INTELLECTUAL PROPERTY RIGHTS NOTICE:

• The User may only download, make and retain a copy of the materials for his/her use for non-commercial and research purposes.
• The User may not commercially use the material, unless has been granted prior written consent by the Licensor to do so; and cannot remove, obscure or modify copyright notices, text acknowledging or other means of identification or disclaimers as they appear.
• For further details, please contact BSC-CNS patc@bsc.es
Using Tibidabo
PRACE Training Course
Programming PRACE and MontBlanc prototypes

Gabriele Carteni

Barcelona, 25 May 2012
System Architecture
First of all, remember that Tibidabo is a prototype

- That is *not in production*: system administration on a best effort basis, limited user support, basic level of production services, frequent maintenance

Cluster configuration

- Minimal configuration: 1 Head Node and 128 Compute Nodes
- Neither I/O nodes nor dedicated login nodes are present
- Archiving system not available. Backup coming soon 😊

Filesystem Types

- ext3/ext4: journaled, less cpu-power, safe, secure, stable (400GB)
- NFSv3 (nodes rootfs, homes, scratch)
System Architecture

Service Network Topology
System Architecture

Production (MPI) Network Topology
System Architecture

Naming schema for compute nodes

node-${rr}-${bb}-${cc}-${nn}  
- rr: rack
- bb: blade, cc: column, nn: node

node-01-04-01-02
node-01-04-01-04
node-01-04-02-01
node-01-04-02-03

column 1  
column 2
System Software Overview
**System Software Overview**

**Head Node**
- Debian 6 (squeeze)
- Kernel 2.6.32
- PPC

**Compute Nodes**
- Ubuntu 10.10 (Maverick Meerkat)
- Kernel 2.6.36
- ARMv7 (ABI armel)
- Debootstrap tool
- Experimental kernel supporting ABI armhf
Software Stack

- Shorewall (masq and port forwarding management)
- NFS
- TFTP/BOOTP/DHCP/NTP
- tibidabo_scripts, pdsh
- SLURM
Access to Tibidabo
Access to Tibidabo

```
ssh myusername@tibidabo.bsc.es
```
Access to Tibidabo

ssh myusername@tibidabo.bsc.es

4 login nodes

SLURM
(mnsubmit mycode.job)

124 compute nodes
Running jobs with SLURM
Running jobs with SLURM

- SLURM is opensource and designed to operate in a heterogeneous cluster with up to 65,536 nodes and hundreds of thousands of processors
  - Developed at Lawrence Livermore National Laboratory (LLNL)
  - Now maintained by SchedMD LLC since 2010
- SLURM is also a scheduler (FIFO, backfilling, GANG)
  - Uses priorities, limits (queues) and shares (users/accounts)
  - Support for Generic Resources (GPU)
  - Support for external schedulers (LSF, MOAB/MAUI)
- SLURM DB (MySQL) for the accounting
Running jobs with SLURM

- mnsSubmit, mnq, mncancel
  - Wrappers around sbatch, squeue, scancel developed by BSC

- mnsSubmit myscript.job
  - myscript.job is a bash script with directives
  - Uses srun to submit a job
  - Directive syntax: #@ directive = value
Running jobs with SLURM

Example of a jobscript

gcarteni@node-01-01-01-03:~$ cat /gpfs/EXAMPLES/myslurm.job
#!/bin/bash
#@ initialdir = ./
#@ job_name = MyJob
#@ class = normal
#@ output = myjob_%j.out
#@ error = myjob_%j.err
#@ wall_clock_limit = 01:00:00
#@ total_tasks = 8
#@ cpus_per_task = 2
#@ tasks_per_node = 1

srun /gpfs/EXAMPLES/ompi/myopenmpi-app

Resources allocation and distribution
Running jobs with SLURM

```
mnq

nct01010@node-01-01-01-03:$ mnq

JOBID  NAME     USER      STATE    TIME  TIMELIMIT CPUS  NODES     NODELIST(REASON)
1926   MyJob    nct01010  RUNNING  0:03  1:00:00   16    8     node-01-02-02-
        node-01-03-01-
        node-01-03-02-01,
        node-01-05-01-01
1925   test_par nct01001  RUNNING  1:56  1:00:00   2    1     node-01-02-01-02

mncancel <JobId>
```
Thank you!

Let’s play later during the Hands On 😊
Developing apps on Tibidabo

PRACE Training Course
Programming PRACE and MontBlanc prototypes

Jorge Rodríguez (jorge.rodriguez@bsc.es)

Barcelona, 25 May 2012
We have a User Guide at:
http://www.bsc.es/support/Tibidabo-ug.pdf

- Updated often
- System overview
- Connection to Tibidabo
- How to run jobs?
- Software

- You can contact us at: support@bsc.es
**Modules**

- **Modules software environment:**
  - No need to set environment variables:
    - **PATH**
    - **LD_LIBRARY_PATH**
    - **MANPATH**
  - **Current modules:**
    - **GCC:**
      - 4.6.2
      - 4.7.0
    - **MPI:**
      - MPICH2/1.4.1
      - OpenMPI /1.5.4
Currently, only GCC is supported:

- module load gcc
  - C: gcc mpicc
  - C++: g++ mpic++
  - Fortran: gfortran mpif77/mpi90

Software:
- GROMACS
- PETSC
- DDT
Libraries

- Extrae libraries: (OpenMPI, MPICH2)
  - libmpitrace
  - libmpitracef
  - libomptrace
  - ...

- SCALAPACK

- ATLAS

- LIBUNWIND

- FFTW
  - From version 3.3.2 : ARM Vector Instructions
Compiler flags

- `mcpu=cortex-a9`
- `mtune=cortex-a9`
- `march=armv7-a`
- `mfloat-abi=softfp`
- `mfpu=vfp`

NEON

- General-purpose SIMD engine for Cortex-A series
- Own independent pipeline and register file
- It has 32 registers, 64-bits wide / 16 registers, 128-bits wide
- Registers are considered as vectors of elements of the same data type
  - Data types can be:
    - int: signed/unsigned 8-bit, 16-bit, 32-bit, 64-bit
    - float: single precision floating point
C Basic Vector Data types:

```c
#include <arm_neon.h>

- int8x8_t  uint8x8_t
- int16x4_t  uint16x4_t
- int32x2_t  uint32x2_t
- int64x1_t  uint64x1_t
- int8x16_t  uint8x16_t
- int16x8_t  uint16x8_t
- int32x4_t  uint32x4_t
- int64x2_t  uint64x2_t
- float32x2_t
- float32x4_t
```
VADD.I16 D0, D1, D2

VMUL.I32.S16 Q0, D2, D3;
You may enable auto-vectorization in your code:

```
-mfpu=neon -ftree-vectorize
```

**Advantages:**

- No need to know the underneath technology
- More portable code
- Faster code
Thank you!

Let’s continue during the Hands On session
Tuning and tracing applications
PRACE Training Course
Programming PRACE and MontBlanc prototypes

Nikola Puzovic
Outline

**Tuning (a simple example)**
- FP micro-benchmark
- Understanding the importance of correct gcc flags

**Tracing**
- HYDRO benchmark (PRACE)
  - MPI+OmpSs
  - MPI
All examples can be found in

/gpfs/EXAMPLES/training/prace_tracing.tgz

prace_training/
  – synth/
  – hydro_mpiompss/
  – hydro_mpi/
  – ...

Programming PRACE and MontBlanc prototypes: Tibidabo Machine
Prerequisites

**PARAVER**
- Installed on your local machine

http://www.bsc.es/computer-sciences/performance-tools/downloads

**OmpSs configuration files**
- `user_functions.cfg`
  - To look at tasks
- `thread_state.cfg`
  - To look at thread state (executing, idle, synchronization…)
- Available in `prace_training/` folder
Synthetic benchmarks

Microkernels
- To test the FP performance of Cortex-A9 CPU
- Developed to see if we can reach max GFLOPS as advertised
- We will use it to test the importance of correct gcc flags

prace_training/synth
double *A;
double accum1, ...;

gettimeofday(&start, 0);

for (j=0; j<t; j++) {
    acum = 0; acum2 = 0;
    acum3 = 0; acum4=0;
    acum5 =0; acum6=0;
    acum7 =0;
    
    for (i=n; i!=0; i--) {
        acum += A[i];
        acum2 += A[i];
        acum3 += A[i];
        acum4 += A[i];
        acum5 += A[i];
        acum6 += A[i];
        acum7 += A[i];
    }
}

gettimeofday(&end, 0);

**Sums all elements of an array**
- Double-precision FP
- Repeats for a given number of times

**1 GFLOPS**
- Expected when everything fits into cache
double *A;
double accum1, ...;

timeofday(&start, 0);

for (j=0; j<t; j++) {
    acum  = 0; acum2 = 0;
    acum3 = 0; acum4 = 0;
    for (i=0; i<n; i++) {
        acum  += A[i] * B[i];
        acum2 += A[i] * B[i];
        acum3 += A[i] * B[i];
        acum4 += A[i] * B[i];
    }
}

timeofday(&end, 0);

---

**Vector dot product**
- Double-precision FP
- Repeats for a given number of times

**1 GFLOPS**
- Expected when everything fits into cache

---

Programming PRACE and MontBlanc prototypes: Tibidabo Machine
gcc flags for Tibidabo

```bash
-march=armv7a -mcpu=cortex-a9 -mtune=cortex-a9
```

- Specifies the target CPU
  - gcc chooses the correct instructions to emit
  - Activates CPU-specific optimizations

```bash
-mffloat-abi=softfp
```

- Generates HW floating point instructions
- Soft-FP calling conventions (affects function calls)

```bash
-mfpu=vfp -mfp=vfpv3-d16
```

- Specifies floating point hardware that is available in the CPU
The importance of correct flags

 Execute synthetic benchmarks

 make

 mnsSubmit launch-synth.sh

 Observe the difference in reported MFLOPS

 ~30x
HYDRO

RAMSES
- Large-scale simulation of galaxy formation
- Scalable to tens of thousands of cores

HYDRO
- Simplified version of RAMSES
- Solves compressible Euler equations of hydrodynamics
- Official PRACE 1IP benchmarks for prototypes
  - MPI, MPI+OpenMP, CUDA, …
- Authors
  - Pierre-François Lavallée, Guillaume Colin de Verdière, Philippe Wautelet, Dimitri Lecasa, Jean-Michel Dupays
  - CEA, IDRIS/CNRS
**Task decomposition**

- **Input:** an array
- **Two passes**
  1. Updates each row of the array based on the previous
  2. Updates each column
- **Configurable number of tasks**
Tracing MPI+OmpSs

 Execute the application with tracing
   - mnssubmit trace-job.sh
   - Changes for tracing:
     • export NX_ARGS="--pes 2 --instrumentation=extrae
      --extrae-keep-mpits --extrae-skip-merge"
     • srun ./trace.sh ./HydroC -i input_file

#!/bin/bash

export EXTRAE_CONFIG_FILE=./extrae.xml
export EXTRAE_HOME=/gpfs/CEPBATOOLS/extrae/latest/openmpi
export TRACE=TRACE_$$

export NX_ARGS="--pes 2 --instrumentation=extrae --extrae-keep-mpits
--extrae-skip-merge"

export LD_PRELOAD="${EXTRAE_HOME}/lib/libnanosmpitrace.so"

@

# Merge the trace...
IGNORE REST OF THE SCRIPT
Effects of the block size

- See the effects of the block size
  - $nx, ny = 1024$
  - Initially, $nxystep = 384$

- To execute and to trace
  
  `prace_training/hydro_mpiopmss`

  `mnssubmit trace-job.sh`

- Copy the resulting trace to your machine and analyze it
Unbalanced execution

- SYNCHRONIZATION == Master starts taskwait

Optimization:

\[ nxystep = 384 \rightarrow 256 \]
Imbalance resolved!

Programming PRACE and MontBlanc prototypes: Tibidabo Machine
Other examples

MPI version of the hydro
  – hydro_mpi
INTELLECTUAL PROPERTY RIGHTS NOTICE:

• The User may only download, make and retain a copy of the materials for his/her use for non-commercial and research purposes.

• The User may not commercially use the material, unless has been granted prior written consent by the Licensor to do so; and cannot remove, obscure or modify copyright notices, text acknowledging or other means of identification or disclaimers as they appear.

• For further details, please contact BSC-CNS patc@bsc.es