Day 1: Get started !

Dr. Tuan-Tu Tran
trant@uni-mainz.de

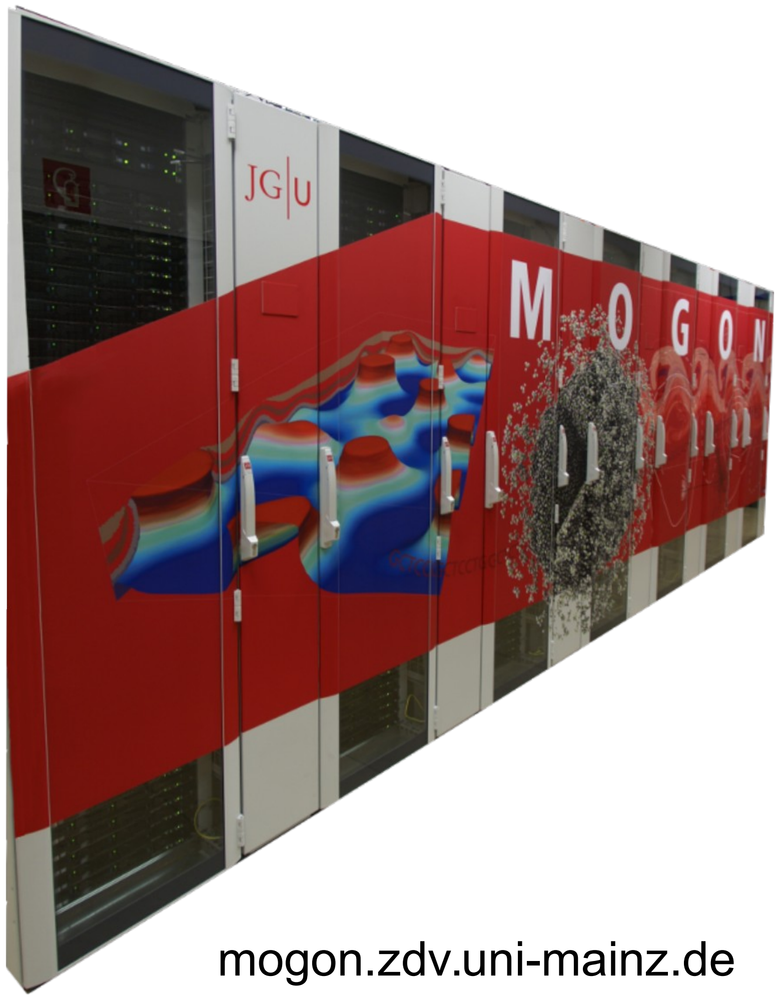
Cuda Practical Winter Term 2013/2014

Part 1

Introduction to the working environment

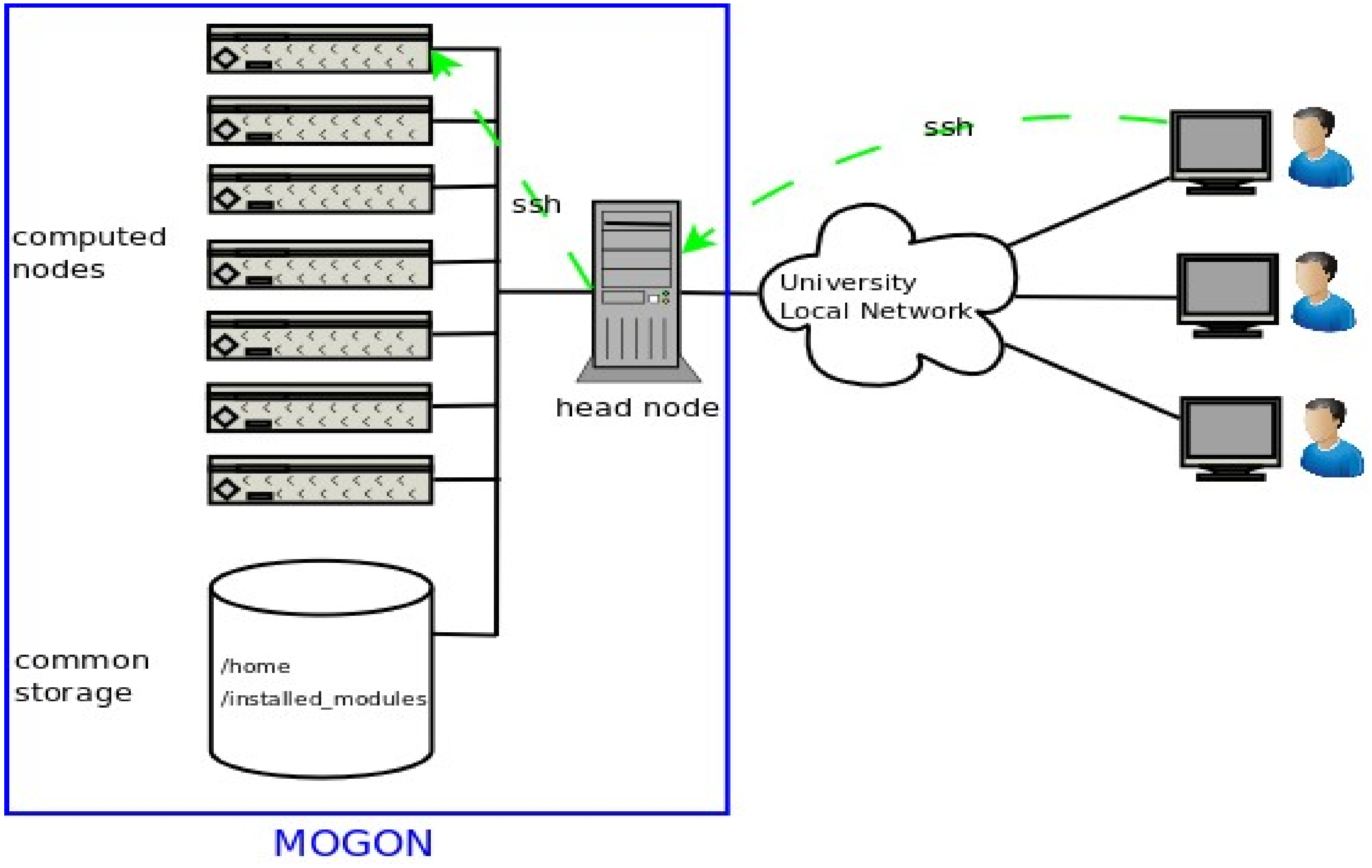
The Mogon Cluster with GPU nodes

JOHANNES GUTENBERG UNIVERSITÄT MAINZ



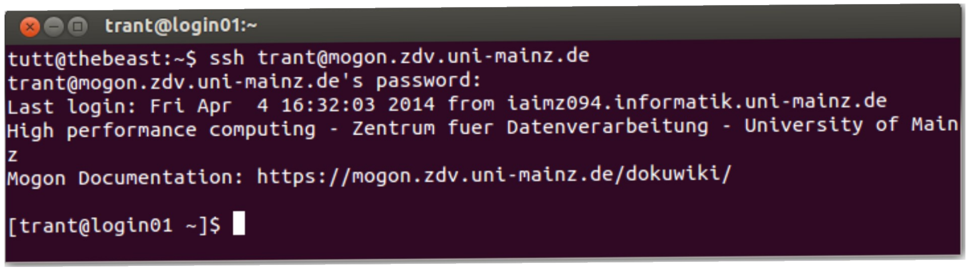
mogon.zdv.uni-mainz.de

Access to the cluster



Access to the cluster by ssh

- Linux/Macs : terminal

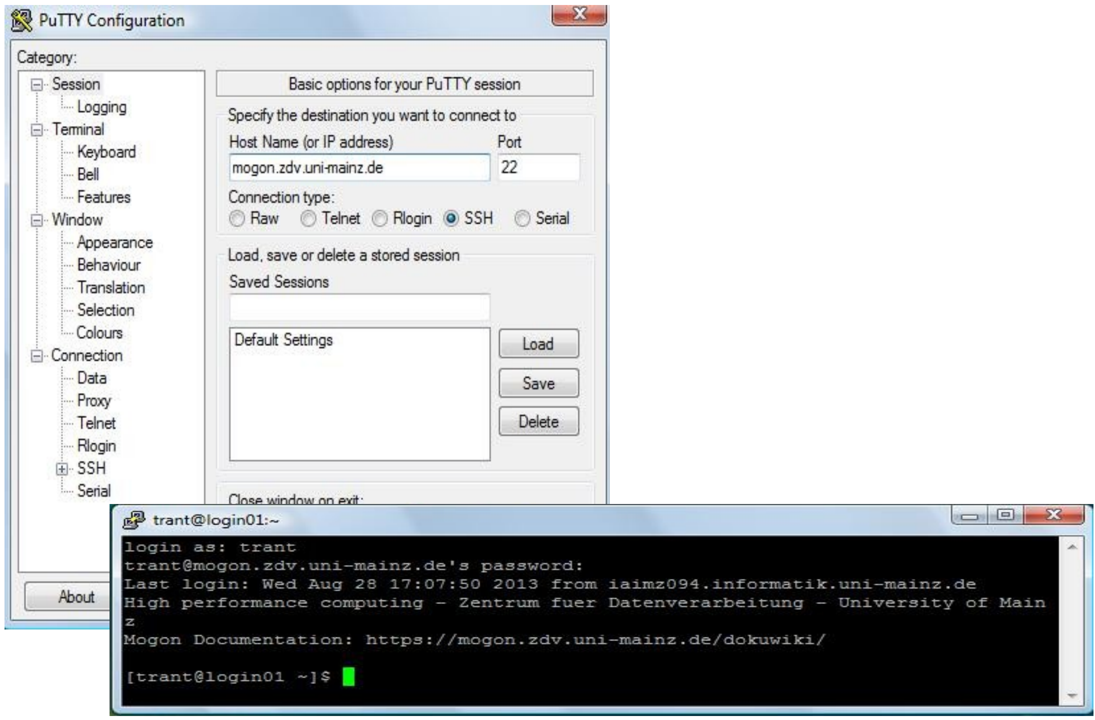


```
trant@login01:~
tutt@thebeast:~$ ssh trant@mogon.zdv.uni-mainz.de
trant@mogon.zdv.uni-mainz.de's password:
Last login: Fri Apr  4 16:32:03 2014 from iaimz094.informatik.uni-mainz.de
High performance computing - Zentrum fuer Datenverarbeitung - University of Main
z
Mogon Documentation: https://mogon.zdv.uni-mainz.de/dokuwiki/

[trant@login01 ~]$
```

- Windows : putty

(http://www.putty.org/)



Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): mogon.zdv.uni-mainz.de   Port: 22

Connection type: Raw Telnet Rlogin [✓ SSH] Serial

Load, save or delete a stored session

Saved Sessions

Default Settings

```
trant@login01:~
login as: trant
trant@mogon.zdv.uni-mainz.de's password:
Last login: Wed Aug 28 17:07:50 2013 from iaimz094.informatik.uni-mainz.de
High performance computing - Zentrum fuer Datenverarbeitung - University of Main
z
Mogon Documentation: https://mogon.zdv.uni-mainz.de/dokuwiki/

[trant@login01 ~]$
```

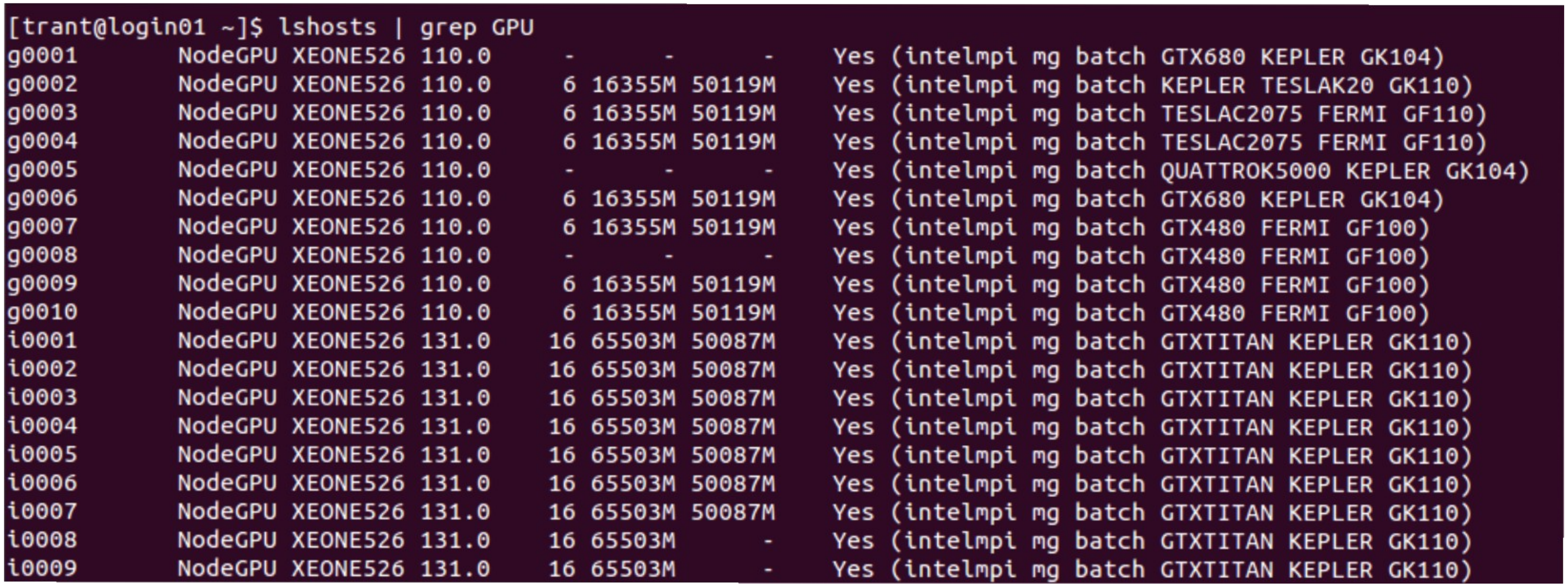
List of GPU nodes

Display all computing nodes:

`$lshosts`

Display all computing nodes which have GPUs:

`$lshosts | grep GPU`



```
[trant@login01 ~]$ lshosts | grep GPU
g0001      NodeGPU XEONE526 110.0    -       -       -    Yes (intelmpi mg batch GTX680 KEPLER GK104)
g0002      NodeGPU XEONE526 110.0    6 16355M 50119M    Yes (intelmpi mg batch KEPLER TESLAK20 GK110)
g0003      NodeGPU XEONE526 110.0    6 16355M 50119M    Yes (intelmpi mg batch TESLAC2075 FERMI GF110)
g0004      NodeGPU XEONE526 110.0    6 16355M 50119M    Yes (intelmpi mg batch TESLAC2075 FERMI GF110)
g0005      NodeGPU XEONE526 110.0    -       -       -    Yes (intelmpi mg batch QUATTROK5000 KEPLER GK104)
g0006      NodeGPU XEONE526 110.0    6 16355M 50119M    Yes (intelmpi mg batch GTX680 KEPLER GK104)
g0007      NodeGPU XEONE526 110.0    6 16355M 50119M    Yes (intelmpi mg batch GTX480 FERMI GF100)
g0008      NodeGPU XEONE526 110.0    -       -       -    Yes (intelmpi mg batch GTX480 FERMI GF100)
g0009      NodeGPU XEONE526 110.0    6 16355M 50119M    Yes (intelmpi mg batch GTX480 FERMI GF100)
g0010      NodeGPU XEONE526 110.0    6 16355M 50119M    Yes (intelmpi mg batch GTX480 FERMI GF100)
i0001      NodeGPU XEONE526 131.0   16 65503M 50087M    Yes (intelmpi mg batch GTXTITAN KEPLER GK110)
i0002      NodeGPU XEONE526 131.0   16 65503M 50087M    Yes (intelmpi mg batch GTXTITAN KEPLER GK110)
i0003      NodeGPU XEONE526 131.0   16 65503M 50087M    Yes (intelmpi mg batch GTXTITAN KEPLER GK110)
i0004      NodeGPU XEONE526 131.0   16 65503M 50087M    Yes (intelmpi mg batch GTXTITAN KEPLER GK110)
i0005      NodeGPU XEONE526 131.0   16 65503M 50087M    Yes (intelmpi mg batch GTXTITAN KEPLER GK110)
i0006      NodeGPU XEONE526 131.0   16 65503M 50087M    Yes (intelmpi mg batch GTXTITAN KEPLER GK110)
i0007      NodeGPU XEONE526 131.0   16 65503M 50087M    Yes (intelmpi mg batch GTXTITAN KEPLER GK110)
i0008      NodeGPU XEONE526 131.0   16 65503M       -    Yes (intelmpi mg batch GTXTITAN KEPLER GK110)
i0009      NodeGPU XEONE526 131.0   16 65503M       -    Yes (intelmpi mg batch GTXTITAN KEPLER GK110)
```

Using GPU nodes

- Access to GPU nodes

```
$ssh i0001
```

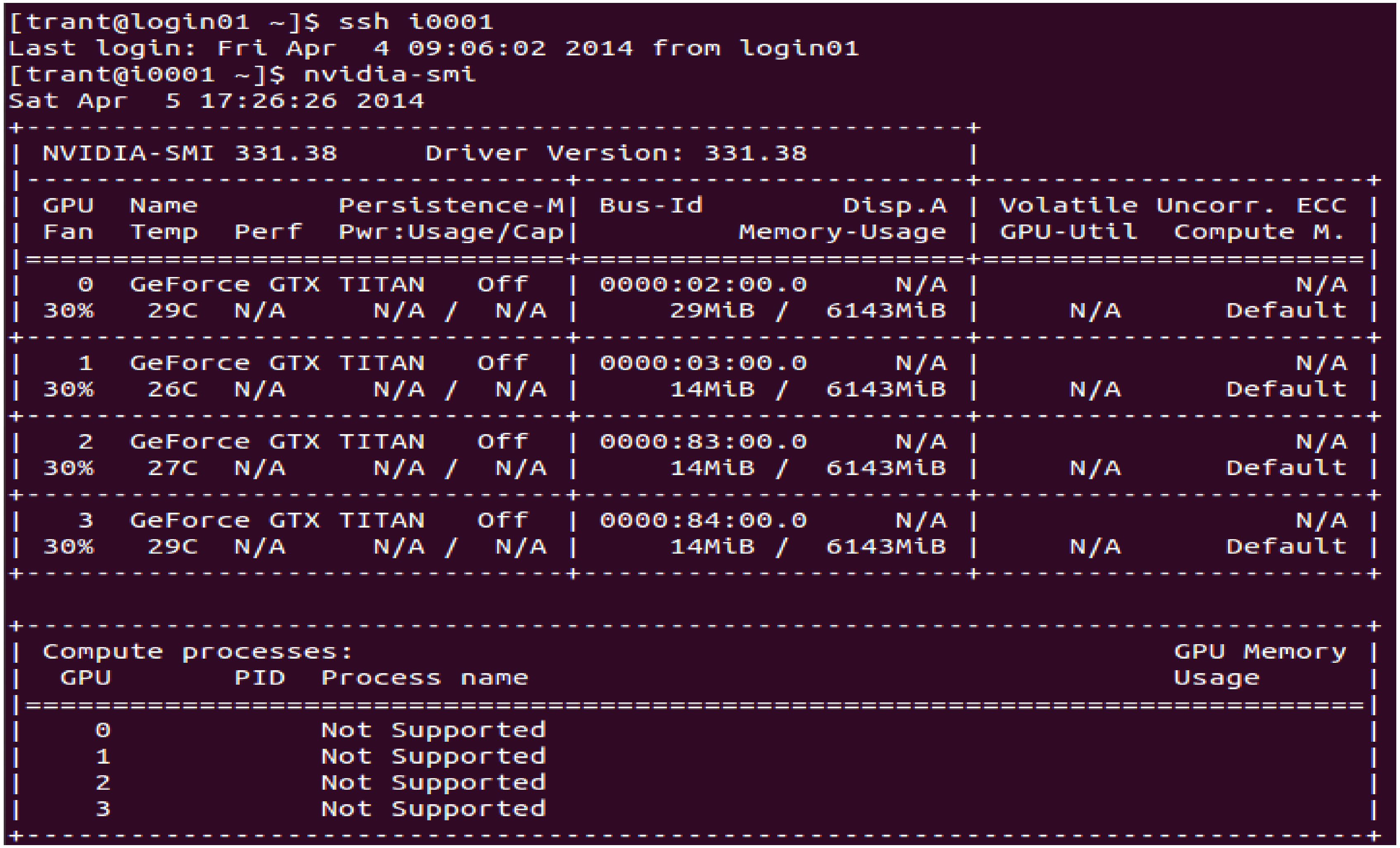
- Load the CUDA module

```
$module add cuda/gcc-4.6.2/5.0.35
```

- Test

```
$nvcc -version
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2012 NVIDIA Corporation
Built on Fri_Sep_21_17:28:58_PDT_2012
Cuda compilation tools, release 5.0, V0.2.1221
```

Display GPU Profiles



```
[trant@login01 ~]$ ssh i0001
Last login: Fri Apr  4 09:06:02 2014 from login01
[trant@i0001 ~]$ nvidia-smi
Sat Apr  5 17:26:26 2014
+------------------------------------------------------+
| NVIDIA-SMI 331.38     Driver Version: 331.38         |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  GeForce GTX TITAN   Off  | 0000:02:00.0     N/A |                  N/A |
| 30%   29C  N/A     N/A /  N/A |     29MiB /  6143MiB |     N/A      Default |
+-------------------------------+----------------------+----------------------+
|   1  GeForce GTX TITAN   Off  | 0000:03:00.0     N/A |                  N/A |
| 30%   26C  N/A     N/A /  N/A |     14MiB /  6143MiB |     N/A      Default |
+-------------------------------+----------------------+----------------------+
|   2  GeForce GTX TITAN   Off  | 0000:83:00.0     N/A |                  N/A |
| 30%   27C  N/A     N/A /  N/A |     14MiB /  6143MiB |     N/A      Default |
+-------------------------------+----------------------+----------------------+
|   3  GeForce GTX TITAN   Off  | 0000:84:00.0     N/A |                  N/A |
| 30%   29C  N/A     N/A /  N/A |     14MiB /  6143MiB |     N/A      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Compute processes:                                               GPU Memory |
|  GPU       PID  Process name                                     Usage      |
|=============================================================================|
|    0            Not Supported                                               |
|    1            Not Supported                                               |
|    2            Not Supported                                               |
|    3            Not Supported                                               |
+-----------------------------------------------------------------------------+
```

Working with the Mogon

- Commands: direct in terminal
- Programming:
  - Terminal: emacs, vi, etc
  - gedit with x-server
  - wordpad with winscp, etc

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

Part 2

GPU programming with CUDA

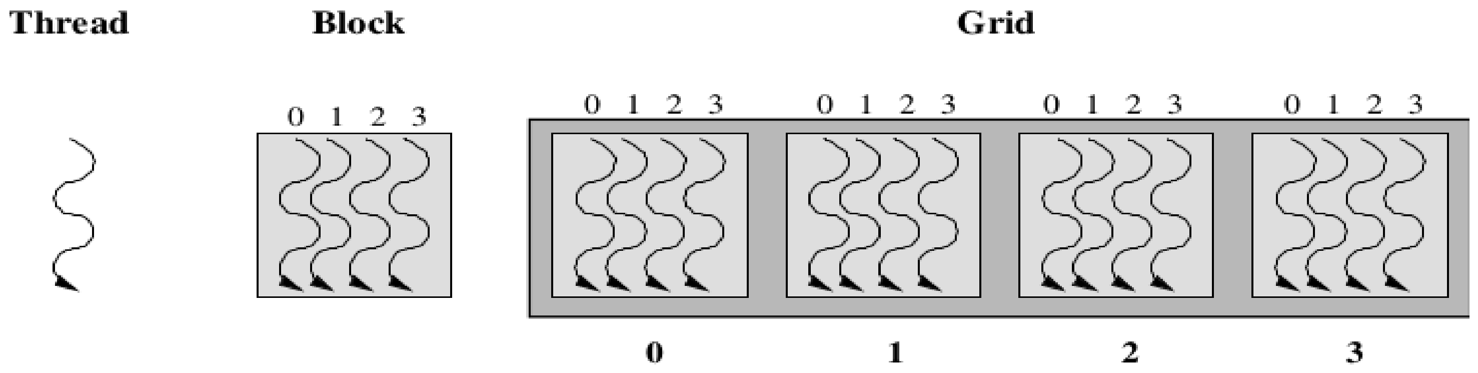
CUDA C Programming Guide

http://docs.nvidia.com/cuda/cuda-c-programming-guide/

The parallelism of CUDA

- Executing multiple **instances of kernel** by multiple CUDA **threads**
  *(**CUDA Kernel**: C function, executed only on GPU)*

- *Grid, Block and Threads*



- Example of 1-D Grid and Blocks:
  - Thread Id (*inside a block*): `threadIdx.x`
  - Block Id (*inside a grid*): `blockIdx.x`
  - Number of threads per block: `blockDim.x`
  - Number of blocks inside a grid: `gridDim.x`

Exercise-1: HelloWorld

Write a CUDA kernel such that each CUDA thread prints following text to the screen:

`Hello, I am thread` **`thread_id`**`, of block` **`block_id`**

Assume that grid and block has only one dimension. The number of threads and blocks are supplied by the program arguments. The following is an example:

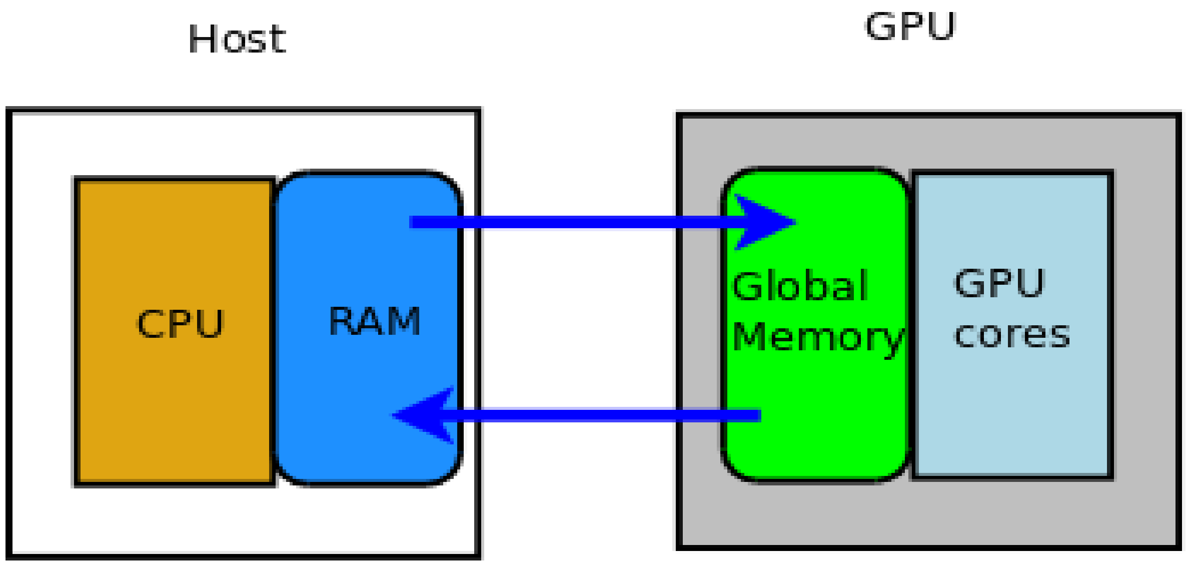
```
./helloworld 2 4
Hello, I am thread 0, of block 0
Hello, I am thread 1, of block 0
Hello, I am thread 2, of block 0
Hello, I am thread 3, of block 0
Hello, I am thread 0, of block 1
Hello, I am thread 1, of block 1
Hello, I am thread 2, of block 1
Hello, I am thread 3, of block 1
```

Exercise-1, manual

- Based on the given skeleton: `helloworld.cu`
- CUDA program code has a `*.cu` extension, compiled with `nvcc`
- To use `printf` inside a CUDA kernel, you must compile your program in `sm_20` or higher architecture `(sm_30, sm_35)`

```
nvcc helloworld.cu –o helloworld -arch=20
./helloworld
```

Host – GPU data exchange



- `For GPU: cudaMalloc(),cudaFree()`
- `cudaMemcpy()`

(http://docs.nvidia.com/cuda/cuda-c-programming-guide/#device-memory)

Exercise-2: Square an array

Write a CUDA program to square every element of a given array:

- The array should be read from a text file whose name comes from the program argument.
- The size of the array is given at the beginning of the input, each element has type of `int`.

The following is an example:

```
cat input.txt
8 1 2 3 4 5 6 7 8
./squareArray input.txt
1 4 9 16 25 36 49 64
```

Exercise-2, manual

- Based on the given skeleton: `squareArray.cu`
- Start with:
  - The number of threads = the size of an array
  - Using 1 block
- Given the number of threads per block, calculate the number of required blocks

Advanced Exercise

- Upgrade the simple version of of squaring an array in exercise-2 into a full application:
  - The size of array is large (upto 100 millions elements): the input file should be in binary format instead of text format
  - Checking error after each CUDA function call using `cudaGetLastError()`
    - **Ref:** http://www.drdobbs.com/parallel/cuda-supercomputing-for-the-masses-part/207603131
  - Measuring the elapsed time by using CUDA events
    - **Ref**: http://docs.nvidia.com/cuda/cuda-c-programming-guide/#events
    - Measuring the kernel execution time
    - Measuring the data transfering time between the host and the GPU
  - Write a serial function of squaring an array:
    - Compare 2 results to verify if it is correct
    - Compare the execution time to calculate the speed-up

Questions ?