



Technische
Universität
Braunschweig



Parallel Computing

Exercise 1

Andres Rodriguez, 23rd April 2015

Course Organization

✓ Exercise supervisor:

Andres Rodriguez

a.rodriquez-escobar@tu-braunschweig.de

✓ Assignments

To be handed in groups of 2 Students.

Groups can not be changed along the semester.

Submission via email – Please use the next format in the subject field:

Assignment -> PC1 - HW## - Group##

Questions -> PC1 – QUESTION - Group##

Login into the Cluster

Requirements:

✓ VPN connection with the university network

- VPN Client (Anyconnect Secure Mobility Client)
<https://www.tu-braunschweig.de/it/downloads/software>
- Installation Guide
https://www.tu-braunschweig.de/Medien-DB/it/vpn_einrichten_und_nutzen_anyconnect30.pdf

✓ ssh-Client

- Unix – Reachable through Terminal
- Windows – Command Prompt + Cygwin (My Personal suggestion)
<https://www.cygwin.com> - Download
<https://www.youtube.com/watch?v=uTeH7vm8JZU> – Installation Guide
- Windows – putty
<http://www.putty.org>

Connection:

ssh yxxxxxx@akb.hlr.rz.tu-bs.de

Using Linux Command Interface

Our Repository

<https://github.com/PC1-SS15-TUBS/>

✓ Creating directories

```
mkdir <new_directory_Name>
```

✓ List of files in directory

```
ls
```

✓ Moving along directories

```
cd <directory_name>    (Go into a directory)
cd ..                  (Step back one directory)
```

Using Linux Command Interface

✓ Exchanging files Cluster2PC / PC2Cluster

`scp -r ./path/to/origin/file yxxxxxxx@akb.hlr.rz.tu-bs.de:./path/to/destination/file` (PC2Cluster)

`scp -r yxxxxxxx@akb.hlr.rz.tu-bs.de:./path/to/origin/file ./path/to/destination/file` (Cluster2PC)

This is done in a terminal not logged in to the cluster!!

Some examples here:

http://www.hypexr.org/linux_scp_help.php

✓ Creating/Opening file for editing in Cluster

`emacs <file_to_edit/create>`

(Create a file with the given name if it do not exist)

Using the Cluster

| Category | Command | Argument 1 | Argument 2 | Usage |
|----------------------|---------|-----------------|----------------|------------------------------------------------------|
| Modules management | module | Avail | - | List available modules |
| | | load | <module name > | Load a module |
| | | list | - | List all loaded modules |
| | | unload | <module name> | Unload a module |
| Job Queue management | qsub | <job file name> | - | Submit a job file for execution |
| | qstat | - | - | List the current queue status |
| | qdel | Id | - | Delete the job in queue identified with the given ID |

Required modules:

✓ gcc-4.4

✓ parastation

Job File

Job file extension:

<job file name>.pbs

Job file structure:

```
#!/bin/sh
```

```
#PBS -N <taskName>
```

```
#PBS -o <taskName>.out
```

```
#PBS -e <taskName>.err
```

```
#PBS -l nodes=<#>:ppn=<#>
```

```
#PBS -l walltime=<hh>:<mm>:<ss>
```

```
mpiexec -np <#> ./compiledFile.out
```

| | | |
|-----------------|---|------------------------------------|
| <u>taskName</u> | = | Name for the job to be queued |
| <u>nodes</u> | = | Requested multiprocess machines. |
| <u>ppn</u> | = | Requested process. |
| <u>walltime</u> | = | Processe self killing time. |
| <u>np</u> | = | Actual number of processes to use. |

Compiling your code

Regular code without MPI:

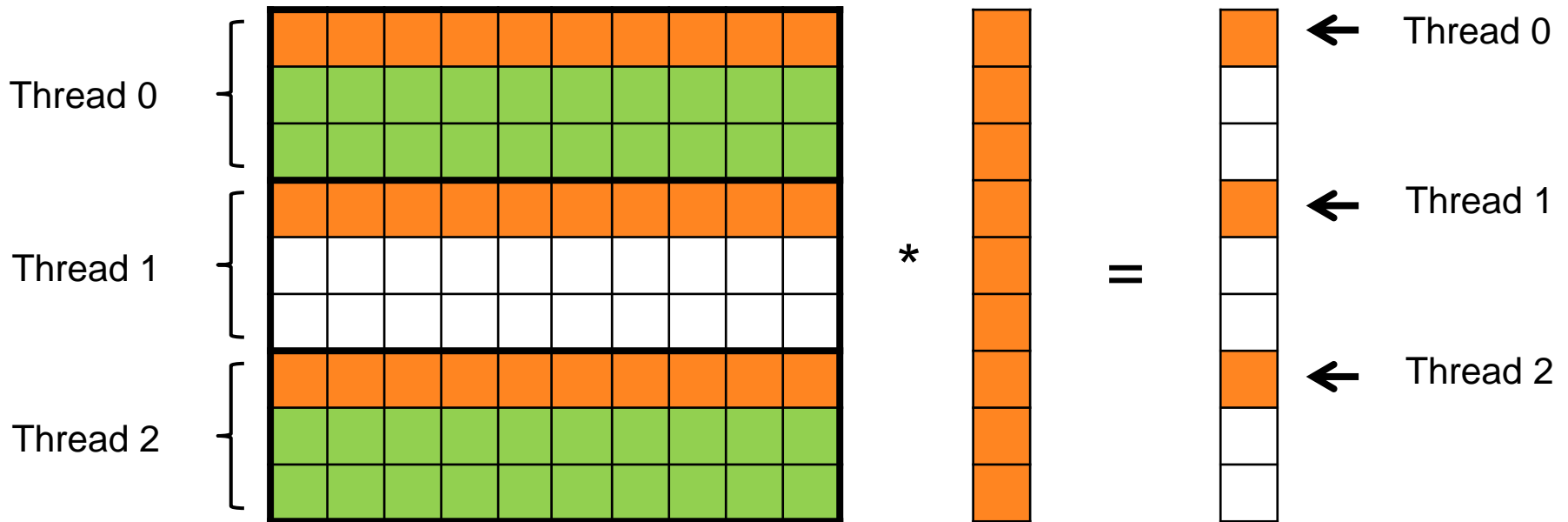
C -> `gcc <fileToCompile>.c -o <outputName> [-I ./path/to/include/directory] [-lpthread]`

C++ -> `g++ <fileToCompile>.c -o <outputName> [-I ./path/to/include/directory] [-lpthread]`

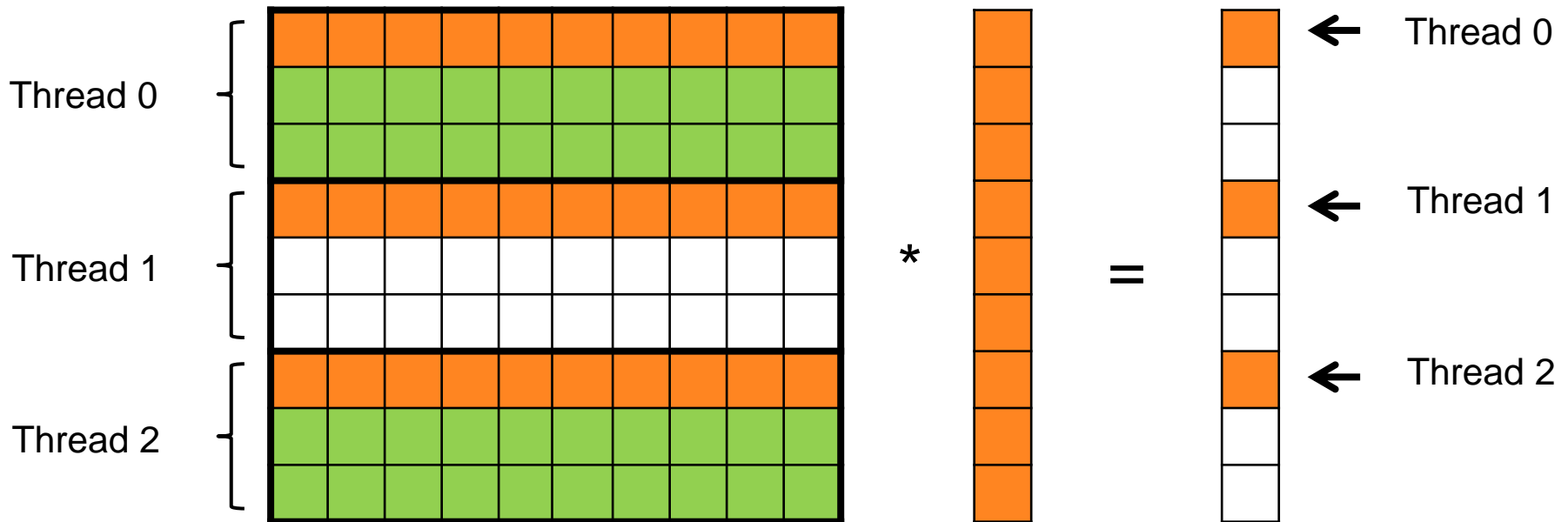
Code with MPI instructions & commands:

C/C++ + MPI `mpicc <fileToCompile>.c -o <outputName> [-I ./path/to/include/directory] [-lpthread]`

Matrix-Vector Multiplication



Matrix-Vector Multiplication



Why is a Mutex no needed in this problem?

How to store a matrix in memory

