Organizational remarks

Oliver Meister

October 23rd\textsuperscript{th} 2013
References

- D. Kirk, W. Hwu: *Programming Massively Parallel Processors*, Morgan Kaufmann, 2010
CUDA requirements

- (Linux) Check your hardware by typing `lspci | grep VGA` into a terminal
- Up-to-date graphics drivers (current version: 319.60)
- CUDA Toolkit (current version: 5.5)
- (Windows) Microsoft Windows XP, Vista, 7, or 8 or Windows Server 2003 or 2008
- (Windows) Microsoft Visual Studio 2008 or 2010, or a corresponding version of Microsoft Visual C++ Express
- (Linux) Up-to-date Linux version with gcc 4.x
CUDA installation steps

For details, refer to the Getting Started Guides for Windows/Linux/Mac on https://developer.nvidia.com/cuda-downloads

- (Linux) Run package manager, update package list and install
  
  sudo apt-get update
  sudo apt-get install cuda
gpuocelot framework

What if no CUDA capable device is available?

- CUDA emulator, works on CPU, AMD and Nvidia chips
- Not fully compatible with CUDA though
- Ubuntu 10.10 / 11.04: Debian packages available
- Otherwise: compile from source
Remote compilation

What if no CUDA capable device is available? (2)

- You can use one of our machines.
- Requires an account on our chair: Fill out the form with your full name, preferred account name (TUM, in.tum, ...), email address
- (Linux) Open an ssh connection via ssh -X atsccs30.informatik.tu-muenchen.de
- You will have to change your password after the first login using passwd
- GUIs should work, i.e. type firefox &
- These machines are *shared*, so try not to overload them. You can check their workload with *top*
Compiling CUDA-code

- (Linux) Open a terminal
- (Linux) Type `export PATH=$PATH:/usr/local/cuda/bin` to include the CUDA binary path
- (Linux) Again, type
  ```
  export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:
  ```
  followed by
  - `/usr/local/cuda/lib` for x86 systems
  - `/usr/local/cuda/lib64` for x64 systems
  This includes the CUDA library path
- Now, compile your code using `nvcc <source files.cu>`