.

1. Enable netboot in cobbler.
   (a) Via Command Line:
      i. SSH into wid-service-1.chtc.wisc.edu
      ii. `$ sudo cobbler system edit --name=[server name] --netboot-enabled=True`
      iii. Check that netboot is enabled → `$ sudo cobbler system report --name=[server name]`
      iv. Sync Cobbler → `$ sudo cobbler sync`
   (b) Via Web Interface
      i. Open Cobbler in your web browser
      ii. Click on "Systems" under the Configuration tab on the left side.
      iii. Find the node you want to rebuild and click on it's name in the list to open the node configuration tool.
      iv. Alternatively, navigate directly to wid-service-1.chtc.wisc.edu/cobbler_web/system/edit/[node name]
      v. Click on the "general" drop down button to reveal more options
      vi. Check the "enable netboot" option
      vii. Hit "save" on the bottom of the page
      viii. When you are redirected to the cobbler system list page, hit "Sync" Under the Actions tab on the left side.
      ix. Once you get a popup notification on the top-right of the screen, the sync has complete. This may take a few seconds.
   (c) Make sure you have the correct profile enabled in Cobbler as well. If you are doing a standard rebuild you probably won't have to change it, but make sure if it's a multi-disk execute node that it is set to the correct SL66 Exec profile.
2. Reboot the machine and netboot it.
   (a) Most machines are set to netboot by default, meaning if you reboot them they will search for a netboot entry and if they find it, they will netboot automatically. If a machine is not netbooting automatically, you may need to press a button on the keyboard (Often F12) when it POSTS in order to force it to netboot. If you have tried these and it still won't boot from the network, go into the BIOS and change the boot order to set netboot as boot priority #1.
   (b) If done correctly, it should launch the Scientific Linux installer. Once you see this is happening, move on to the next step.
   (c) To diagnose problems with the boot process, when the loading bar appears, you can press Alt+d to display a verbose boot screen.
3. Run Puppet.
   (a) We are going to need to run puppet once the machine is rebuilt in order to configure it.
   (b) While the machine is rebuilding, SSH into wid-service-1.chtc.wisc.edu
   (c) Run this command: `$ sudo puppetca -c [node name]` on wid-service-1.
   (d) If you do this before the machine finishes rebuilding, it may run puppet automatically when it rebuilds. To see if it is doing this: When the machine is booting Scientific Linux, hit an arrow key ← or →, or ALT+D on the console keyboard to view the verbose boot log. If the log is paused on "`Starting: anamon... [OK]` " for a while, that mean's it's running puppet. Good job! Once it finishes booting now, you should be able to log in with your username.
   (e) If the machine does not run puppet automatically, connect a console to the machine and log in as a root user (ask Admin for root login)

(f) ON THE TARGET NODE: run `$ sudo rm -rvf /var/lib/pup/ssl` (After clearing the puppet files from wid-service-1 in the previous steps). DO NOT RUN THIS ON wid-service-1.

(g) After clearing the puppet files from wid-service-1 in the previous steps, run `$ sudo rm -rvf /var/lib/puppet/ssl` on the target node. DO NOT RUN THIS ON wid-service-1.

(h) Then run `$ sudo puppetd -tv --configtimeout=1000`

(i) Note: Puppet will not run if networking is broken (try to restart networking or reboot the machine if this is the case) or the system clock is broken. To set the clock run: `$ rdate -s ntp.doit.wisc.edu`

4. Log in and confirm condor is running.

(a) Once puppet finishes running, you should be able to log in with your own username

(b) Confirm condor is running: `$ condor_status $HOSTNAME`

(c) You should also see Condor running with `$ ps aux | grep condor`

3 Building a New Node

TODO - Flesh out this section.

Building a new node is very similiar to rebuilding a current machine, with the added caveat of configuring the node in Cobbler and DHCP.

1. First, we need to create a new entry for the node in Cobbler. You should be provided a hostname and IP address to use for the new machine. You can create a new Cobbler entry via the Web interface or CLI.
    - Via Web Interface
        - Go to cobbler web.
        - Create a new system. The Easiest way to do this is to copy an existing entry and change the relevent information.
        - Open up the new machine entry you just created, change the hostname, the IP address, and MAC address, and make sure that the profile and image are set to the correct one.
        - Enable netboot, save it, and sync cobbler once it's configured correctly.
    - Via CLI
        - CLI instructions... TODO
2. Next, we need to create a DHCP entry for this machine.
    - DHCP is managed by different machines depending on the node location.
        - B240: cobbler-b240
        - 2360/CSL & WID Data Center: wid-service-1
        - 3370: host-6
    - SSH into the correct server and become root. Then open up dhcpd.conf in VIM.
    - Copy an exec node entry and edit it to create a new entry with the correct IP and MAC address of the new machine. Save the file.
    - run `$ service dhcpd restart` - if an error comes up, you screwed up the .conf file.

4 Decommissioning a Node

TODO

5 Creating Tickets

If you don't have any tickets assigned to you in your RT Queue that you can work on, then you can go ahead and create your own tickets for work that needs to be done.

We'll use a simple example to outline how to create tickets.

Let's say your queue is empty - you can log into Icinga (Or another monitoring tool that we use) to look for any nodes that might be offline or experiencing other issues.

1. Head to monitor0.chtc.wisc.edu in your browser.
2. Click on the link to Icinga, and log in (ask Admin for login info). This will take you to the Icinga dashboard. From here, you can get a basic overview of server status.
3. Look for machines listed under the "Host Problems" section.
    - Let's say we see a node, e1000.chtc.wisc.edu, listed under that section with the status message "Critical - Host Unreachable".
    - Click on that host to bring up information about it. From here, you can get a better overview of what's wrong with the machine. In this case, the machine is "Down" - offline for whatever reason. It might have crashed, it might be powered off, we don't know yet. All we know is that Icinga can't connect to it.
4. Great, now that we see a server is down, we need to check RT to see if a ticket already exists for this issue.
    - Log in to RT (crt.cs.wisc.edu) in a new tab.
    - Start a new search - select "New Search" under the "Tickets" tab at the top of the screen.
    - Enter the server name (e1000) in the "Subject" box, and make sure the relational selector is set to "Matches".
    - Click on "Add these terms and search".
5. At this point, you will either see (a) ticket(s) matching the server, or none at all. We want only want to have 1 ticket for managing each server, so it is important to not open new tickets when one already exists.
6. Let's say that you see several tickets that come up. First, check the ticket status.
    - If all tickets are "Resolved", then you are fine to re-open one. You will always re-open the most recent, or most-related ticket.
    - If there is an OPEN ticket related to the issue, then someone is already working on this issue and you can not continue creating a ticket.
    - If there are NO TICKETS matching, then you can create a new ticket.
7. If you want to re-open a closed ticket, first you will have to assign the ticket to yourself. Under the "People" Tab, choose your name from the "Owner" drop-down, and hit "Save Changes".
8. Then to re-open the ticket, hit "Re-Open" under the "Actions" tab.
    - You may now work on the ticket like you would any other.
9. If there are no tickets matching, you will need to create a new ticket.
    - At the top-right of the page next to the search bar is a ticket creation button. From the drop-down, select "htcondor-inf" - This will open the page to create a new ticket in the htcondor-inf queue.
    - In the right-most box labeled "Basics", put yourself in the "Owner" field and "CHTC" in the inf_subgroup field.

- In the left-most box, enter an appropriate subject (”e1000 down”), and a short-description if desired.
- Click ”Create” on the bottom of the page.
- Assuming you made yourself the ticket owner, you will now see this ticket in your queue!

10. LASTLY, if you re-opened or created a new ticket, you need to acknowledge that you did so on Icinga so that other poeple won’t create a duplicate ticket.

    - Return to Icinga, and on the dashboard for that particular node, click on ”Acknowledge” under the ”Problem Handling” section.
    - enter a short message, (eg ”Created a ticket - [your name]”).
    - Hit ”Acknowledge Problem” - And you’re done!

Congratulations - you now know how to find problems on Icinga and create a ticket in order to fix it!

6 Checking for Errors

6.1 Monitoring with Icinga

6.1.1 Icinga Basics

6.1.2 Web Interface

6.1.3 Icinga CLI

6.2 Error Logs

6.3 Hardware Errors

7 Advanced Troubleshooting

TODO

8 Advanced Topics

8.1 Networking

8.1.1 Networking Basics

8.1.2 CHTC Network Structure

8.1.3 Troubleshooting Networking Issues

A Node Locations

A.1 B240

Rack	A2	A4	C1	C2	C4	E2	E5	E6
Node	c071	mussel	e036	c031	atlas13	e061	atlas80	atlas64
	c061	atlas18	e027	c032	atlas14	e060	atlas81	atlas65
	c073	atlas19	e039	c033	atlas28	e059	atlas82	atlas66
	c074	atlas20	atlas17	c011	atlas15	e058	atlas83	atlas67
	c041	e047	atlas11	c035	starfish	e002	atlas84	atlas68
	c076	e035	atlas12	c037	e013	e025	atlas85	atlas69
	c077	e034	atlas16	c038	e012	e030	atlas86	atlas70
	c078	[???]		c039	e011	e057	atlas87	atlas71
	c064	e006		c040	spalding11	e022	atlas88	atlas72
	c080	cobbler		c021	spalding10	e040	atlas89	atlas73
	c062	e010		c022	spalding09	e023	atlas90	atlas74
	c053	e045		c023	spalding08	e018	atlas91	atlas75
	c054	e044		c024	spalding07	e016	atlas92	atlas76
	c055	[???]		c025	spalding06	e026	atlas93	atlas77
	c046	e003		c026	spalding05	e028	atlas95	atlas78
	c057			c027	c081	e041	atlas96	atlas79
	c058			c028	c082	e038	atlas97	atlas80
	c059			c029	c084	e042	atlas98	atlas42
	c060			c030	c085	e046	atlas99	atlas62
	e049			c012	c086	e056		
	e050			c013	c088	e052		
	e051			c049	c089	e043		
	e054			c015	c090	db1		
	e055			c016		db2		
	e062			c017		db3		
	e063			c069		db4		
	e064			c020		db5		
	e065			c001		db6		
	e031			c002		db7		
				c004				
				c070				
				c065				
				c007				
				c008				
				c009				
				c010				

Rack	G1	G2	G5	G6
Node	satellite	e024	[???]	atlas45
	e029	e053	atlas29	atlas46
	stress1	e009	atlas30	atlas47
	stress2	e008	atlas31	atlas48
	stress3	e007	atlas32	atlas49
	stress4	e005	atlas33	atlas50
	host-5	e089	atlas34	atlas51
		e090	atlas35	atlas52
		e088	atlas36	atlas53
		e087	atlas37	atlas54
		e086	atlas38	atlas55
		e085	atlas39	atlas56
		e084	atlas40	atlas57
		e083	atlas41	atlas58
		e082	atlas43	atlas59
		e081		atlas60
		e080		atlas61
		e079		atlas63
		e078		
		e077		
		e076		
		e075		
		e074		
		e073		
		e072		
		e071		
		e072		
		e069		
		e068		
		e067		

A.2 2360/CSL

Rack	GR 6	GR 0	GR 4	GR 3	10	GR 7
Node	atlas07	e299	e339	e379	e419	e459
	atlas04	e298	e338	e378	e418	e458
	atlas06	e297	e337	e377	e417	e457
	atlas05	e296	e336	e376	e416	e456
	atlas03	e295	e335	e375	e415	e455
	atlas02	e294	e334	e374	e414	e454
	mem2	e293	e333	e373	e413	e453
	mem1	e292	e332	e372	e412	e452
	gpu4	e291	e331	e371	e411	e451
	host-22	e280	e330	e370	e410	e450
	host-20	e289	e329	e369	e409	e449
	host-18	e288	e328	e368	e408	e448
	host-16	e287	e327	e367	e407	e447
	host-14	e286	e326	e366	e406	e446
	host-12	e285	e325	e365	e405	e445
	host-10	e284	e324	e364	e404	e444
		e283	e323	e363	e403	e443
		e282	e322	e362	e402	e442
		e281	e321	e361	e401	e441
		e280	e320	e360	e400	e440
		e279	e319	e359	e399	e439
		e278	e318	e358	e398	e438
		e277	e317	e357	e397	e437
		e276	e316	e356	e396	e436
		e275	e315	e355	e395	e435
		e274	e314	e354	e394	e434
		e273	e313	e353	e393	e433
		e272	e312	e352	e392	e432
		e271	e311	e351	e391	e431
		e270	e310	e350	e390	e430
		e269	e309	e349	e389	e429
		e268	e308	e348	e388	e428
		e267	e307	e347	e387	e427
		e266	e306	e346	e386	e426
		e265	e305	e345	e385	e425
		e264	e304	e344	e384	e424
		e263	e303	e343	e383	e423
		e262	e302	e342	e382	e422
		e261	e301	e341	e381	e421
		e260	e300	e340	e380	e420

A.3 3370

Rack	1	4	5	6	7	8	9	GR 5
Node	e198	[???]	osghost	host-6	e222	submit-4	e116	spalding-4
	e197	swamp05	itb-data1	pagesubmit	e220	host-23	e117	spalding-1
	e196	swamp04	itb-data2	atlas21	e219	host-21	nmi-0067	spalding-2
	e195	swamp03	itb-data3	atlas22	e215	host-19	e115	spalding-3
	e194	swamp02	itb-data4	atlas27	e214	host-17	gpu-1	
	e193	swamp01	itb-data5	atlas26	e119	host-15	[???]	
	e192		itb-data6	atlas25	e212	host-13	atlas10	
	e191		itb-host1	atlas24	e211	host-11	atlas09	
	e190		itb-host2	atlas23	e210	e238	e111	
	e189		itb-host3		e209	e237	matlab-build-5	
	e188		vdt-bastion		e208	e236	e113	
	e187		vdt-centos5-test		e207	e235	e112	
	e186		vdt-debian6-test		e206	e234	e246	
	e185		vdt-debian7-test		e205	e233	e245	
	e184				e204	e232	e244	
	e183				e203	e231	e243	
	e182				e202	e230	e242	
	e181				e201	e229	e241	
	e180				e200	e228	e240	
	e179				e199	e227	e239	
	e178				host-24	e226	e259	
	e177					e225	e258	
	e176					e224	e256	
	e175					e223	e255	
	e174						e254	
	e173						e253	
	e172						e252	
	e171						e251	
	e170						e250	
	e169						e249	
	e168						e248	
	e167						e247	
	e166							
	e165							
	e164							
	e163							
	e162							
	e161							
	e160							
	e159							
	e158							
	e157							
	e156							
	e155							
	e154							
	e153							
	e152							
	e151							

A.4 WID Data Center

Rack	A14	A12	B1	B5
Node	e093	deepdivesubmit	wid-003	e110
	e094	host-3	[???]	e109
	e095	[???]	e019	e108
	e121	[???]	e017	e107
	e122	[???]	e020	e106
	e123	host-7	e033	e105
	e124	submit-5	e021	e104
	e125	[???]	e014	e103
	e126	quickstep	e015	e102
	e127	wid-service-1	matlab-build-1	e101
	e128	e092	e001	e100
	e129	osg-ss-se	e091	e099
	e130	[batlab]	[???]	e098
	e131	host-1	[???]	e097
	e132	host-2	[???]	e096
	e133			
	e134			
	e135			
	e136			
	e137			
	e138			
	e139			
	e140			
	e141			
	e142			
	e143			
	e144			
	e145			
	e146			
	e147			
	e148			
	e149			
	e150			

B List of Resources

TODO - A useful list of external resources.

- http://pages.cs.wisc.edu/ van-lyse/ - Neil's CS Page contains tons of useful CHTC links.
- monitor0.chtc.wisc.edu - Our webpage for server monitoring. Contains links to Icinga, Grafana, Ganglia, and more.

C Glossary

TODO - Glossary of terms commonly used in this manual and in daily work.

- Cobbler - Cobbler is a frontend tool for configuring Linux network installs with PXE boot. Learn more: http://cobbler.github.io/
- Icinga - An open-source system monitoring tool with a web interface. Our Icinga server is at monitor0.chtc.wisc.edu. Learn more at: https://www.icinga.org/
- Puppet - Puppet is a tool for managing automatic configuration of systems. Our puppetmaster machine is wid-service-1. Learn more: https://puppet.com/
- RT (Request Tracker) - Ticket-tracking software. Access RT at: https://crt.cs.wisc.edu/rt/