Lab instructions

Multicore host machine

The first machine we'll use for the class exercises is a machine equipped with a Intel Xeon PHI Knights Landing (KNL) whose address and IP are respectively

- c6320p-2.itc.unipi.it
- 131.114.136.132

Accounts

All students registered on the course classroom have got an account named **spm18-<surname>**, with a password (automatically generated, should be changed at first access through a **passwd** command at the shell prompt) and may access the machine from anywhere in the world through **ssh**:

ssh spm18-<surname>@c6320p-2.itc.unipi.it

In order to be able to access the remote machine without giving the password explicitly, please follow these steps:

- On the machine you use to ssh, access your local public key (usually .ssh/id_rsa.pub under your home directory)
- Ssh to the remote machine using the assigned password
- Change the passwd (first time only) with the passwd command
- Add the id_rsa.pub to the .ssh/authorized_keys file taking care of
    - Ensuring the **.ssh** access rights are **drwx------** (only and exactly)
    - Ensure the **authorized_keys** file access rights are **-rw-r--r--**
    - Ensure that the id_rsa.pub key is copied into the file as a **single line** (no hidden CR characters)

The next time you'll access the machine no password will be required. Authentication will go on using the registered public keys. Be careful to have an id_rsa.pub key generated with an empty passphrase, otherwise you'll be asked to insert the passphrase each time.

GNU Compiler

At the moment being we have only the GNU compiler. This is the one in /usr/local/ and therefore add the following two lines to the .bashrc file you all have in your home folder:

```
export PATH=/usr/local/bin:$PATH
```

```
export LD_LIBRARY_PATH=/usr/local/lib64:$LD_LIBRARY_PATH
```

These are needed to prevent using the old version compiler installed by default under /usr/bin which I not removed as it is needed for default tool installation.
If you correctly set up the environment, next time you log you get access to the most recent version of the GNU compiler suite, which is:

```
[marcod@C6320p-2 ~]$ g++ --version
g++ (GCC) 7.3.0
Copyright (C) 2017 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is
NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.

[marcod@C6320p-2 ~]$ which g++
/usr/local/bin/g++
[marcod@C6320p-2 ~]$
```

Intel compilers and tools

Intel compilers and tools are located at /opt/intel/ . In order to be able to use the compiler, please add the following line (or execute it at the terminal prompt) to the .bashrc:

```
source /opt/intel/compilers_and_libraries/linux/bin/compilervars.sh intel64
```

which is needed to get the correct paths. Since the source command has been executed (at each login/shell creation if you put it into the .bashrc), you'll have access to icc compiler. In case you get a number of error messages is because of the missing license (still working to fix this).

This procedure has to be followed in case of any other Intel tools. E.g. to use vtune, please add

```
source /opt/intel/vtune_amplifier_xe/amplxe-vars.sh
```

or more in general execute the variable setup script under `/opt/intel/<specific-tool-directory>` to get specific tool available.

Currently installed tools and frameworks

We provide (in excess to normal Centos tool distribution):

- g++ 7.3
- Intel Tbb

- OpenMP (-fopenmp in g++)
- Cilk (-fcilkplus)
- FastFlow (under /usr/local/)
- GRPPI (under /usr/local/)
- Cmake 3.11 (under /usr/local)