HPC Servers (p5) Seminar: Exercise Instructions

November 2, 2005
Hirofumi Amano (**amano@cc.kyushu-u.ac.jp**)
Computing and Communications Center, Kyushu University

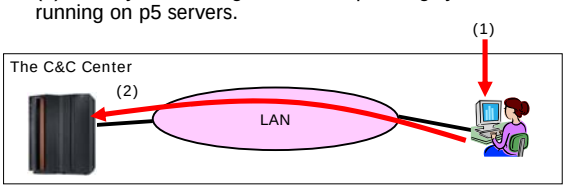
You can download this presentation file from:
http://isabelle.cc.kyushu-u.ac.jp/~amano/ccc/p5/exercise.pdf
The sample file package is also available from:
http://isabelle.cc.kyushu-u.ac.jp/~amano/ccc/p5/sample20051102.tar

Exercise (1): Interactive Sessions

- Accessing **kyu-cc**
- Preparing sample files
- Compiling and executing a sequential program
- Automatic parallelization
- OpenMP: compilation and execution
- MPI: compilation and execution

Accessing **kyu-cc**

- In this seminar room:
  - (1) First, you must log in the Windows XP operating system.
  - (2) Then, you must log in the AIX operating system running on p5 servers.



- The ID-password pairs will be given at the seminar.

(1) Logging in Windows XP

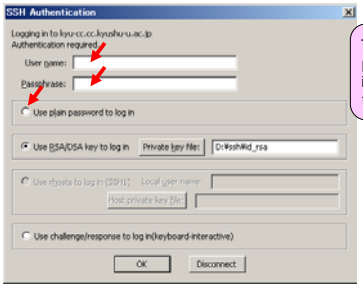
- I hope you know what to do :-)

(2) Logging in AIX

- On Windows desktop:
  - double-click “TTSSH” (or TeraTerm Pro).
- In “Tera Term New Connection” window:
  - Enter “**kyu-cc.cc.kyushu-u.ac.jp**” at “Host”.
  - Check “SSH” in “Service”.
  - Then, click “OK”.
- Even if “SECURITY WARNING” window pops up:
  - Check “Add this machine …” , and click “Continue”.
  - Disregard errors, and click “OK”.

(2) Logging in AIX (continued)

- In “SSH Authentication” window:
  - Type your ID and password.
  - Check “Use plain password to log in”.



The ID-password pair for this login is DIFFERENT from that of Windows!

Changing Your Password

- You will be requested to change your password.

```
Your password has expired; Choose a new one.
You must change your password now.
Changing password for "s70080a"
s70080a's Old password:
```

- Enter the old (initial) password once, and enter your new password twice.
- The session will be terminated after changing the password.
  - You must login to AIX again with your new password.
  - Do not forget your new password.
- Try the next and all the later sessions with the new password.

7

Preparing Sample Files

- Copy the sample file package (**cp**) and unpack it (**tar**)

```
kyu-cc% cp /tmp/sample20051102.tar .
kyu-cc% tar xvf sample20051102.tar
kyu-cc%
```

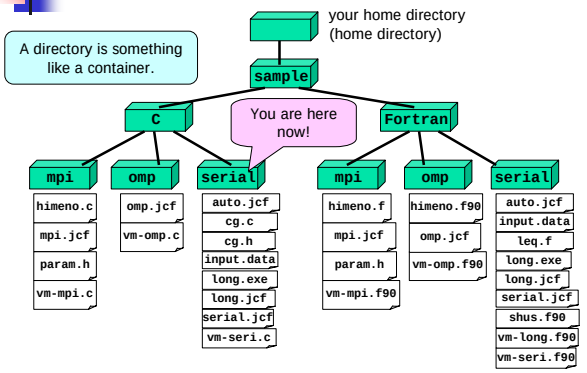
- Move down to the first exercise directory

```
kyu-cc% cd sample
kyu-cc% cd C
kyu-cc% cd serial
kyu-cc%
```

"**cd**" (change directory) is the command to move along the directory tree (hierarchy).

8

Directory Tree (Hierarchy) of Sample Files



9

Prompt from a Shell

- Shell: a command language interpreter
  - Displays a "prompt";
  - Accepts your command and executes it;
  - Displays the next prompt.

```
kyu-cc% cp /tmp/sample20051102.tar .
kyu-cc% tar xvf sample20051102.tar
kyu-cc%
```

prompt

10

Compiling and Executing a Sequential Program

- Compilation
  - Generate an executable code in a file "**vm-seri**".

```
kyu-cc% cc -o vm-seri vm-seri.c
```

- Browse the list of files in the current directory to check the result.
  - Is there a file named "**vm-seri**"?

"**ls**" is a command to display the list of files.

```
kyu-cc% ls -l
```

- Execute the file "**vm-seri**" in the current directory.

```
kyu-cc% ./vm-seri
```

11

Measuring the Execution Time

- Use "**timex**" command.

```
kyu-cc% timex ./vm-seri
(... output from the program ...)
real 12.38        <- Elapsed time (seconds)
user 12.30        <- The sum of these two values is
sys  0.00            the actual CPU time.
kyu-cc%
```

12

Optimization Levels

Optional

- Compare the elapsed times for various optimization levels:

uppercase lowercase Enter key symbol ↵ will be omitted from here.

```
kyu-cc% cc -O -o vm-seri vm-seri.c
kyu-cc% timex ./vm-seri
kyu-cc% cc -O3 -o vm-seri vm-seri.c
kyu-cc% timex ./vm-seri
kyu-cc% cc -O3 -qarch=pwr5 -qtune=pwr5 -o vm-seri vm-seri.c
```


Automatic Parallelization (1)

- Compilation
  - Generate an executable code in a file "**vm-auto**".

```
kyu-cc% cc_r -qsmp=auto -o vm-auto vm-seri.c
```

- Set the number of threads to 2.

```
kyu-cc% setenv OMP_NUM_THREADS 2
```

- Execute the file "**vm-auto**" .

```
kyu-cc% ./vm-auto
```


Automatic Parallelization (2)

Optional

- Compare the elapsed times for various numbers of threads:

```
kyu-cc% setenv OMP_NUM_THREADS 2
kyu-cc% timex ./vm-auto
kyu-cc% setenv OMP_NUM_THREADS 3
kyu-cc% timex ./vm-auto
kyu-cc% setenv OMP_NUM_THREADS 4
kyu-cc% timex ./vm-auto
kyu-cc% setenv OMP_NUM_THREADS 8
kyu-cc% timex ./vm-auto
```


Fortran Programs

Optional

- Move to the Fortran-serial directory and verify the current working directory.

```
kyu-cc% cd
kyu-cc% cd sample
kyu-cc% cd Fortran
kyu-cc% cd serial
kyu-cc% pwd
```

Print Working Directory

- Compilation

```
kyu-cc% f90 -o vm-seri vm-seri.f90
```

The similar set of files are prepared also in Fortran directory. If you have time, you can try the same exercises up to this point in Fortran, too.

OpenMP: Compilation

- Move to the next exercise directory and verify the current working directory.

```
kyu-cc% cd
kyu-cc% cd sample
kyu-cc% cd C
kyu-cc% cd omp
kyu-cc% pwd
```

Print Working Directory

- Compilation

```
kyu-cc% cc_r -qsmp=omp -o vm-omp vm-omp.c
```


OpenMP: Execution (1)

- Set the number of threads to 2.

```
kyu-cc% setenv OMP_NUM_THREADS 2
```

- Execute the file "**vm-omp**" .

```
kyu-cc% ./vm-omp
```

OpenMP: Execution (2)

Optional

- Compare the elapsed times for various numbers of threads:

```
kyu-cc% setenv OMP_NUM_THREADS 2
kyu-cc% timex ./vm-omp
kyu-cc% setenv OMP_NUM_THREADS 3
kyu-cc% timex ./vm-omp
kyu-cc% setenv OMP_NUM_THREADS 4
kyu-cc% timex ./vm-omp
kyu-cc% setenv OMP_NUM_THREADS 8
kyu-cc% timex ./vm-omp
```

19

MPI: Compilation

- Move to the next exercise directory and verify the current working directory

```
kyu-cc% cd
kyu-cc% cd sample
kyu-cc% cd C
kyu-cc% cd mpi
kyu-cc% pwd
```

- Compilation

```
kyu-cc% mpcc –o vm-mpi vm-mpi.c
```

20

MPI: Execution

- Execute the file "**vm-mpi**" .

```
kyu-cc% ./vm-mpi –nprocs 4
```

An anomaly has been reported about **time**/**timex** commands. **time**/**timex** commands do not display an accurate CPU time when they are used with MPI programs. There is no fix or work-around available to this problem.
However, the CPU time information for a batch job will be notified in the automatic e-mail returned by the batch processing system.

The similar set of files are prepared also in Fortran directory. If you have time, you can try the same exercises up to this point in Fortran, too.

21

Exercise (2): Batch Jobs

- Job Command Files (JCFs)
- Job Submission
- Job Status Monitoring
- Job Canceling

22

Job Command Files (JCFs)

- Move to the next exercise directory and verify the current working directory.

```
kyu-cc% cd
kyu-cc% cd sample
kyu-cc% cd C
kyu-cc% cd serial
kyu-cc% pwd
```

Print Working Directory

- Browse a JCF.

```
kyu-cc% cat serial.jcf
```

"**cat**" is a command to display the contents of a file(s).

23

Job Submission

- Create the executable code.

```
kyu-cc% cc –o vm-seri vm-seri.c
```

- Verify that the executable file specified in the JCF really exists.

```
kyu-cc% ls –l
```

- Submit you job.

```
kyu-cc% llsubmit serial.jcf
```

24

Job Status Monitoring and Job Canceling

- Submit a longer job.

```
kyu-cc% llsubmit long.jcf
```

- Display the list of your job(s).
  - Find the job ID.

```
kyu-cc% qps
```

- Cancel your job.

```
kyu-cc% llcancel job_ID
```

25

Parallel Jobs, Fortran Jobs

Optional

- JCFs for auto-parallel, OpenMP and MPI jobs are also prepared in "**serial**", "**omp**" and "**mpi**" directories, respectively.
- JCFs for Fortran jobs are also available.

26