Experience with UNICORE Services for Multiscale Materials Modelling

Michele Carpené
Super Computing Application & Innovation (SCAI)
m.carpen@cineca.it

Agenda

Outline:

- Introduction – Overview of MMM@HPC
- MMM@HPC and UNICORE
- UNICORE in CINECA – The UNICORE CINECA Site
- The PLX UNICOREX configuration
- The UNICORE TSI – How the code has been modified
  - Code Changes
  - Remarks
- Conclusions
- Acknowledgments
- References

Introduction

MMM

The aim of the **MMM@HPC** project (**http://www.multiscale-modelling.eu/**) is to build a distributed **HPC/HTC** environment for Multiscale Materials Modelling in nano material science.

The integration of individual simulation codes operating on different size and time scales is one of the main objectives in the project.

Main actors: CINECA, CSC, KIT and KIST

Main technologies used:

Codes for Computational Chemistry (MOPAC, DEPOSIT, OpenBabel, Pairfinder)

UNICORE Client + Services

GridBeans

Java

Introduction

Our intent is to give a brief overview of the activities that have been carried out in CINECA (http://www.cineca.it) to help MMM@HPC researchers to use UNICORE by describing how we constructed our site configuration, both to accommodate the characteristics of our batch scheduler and the user environment on the cluster.

User/Researcher

CINECA Staff

MMM & UNICORE

Multiscale Materials Modelling

- Several approaches adopted;
- Multiscale model stripped down to multiple **submodels** called **steps**, that can be solved using different standard simulation code;
- To construct a workflow the workflow editor of the UNICORE Rich Client (URC) is used;
- The simulation codes are integrated in the workflow using GridBeans.

UNICORE:

- Site administrators create an entry in the UNICOREX IDB for each MMM simulated code supported by the workflow;
- GridBean jobs are submitted to the CINECA site from URC.

UNICORE CINECA Site overview - 1

Front End Cluster (FEC)

Gateway/Registry/Xuudb/ Workflow Engine + Service Orchestrator

PLX

PBS/qsub

UNICORE

XNJS+TSI

SP6

LoadLeveler/ llsubmit

DECOMMISSIONED on the second quarter of 2012

CINECA PLX is the reference cluster for the MMM project

The PLX is an IBM Cluster provided with GPUs

the peak performance of the whole cluster
is about 30 Tflops if only the Intel Westmere cores are used, but this rises to above 300 Tflops if the GPUs can be exploited as well.

UNICORE CINECA Site overview - 3

CINECA: PLX GPU Linux Cluster Model: IBM PLX (iDataPlex DX360M3)

- **Tier 1**
- **Resources:**
  - **Architecture: Linux Infiniband Cluster**
  - **Nodes: 274 IBM iDataPlex M3**
  - **Processors: 2 six-cores Intel Westmere 2.40 GHz per node (548 processors, 3288 cores in total)**
  - **GPU: 2 NVIDIA Tesla M2070 per node (for 264 nodes) + 2 NVIDIA Tesla M2070Q per node (for 10 nodes) for a total of 548 GPUs**
  - **RAM: 48 GB/node**
  - **Internal Network: Infiniband with 4× QDR switches**
  - **Disk Space: 100 TB**
  - **Operating System: Red Hat RHEL 5.6**
  - **Peak Performance: 300 TFlop/s (142 TFlops sustained - Linpack benchmark)**
- **Access: ssh**
- **Software: Local Batch Scheduler (PBSpro 10.4) + UNICORE**
- **Provided with access to GPFS FS (CINECA SCRATCH + CINECA DATA)**

UNICORE CINECA Site overview - 4

Software tools/libraries made available on PLX:

- **Mopac**
- **Deposit**
- **Pairfinder**
- **OpenBabel**

The executable paths of these applications have been added inside the UNICORE idb configuration file.

We're working also to a “mmmhpc” module to load automatically a proper MMM@HPC user environment

The PLX UNICORE Server configuration

First, the **UNICOREX idb file** was configured to meet the underlying infrastructure and user requirements:

- Application paths have been configured inside the idb file;

- **MPI ExecutionEnvironment** has been configured inside the idb file to invoke **mpirun** command when specified from the URC (on PLX the scheduler does not invoke automatically mpirun as on SP6);

- Added the **nGPUs** resource to set the GPUs number in the URC;

- Introduced a **choice** to specify the batch queue (**parallel, longpar** and **debug).**

The UNICORE TSI

The UNICORE TSI version initially installed on the PLX **didn't match completely the underlying infrastructure** characteristics and there was no TSI compatible with the scheduler version on the **PLX (PBSpro 10.4)**.

A new version of the PBS TSI was created

The original Submit.pm module has been modified respect to the original version including deep modifications and new added code parts.

PROBLEM:

Different Scheduler versions → **different syntax** to define resources and **different configurations**.

The UNICORE TSI – Code Changes

1. All parameters are now collected inside a **TSI bash script** according to **PBSPro** syntax (job description) → the TSI submits the executable to the batch system;

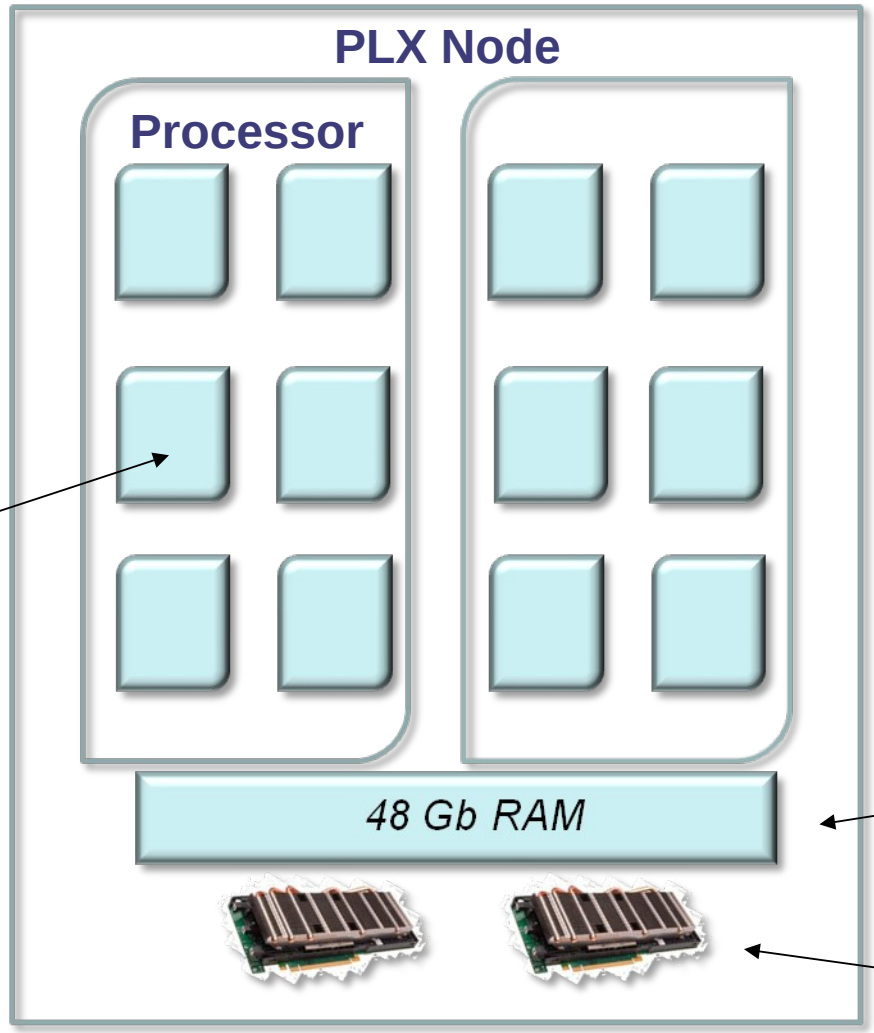
2. Introduced **ngpus** as a PBS parameter into the TSI bash script;

3. An entry has been added to set the ncpus parameter which defines the number of physical cores, this is the right syntax:

"#PBS -l select=1:ncpus=$ncores";

But on the **CINECA PLX** we cannot select more then 12 cores for a node: **PROBLEM! If select=1 and cores requested > 12 the job is not executed...**

The UNICORE TSI – Code Changes



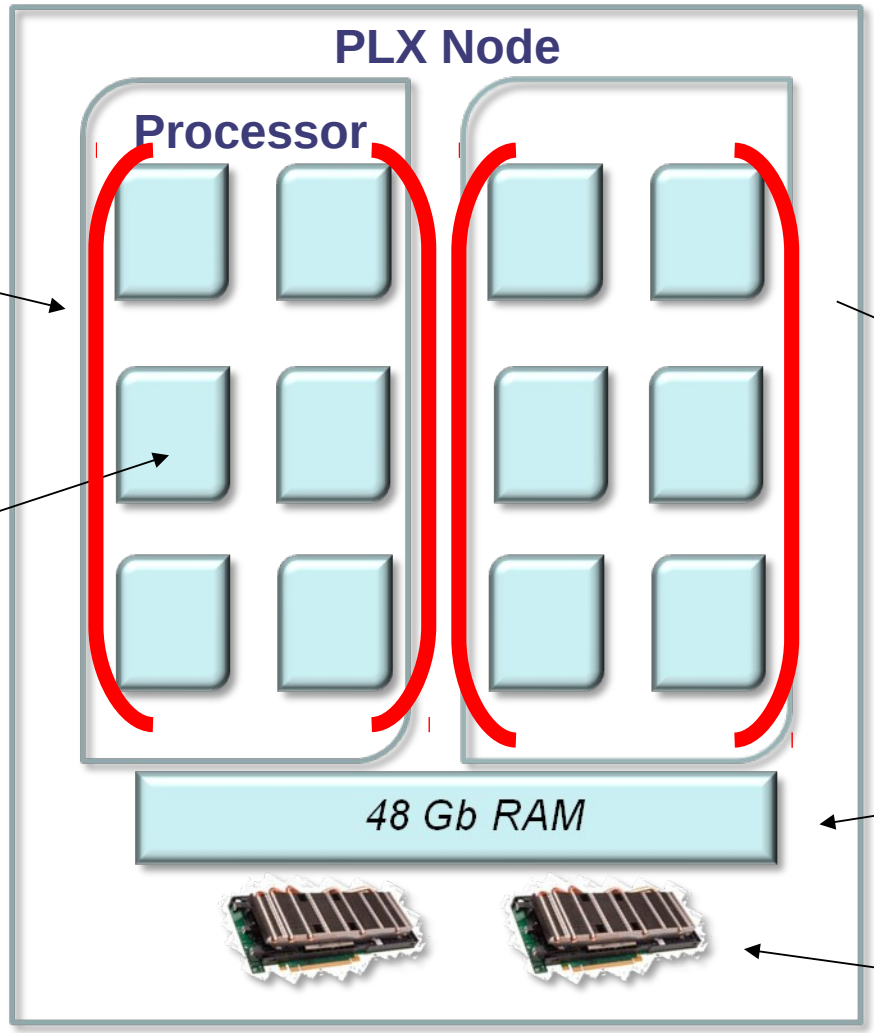
cores per processor (Max 12 available per node)

Shared memory

2 GPUs per node

The UNICORE TSI – Code Changes

Example: we want To select 14 cores, 12 in the first node and 2 cores in a second node



We need another 2 cores chunk

cores per processor (Max 12 available per node)

Shared memory

2 GPUs per node

The UNICORE TSI – A partial solution

3. New algorithm (specific for our PBS BSS) to select the right cores number depending on the user request (no more than 12 cores for a chunk/node);

```
# pseudo code
if $ncores <=12
$resource="#PBS -l select=1:ncpus=$ncores:mpiprocs=$ncores"
if $ncores>12 {
$nchunks=int($ncores/12);
$rem=$ncores % 12;
## if we have an exact multiple of 12 cores use chunks of the same size
if $rem=0 #reminder
$resource="#PBS -l select=$nchunks:ncpus=12:mpiprocs=12";
## add any remainder to a second chunk
if $rem >0
$resource .= "+ncpus=$rem:mpiprocs=$rem";
}
```

Remarks

During User Supporting activities some relevant considerations have been made:

1. There is no ”simple way” to change the TSI – system administrators are forced to modify the perl code to allow for particular customisations for different schedulers or local configurations;

2. With UNICORE we cannot emulate the full flexibility of PBS with respect to resource specification. For example, it is not possible to specify different ”chunk” sizes as in the following PBS example:

#PBS -l select=2:ncpus=8:mpiprocs=8+1:ncpus=5:mpiprocs=5

The UNICORE TSI – Code Changes

Finally: added a script to find automatically the **project account** entry in the bash script (specific for CINECA users).

```
###### CINECA specific for PLX ######
    my $account_number="";
    my $varacc = $ENV{USER};
    my $saldo = `/cineca/bin/saldo -b`;

    if ($saldo ne "username not existing") {
        my @lines=split(/\n/m,$saldo);
        chomp @lines;
        my @fields=split(" ",$lines[5]);
        $account_number=$fields[0];
        if ($account_number ne "")
            {
                $account_no = "#PBS -A $account_number";
                debug_report("account number: $account_no\n");
            }
    }
```

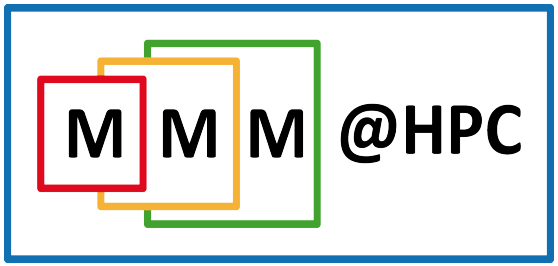
Conclusions

- The case of MMM@HPC has been perfectly fitting to highlight necessities of UNICORE users, anyway limits we encountered do not affect UNICORE basic mechanisms;

- We highlighted problems arose from the necessity to fully exploit the features of our machine, the IBM PLX cluster, and its hardware and scheduler capabilities;

- These limitations may be considered basically overcome with a few coding ... but a path towards a general solution is not yet completely clear (e.g. Scheduler customization, different resource allocation).

Aknowledgments



This work has been funded by the 7th Framework Programme of the European Commission within the Research Infrastructures with grant agreement number RI-261594, project MMM@HPC.

References

1. CINECA http://www.cineca.it/

2. PRACE http://www.prace-ri.eu/

3. Multiscale Material Modelling on High Performance Computer Architectures (MMM@HPC) http://www.multiscale-modelling.eu/

4. Kondov, I.; Maul, R.; Bozic, S.; Meded, V., and Wenzel, W.; UNICORE-Based Integrated Application Services for Multiscale Materials Modelling.

5. Bozic S.; Kondov I.; Meded, V.; Wenzel, W.; UNICORE based Workflows for the Simulation of Organic Light-Emitting Diodes (ibid.)

Questions?

☺☺