大工数学科学学院高性能服务器

快速入门手册

Lenovo 1800 Quick Start

Contributers: JJCAO 公司网址: <u>http://sites.google.com/site/hpcdut/</u> 文档版本: 1.0

1. Preparation

For conducting c++ parallel computation on the cluster, we have to master two kinds of login modes: web and shell.

"命令行执行" is a very bad tool. Forget it please. Only very few commands can be run in it successfully. Hence, for mastering the cluster, we have to incur to a shell tool, such as SSH Secure Shell.



1.1. Login from the web



1.2. Login from the shell

We recommend SSH Secure Shell for login. The host name is the web address of the cluster.

道 - default - SSH Se			
Eile Edit View y	<u>M</u> indow <u>H</u> elp		
B 4 Q B /	i Bee A	🔊 🍋 🍓 🍖 k ?	
👔 Quick Connect	📄 Profiles		
SSH Secure Shell Copyright (c) 200 This copy of SSH This version	3.2.9 (Build 283) D-2003 SSH Commun Secure Shell is a nect to Remote Host	ications Security Corp non-commercial version	- http://www.ssh.com/ h.
	<u>H</u> ost Name:	202. 118. 75. 60	Connect
T.	🗳 <u>U</u> ser Name:	test	Cancel
	Port	22	
	Authentication	<pre>@rofile Settings> ▼</pre>	

After login, we are on the *console* node actually. It is evidently if we login using a shell tool. For example in [test@console], *test* is user name, and *console* is the node name.



We can not compile any code on the console node, and it has to be done on a computation node, such as c0101, c0102, ... c0116. We can use the command *rsh* to switch from one node to another node, referring to step 4 of section 2. While *rsh* always failed to run on the web, we have to recur to the shell. Furthermore, shell offers more functions and agility.

2. C++ Parallel Computation

- 1. Setting environment (This step is needed only after the user is just created or some new software is installed)
 - a) [test@console]\$ cp /export/lenovo-hpc/bashrc.example .bashrc
- 2. Making a new dir for the computation, first.
 - a) [test@console ~]\$ mkdir example1
- 3. Uploading necessary files into the dir example1
 - a) Use SSH Secure File Transfer

2:202.118.75.60 -	default - SSH Secure File T	ransfer			
∬ Eile Edit ⊻iew C)peration <u>W</u> indow <u>H</u> elp				
🖬 🎩 🎉 🖻 🖻	1 🗿 🎦 🛸 🗍 J 🗘		abc 010 016 🖸	@ ∖ ?	
🛛 🛃 Quick Connect 🛛	📄 Profiles				
🔁 🗗 🗢 🖻	🛉 🗙 🔤 (Downloads)	▼ Add	🔁 🖾 🛍 🖆	× rt/home/tes	t/example1 🗾 🛛 Add
Local Name	🗸 🦳 Size Type	Modified	Remote Name	 Size 	e Type Modified
examplec.c	717 C Sou	rce 2009-04-0{	🖾 cpi	32,679	・ 文件 2009-04-
🔤 examplec.c.bak	737 BAK 3	之件 2009-04-0:	🕒 cpi.c	3,719	C Source 2009-04
<	III.		•	III .	
C Source File	Source Directory	Destination Directo	ry Size Sta	tus	Sneed Time
	Source Birdetory	Destination Directo	, <u>, , , , , , , , , , , , , , , , , , </u>		opeca nine

Connected to 202.118.75.60 - /export/home/test/example1 | SSH2 - aes128-cbc - hmac-md5 - r 2 items (36.4 KB) 🗌 🏹 📝

b) or On the web

1 作业与资源管理	☞ 文件管理	
 ■ 「中山与政保官理 ● ■ 系统配置 ● 启停控制 	目录列表[当前目 全选 录:/export/home/test/example1] 清空	删除目录
◎ 作业管理	上一级目录	
 ● 作业提交 ● 文件管理 	文件列 全选 清 编辑文件 新建文件 删除文件 删除文件	下载文件
◎ 命令行执行	上载文件	
🕀 🖿 作业记帐管理	E:\WUTemp\cpi.c	上载

- 4. Compiling (it must be done on a computational node, such as c0101, ... c0116)
 - a) [test@console ~]\$ rsh c0102
 - b) [test@c0102 ~]\$ mpicc -o example1/cpi example1/cpi.c

If you meet any error, there will be some output on the screen.

- 5. Setting a job script (Our cluster uses HCA card of Qlogic, so the mpi used is <u>QLogic MPI</u>.)
 - a) [test@console ~]\$ cp /export/lenovo-hpc/ljrstest example1/ljrstest
 - b) The content of the script is:

echo this script pid is \$\$ echo Jobing directory is \$LJRS_O_JOBDIR echo \$LJRS_O_JOBDIR = \$LJRS_O_JOBDIR cd \$LJRS_O_JOBDIR echo Runing on host `hostname` echo Time is `date` echo Directory is `pwd` echo This jobs runs on the following processors: echo \\$LJRS_NODEFILE=\$LJRS_NODEFILE echo `cat \$LJRS_NODEFILE` NPROCS=`wc -I < \$LJRS_NODEFILE` echo This job has allocated \$NPROCS nodes mpirun -m \$LJRS_NODEFILE -np \$NPROCS /export/home/YOU PROGRAM echo `date`

- c) Edit the script. Replace "YOU PROGRAM" with absolute path of your executable file. (In this example, replace YOU PROGRAM with /export/home/test/example1/cpi)
- d) After editing, the new script is (Keeping the file in a Linux format by editing it on Linux
 - or using software such as EditPlus on Windows): echo this script pid is \$\$ echo Jobing directory is \$LJRS_O_JOBDIR echo \$LJRS_O_JOBDIR = \$LJRS_O_JOBDIR cd \$LJRS_O_JOBDIR echo Runing on host `hostname` echo Time is `date` echo Time is `date` echo Directory is `pwd` echo This jobs runs on the following processors: echo \\$LJRS_NODEFILE=\$LJRS_NODEFILE echo `cat \$LJRS_NODEFILE` NPROCS=`wc -I < \$LJRS_NODEFILE` echo This job has allocated \$NPROCS nodes mpirun -m \$LJRS_NODEFILE -np \$NPROCS /export/home/test/example1/cpi echo `date`
- 6. Committing the job on the web or on the console
 - a) On the console
 - i. [test@console example1]\$ qsub -l nodes=16:ppn=8 ljrstest



- nodes: number of nodes
- ppn: number of cores of CPUs (The CPU of our cluster is quad-core CPU)
- There will be two standard output files: ljrstest.o82 (normal output) and ljrstest.e82 (error info)
- b) On the web
 - i. 作业与资源管理/作业提交

	☞ 作业提交				
				# #	48.77
◎ 启停控制				<u>#1</u>	I IEX
○ 作业管理	基本选项				
 作业提交 文件管理 	脚本名 /export/home/test/examplel/ljrstest 选择文件 新達作业提交脚本				
 ○ 命令行执行 ● ■ 作业记帐管理 	作业名	目的结点池 test_pool ↓	结点个数	每个结点CPU个数	
* SZ - I	输出选项				
1.66	输出是否合并	并 ○ 合并到标准输出 ○ 合并到错误输出 ◎ 不合并			
- s top at	使用缺省输出文件	一标准输出[jobname].o[seq]		标准错误输出[jobname].e[seq]	
	指定标准输出	文件名		结点名	
and the state of the	指定错误输出	文件名		结点名	
1.60	邮件选项				
- + 2 f 45	邮件地址		邮件发送时间: 🗐 作到	业开始执行时 📄 作业终止	时 🦳 作业结束时
	资源配置				
21:500	选择资源	cput 👻	资源值		增加
- "1" -	选定的系统资源列表		全选	清空	删除
	结点分配策略配置				
~ * 24	诸选择分配策略	◎缺省 ◎综合 ◎ α	pu 🔘 mem		
				重要	提 提交

7. Check the result

In this example, result is in ljrstest.o82.

3. Job management

More info is in 联想深腾 1800 作业调度与资源管理系统用户手册.doc.

3.1. Examine standard outputs

If the job submitted was done, you can find two outputs files in the dictionary where the job script is, namely *.e??, and *.o??. 'e' stands for error info, 'o' normal output info, and '??' job id. An example for normal output is:

[nistest@console ~]\$ cat test.o1 this script pid is 24271 Jobing directory is /export/home/nistest = /export/home/nistest Runing on host c0101 Time is Tue Jun 17 14:01:39 CST 2008 Directory is /export/home/nistest This jobs runs on the following processors: \$LJRS_NODEFILE=/usr/local/spool/ljrs/aux//11.console c0101 c0101 c0101 c0102 c0102 c0102 c0102 This job has allocated 8 nodes g0101 g0101 g0101 g0102 g0102 g0102 g0102 running /export/home/nistest/cpi on 8 LINUX ch_p4 processors Created /export/home/nistest/PI24280 pi is approximately 3.1416009869231249, Error is 0.0000083333333318 wall clock time = 0.003906 Tue Jun 17 14:01:39 CST 2008

We can compare this output with the c++ source file generated it, /export/lenovo-hpc/cpi.c.

3.2. If your job is always queued

4. FAQ

- Why I got a compilation error (catastrophic error: could not open source file "iostream.h") when including iostream.h? Use <iostream> instead.
- Why I got a compilation error (catastrophic error: could not open source file "iostream") when including iostream? Change the postfix from c to cpp.