High Performance Computing at Shahid Chamran University of Ahvaz



Vahid M. Saffarzadeh

Clustering of Computers

Advantages:

- High performance
- Large capacity
- High availability
- Incremental growth

Applications:

- Scientific computing
- Making movies
- Commercial servers (web/database/etc)

SCU HPC Cluster Overview

It's a starting Point:

- 5 computing nodes;
- 160 CPU cores, 32 per node;
- Designed for long-running computationally intensive jobs
- Optimized for batch jobs rather than interactive jobs
- Designed to run jobs that employ parallel processing or many single-processor jobs
- Runs Linux (CentOS Linux), Windows users should change some habits!



Why Linux Cluster?

- Linux runs on a wide range of hardware
- Linux is exceptionally stable
- Linux source code is freely distributed.
- Linux is relatively virus free.
- Having a wide variety of tools and applications for free.
- Good environment for developing cluster infrastructure



How to Connect

• Due to network limitations users currently can only connect at university campus and via SSH connection (no desktop).



How to Use the HPC

Different types of applications can be run on the cluster:

- **Progrmans** written in C, C++, Fortran using supported compilers from GNU.
 - Serial programs
 - **Parallel programs** using shared parallelism (OpenMP) or distributed parallelism (MPI)
- **Pre-installed applications** including: MATLAB-like applications and R
- Molecular dynamics/Quantum: Gromacs
- Fluid dynamics: Fluent, Abaqus

A Fluent Run Example

🚯 Al	pplications	Places Sys	stem 🕹 (2					Sun No	ov 9,	6:56 PN
					t	oin - File Brows	er	-	• ×		
		<u>File</u>	dit <u>V</u> iew <u>G</u>	o <u>B</u> ookmark	2		Terminal			_	o x
	Computer	de Ba	ck 🖌 🃦	Forward 🗸	<u>File E</u> dit <u>V</u> iew	<u>Search</u> Term	inal <u>H</u> elp				
			home	vahidms	/home/vahidms/ /home/vahidms/ <.3.7.3 -f flu	apps/Fluent. apps/Fluent. ent (fluent	Ansys.6.3/Fluent.Inc/f Ansys.6.3/Fluent.Inc/f " -r6.3.26 -path/home	luent6.3.26/bin/flue luent6.3.26/cortex/ /vahidms/apps/Fluent	ent -: Lnamd(:.Ansy	:6.3. 54/cc ys.6	.26 <u>~</u> orte .3/F
		fluent@	⊉cluster.scu.	ac.ir [0] Fluer	t Inc	_ = ×					
	Residuals 	1e+01 = 1 1e+00 = 1 1e-01 = 1 1e-02 = 1 1e-03 = 1 1e-04 = 0		15 20	25 20 25 4		fluent	-		×	10
L	Scaled Residuals			Iteratio	FLUENT 6 3 (3)	Nov 09, 2014				b	
A	Ansys.Fluent.	6.3 "fluent" s	elected (2.9 h								
e t	oin - File Brov	vser	fluent		🕞 Terminal		[fluent]	fluent@cluster.scu.a	ıc)		

Logging in to the Cluster

- Where you can login from:
 - Campus computer (wired or via Wireless)
- What you need to login:
 - Use ssh utility such as Putty, cygwin (Windows) or Terminal (Mac). Telnet, RCP or Remote Desktop will not work.
 - An approved cluster account
 - Enter the UserID and the Single Sign-On password

Cluster Logging Example (for Windows users)(1)

🕵 PuTTY Configuration		×					
Category:							
E Session	Basic options for your PuTTY session						
Logging ⊡ Terminal Keyboard	Specify the destination you want to con Host Name (or IP address) 172,16,48,70	Port 22					
Features	Connection type: Raw Telnet Rlogin SSH Serial						
 Appearance Behaviour Translation Selection Colours Connection Data Proxy Telnet Rlogin SSH Serial 	Load, save or delete a stored session Saved Sessions	Load Save Delete					
About	Only or exit:	n clean exit					

Cluster Logging Example (for Windows users)(2)

• Once the SSH Connection is open, you should see a terminal prompt asking for your username:

login as:

Enter your primary account, username.

• Next, enter your password. Note that you will NOT see your any characters typed when typing your password.

Using keyboard-interactive authentication. Password:

Cluster Logging Example (for Windows users)(3)

• You are now logged into the server with SSH. You see output like this:

username@cluster:~\$

• Now, you are in the head node of the cluster and you can begin typing commands at the prompt.

Cluster Logging example (for Linux & Mac users)

- Open a terminal or xterm. Then type this command
 - ssh username@172.16.48.70
- After entering your password you will be logged into the head node of the cluster.

Copy files to and from cluster using winscp

🎦 Documents - test@192.	168.1.100 - WinSCP									
Local Mark Files Com	mands Session Options Remot	e Help								
🖶 🚼 📮 Synchronize	📰 🧬 💽 🧶 🔛 Que	eue 🔹 Transfer Settings De	fault 👻 🧭 🗸							
📮 test@192.168.1.100 📹	New Session									
Mv documents	- - 🚰 🔽 (🔶 - 🔿 -) 🔂 🛱				🔒 test 🔹 n 🖉 🔽	7 +	🖅 🗿 Find Files	B _		
I CP Unload CP DV Edit	V / B. Dromotion									
Chilleenhachid De surreente										
	<i></i>				/state/partition1/nome/test	<i>a</i> . <i>a</i>	2 : 1 -	<u> </u>		
Name	Size Type	Changed			Name	Size Changed	Rights	Owner		
🗯	File folder	3/19/2015 2:11:40 PM 2/28/2015 9:41:57 PM			5	5/15/2015 10:40:3	PM DVXP-XF-X	root		
Python Scripts	File folder	2/28/2015 9:41:57 PM			test.m	1 KB 5/16/2015 8:34:31	AM pw-pw-r	test		
The KMPlayer	File folder	2/12/2015 5:51:20 PM								
Visual Studio 2008	File folder	2/11/2015 7:22:11 PM								
퉬 Visual Studio 2013	File folder	3/19/2015 3:39:31 PM								
0 B of 0 B in 0 of 5				4 hidden	0 B of 143 B in 0 of 2					13 hidden
										SFTP-3 ↓ 0:01:01

Copy files to and from cluster using Filezilla



Linux Basic on the Cluster (1)

Using the Linux shell:

• Linux shell is basically where you enter the commands to manipulate files or to run a code or to execute a script

Files and directories

- The root directory is /, and the directories underneath this root are /home, /usr, /tmp, /etc
- Your home directories is located at /home/Username
- You can create subdirectories underneath your home directory (use mkdir command). Each job usually runs on its own directory
- You can create file, copy file (with cp command), move or rename files (with mv command), or remove files (with rm)

Linux Basic on the Cluster (2)

Basic Linux shell commands related to files and directories

(type man *command* for help on each commands below)

cat : to list the content of a text file	locate : to list all the files in the local machine with the given name
cd : to change the current directory	ls : to list the files and directories in the directory
cp : to copy files	mv : to move the files or rename them
df : to list out the disk usage of the filesystems	pwd : to list the current directory
less : to list the content of a text file by page	rm : to delete files
In : to create a symbolic link to another file or a directory	which: to indicate the location of the executables/binaries

Modules and Environment Module command

This sets up the environment necessary to run your applications (binary, libraries, shortcuts)

• To see all applications you can load in your environmnet type:

module avail

• To load a module into your current environment type:

module load modulename

module load octave3.08

Using the nano Editor

- nano is a text editor.
- We will use nano as an example on how to write a short Fortran code

```
cd ~/examplePrograms
nano squares.f
PROGRAM SQUARE
do i = 1,10
write(*, *) i*i
enddo
END
```

• Press Ctrl+x and Enter to save and exit the editor

An Example of Loging and Using MATLAB (1)

- After you logged in to your account on the cluster type:
 - module avail

Welcome to Shahid Chamran University High Performance Computing Center

An Example of Loging and Using MATLAB (2)

•You now see a list of available modules containing MATLAB module, you select your desired version (eg. 2012a) and type:

module load matlab2012a

#######################################									
[test@cluster ~]\$ module avail									
dot	module-git	module-info	modules	null	ob				
openmpi-x86_64									
matlab2012 [test@cluster ~]\$ module load matlab2012									

An Example of Loging and Using MATLAB (3)

• Now when you type "matlab" command, MATLAB application will be open in **nodisplay mode**.

----- /share/apps/modulefiles ------

matlab2012 [test@cluster ~]\$ module load matlab2012 [test@cluster ~]\$ matlab Warning: No display specified. You will not be able to display graphics on the screen.

< M A T L A B (R) >
Copyright 1984-2012 The MathWorks, Inc.
R2012a (7.14.0.739) 64-bit (glnxa64)
February 9, 2012

To get started, type one of these: helpwin, helpdesk, or demo. For product information, visit www.mathworks.com. MATLAB Parallel Computing An Introduction - Step1

Is your code correct and have your code ready and running in ordinary MATLAB?

MATLAB Parallel Computing An Introduction - Step2

Prepare your ordinary-running code to become a parallel-running code to run using multiple processors. You need two main instructions in your code for this purpose:

MATLAB Parallel Computing An Introduction - Step2(matlabpool)

matlabpool

It tells MATLAB that you want to use multiple processors in your run

At the beginning of the code, type:

matlabpool('open',#);

where *#* is the number of cores you want to use. This instruction **opens the parallel computing toolbox.**

MATLAB Parallel Computing An Introduction - Step2(matlabpool)

At the end of the code, type: matlabpool('close');

This will close the parallel computing, however, the license will still be in use until you close your MATLAB session.

MATLAB Parallel Computing An Introduction - Step2 (parfor)

• parfor

parfor is the parallel computing version of the traditional "for" command for a loop. Use it in place of "for", just as simple as that:

```
parfor i = 1 : 100
```

(your code)

end

The loop variable (i) can only be a vector of consecutive integers. Also, the code inside this loop is the portion of your code that will be executed in parallel.

Check "MATLAB help" for particular conditions for both "matlabpool" and "parfor".

MATLAB Parallel Computing An Introduction – Step3 – Some Tips(1)

- When using "parfor":
 - all the executions of the loop need to be independent from each other.
 - "parfor" loop doesn't even execute consecutively (i = 1, 2, 3, 4, ...), but quite randomly (i = 3, 4, 59, 37, ...) decided by MATLAB.
 - The "parfor" loop can only be done for one level, i.e., you cannot have a "parfor" loop inside another "parfor" loop.

MATLAB Parallel Computing An Introduction – Step3 – Some Tips(2)

• To have "for" loops inside the "parfor" loop, you need to pre-allocate all the variables that grow inside the "for" loops.

• The best way to get your results out is to print to file and save data inside the "parfor" loop, this is also a good way to take results out of the "parfor" loop.

A Simple MATLAB Code

```
matlabpool ('open',2);
x = zeros(5,10);
parfor i = 1:5
   y = zeros(1,10);
   for j = 1:10
     y(j) = i;
   end
   y
   x(i,:) = y;
variable
end
Χ
matlabpool close;
```

% Call to open the distributed processing % Initialize the main variable % Parallel loop % Initialize the secondary variable % Inner loop

% Display the inner variable % Get values from loop into the main

% Display main variable % Close the distributed computing

A Simple MATLAB Code - Result

Sta	rting	matlab	opool u	ising	the 'p1'	prof	ile	. coni	nected	to 2	labs.
y =											
-	2	2	2	2	2	2	2	2	2	2	
y =											
	1	1	1	1	1	1	1	1	1	1	
y =											
	4	4	4	4	4	4	4	4	4	(4)	
y =											
	3	3	3	3	3	3	3	3	3	3	
v =											
,	5	5	5	5	5	5	5	5	5	5	
x =											
	1 2	1 2	1 2	1 2	1 2	1 2	1 2	1 2	1 2	1 2	
	3	3	3	3	3	3	3	3	3	3	
	4 5	4 5	4 5	4 5	4 5	4 5	4 5	4 5	4 5	4 5	
Sen >>	ding	a stop	signal	to a	all the 1	labs .	sto	pped.			