Cluster Nazionale CSN4

Parallel computing and scientific activity

Roberto Alfieri - Parma University & INFN, Gr.Coll. di Parma

LNL - 16/02/2011

Outline

- PC-clusters in the theoretical physics community

- The CSN4cluster project

- New granularity attributes on the grid

- CSN4cluster jobs submission

- Examples of physics parallel applications executed on the cluster

- Conclusions

Parallel computing in the INFN theo-phys community

Lattice simulations (local communication oriented): **APE projects**

General purpose (parallel and serial): **PC clusters**

→ 2005 : A lot of clusters small-medium sized - often managed by users

2005-2007 : Single centralized cluster (Cnaf) - 24 nodes Xeon +infiniband

2007-2009 : 4 PC clusters based on federated projects - grid access for serial jobs

2010 → : Single centralized cluster (“CSN4cluster” project)

CSN4cluster: timeline

late 2009: collaboration CSN4-CCR to define cluster requirements and evaluate sites proposal

Febr. 2010: INFN-Pisa project approved

June 2010: cluster in operation for sequential jobs

July 2010: call for scientific proposals

Sept. 2010: 15+1 projects approved and fair-shares defined

Dec. 2010: cluster in operation for parallel jobs

CSN4cluster access: theophys VO

Access method: via Infngrid only (both serial and parallel jobs)

Access policy: - thanks to G. Andronico (CT) -

- Theophys VO members (~124 up to now) with low priority

- Theophys subgroups (or others) can apply for a granted fairshare to the CSN4 cluster committee

Active fairshare grants:

- 16 has already been assigned, corresponding to 16 CSN4 IS proposals
- Requests: 130K day*core serial + 250K day*core parallel = 380K day*core

  - availability: 365 K day*core per year -

Details: http://wiki.infn.it/cn/csn4/calcolo/

MPI and multi-thread support in EGEE

MPI has always been supported by EGEE but..

- Survey for users and administrators in April 2009: **MPI is still scarcely used**
- **Multi-thread programming** should be supported in EGEE to exploit the upcoming multi-core architectures.

=> Set-up of a new **EGEE MPI-WG**.

Recommendation document released in 06/2010: http://www.grid.ie/mpi/wiki/WorkingGroup

New attributes in the JDL are proposed by the MPI-WG for multi-thread support

Attribute	Meaning
CPUNumber=P	Total number of required CPUs
SMPGranularity=C	Minimum number of cores per node
HostNumber=N	Total number of required nodes
WholeNodes=true	Reserve the whole node (all cores)

Granularity attributes : JDL examples

WholeNodes=false (not set)

```
CPUNumber = 24;          # Default: 24 CPUs any number of nodes
```

```
CPUNumber = 64;          # 32 nodes, with 2 CPUs per node
SMPGranularity = 2;      # (SMPsize >=2 )
```

```
CPUNumber  = 16;         # 2 nodes, with 8 CPUs per node
HostNumber = 2;          # (SMPsize >=8 )
```

WholeNode=true

```
WholeNodes=true;         #2 whole nodes with SMPsize>=8
HostNumber=2;
SMPGranularity=8;
```

```
WholeNodes=true;         # 1 whole node with SMPsize>=8
SMPGranularity=8;        # (default HostNumber=1)
```

```
WholeNodes=true;         # 2 whole nodes with SMPsize>=1
HostNumber=2;            # (default SMPGranularity=1)
```

Granularity support: preliminary patch

New JDL attributes proposed by the MPI-WG **aren’t implemented in gLite yet**

A **preliminary patch for Cream-CE** has been developed and tested in collaboration with the gLite middleware developers

- thanks to M. Sgaravatto (PD), S. Monforte (CT), A. Gianelle (PD) -

The patch has been **installed on the CSN4cluster** and is now operational with a temporary syntax - waiting for the final integration of the attributes in gLite -

Temporary syntax JDL examples:

```
CeRequirements = “wholenodes=\”true\”  && hostnumber==2”;     # 2 whole nodes
```

```
CPUNumber = 16;                                    # 8 nodes with 2 CPUs per node
CeRequirements = “SMPGranularity==2”
```

CeRequirements are interpreted by the CEs.

Match-Making process is not involved.

CSN4cluster: computational resources

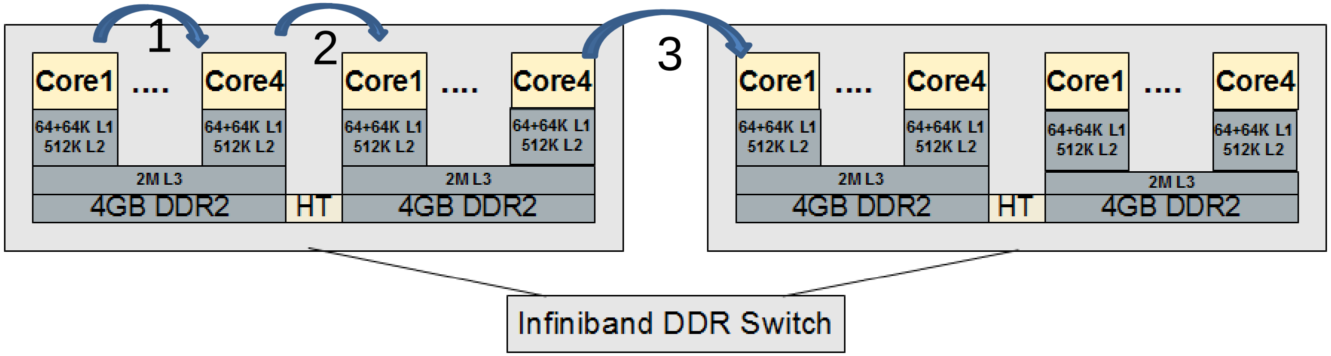
Resources: - thanks to E. Mazzoni, A. Ciampa, S. Arezzini (PI) -

1 CE gridce3.pi.infn.it (Cream - LSF)

128 WNs Dual-Opteron 8356 (2x4 cores per node) ≈ 10 TFlops peak perf.

In modern multicore processors the memory architecture is **NUMA**

- Cpu/memory **affinity** is the ability to bind a process to a specific CPU/memory bank -



Measured network performance (using NetPIPE):

	Comm Type	Latency	MAX Bandw.
1	Intra-socket	640 ns	14 GBytes/s
2	Intra-board	820 ns	12 GBytes/s
3	infiniband	3300 ns	11 GBytes/s

Memory performance (peak):

Memory Type	Latency	MAX Bandw.
L3 cache	≈35 ns	
DDR3	≈50 ns	≈32 GBytes/s
Numa (HT or QPI)	≈90 ns	≈11 GBytes/s

CSN4cluster: resources access

Direct job submission to Cream-CE in the JDL

```
Requirements =(other.GlueCEInfoHostName == "gridce3.pi.infn.it")
```

gives access to 2 queues:

- **theompi** : parallel job only - reservation time 8h -

  Role=parallel required

```
voms-proxy-init –voms theophys:/theophys/<group_name>/Role=parallel
```

- **theoshort**: serial short jobs - runtime 4h -

  Role=parallel should not be specified

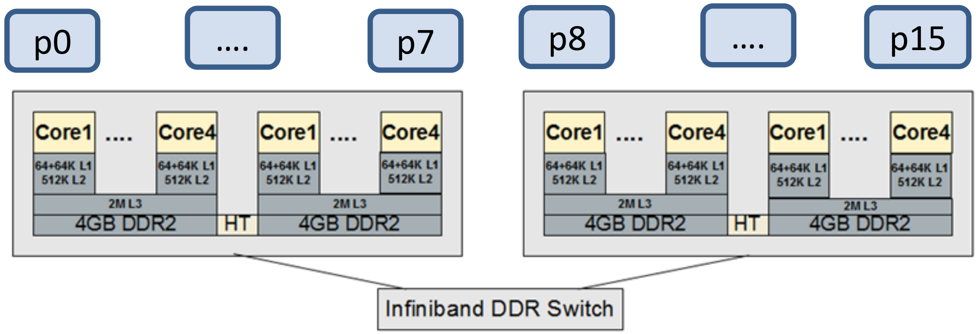
```
voms-proxy-init –voms theophys:/theophys/<group_name>
```

- The serial queue allows the exploitation of cores when they are unused by parallel jobs -

MPI job : explicit submission

Direct job submission means we know SMP architecture, MPI flavours, ecc..

- This example executes 16 MPI ranks (2 whole nodes) -



LSF hostfile

```
csn4wn110
csn4wn110
csn4wn110
csn4wn110
csn4wn110
csn4wn110
csn4wn110
csn4wn110
csn4wn111
csn4wn111
csn4wn111
csn4wn111
csn4wn111
csn4wn111
csn4wn111
csn4wn111
```

JDL

```
Executable = "mpi.sh";
Requirements =(other.GlueCEInfoHostName == "gridce3.pi.infn.it");
CeRequirements = "wholenodes==\"true\"  && hostnumber==2";
```

mpi.sh

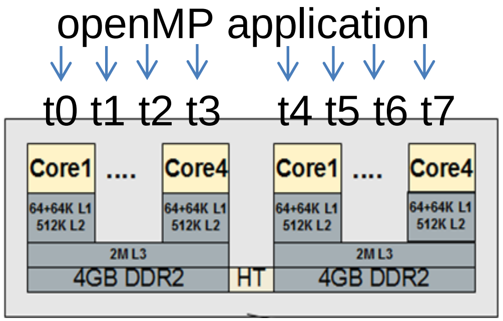
```
NP=$[`cat $HF | wc --lines`]
mpirun -np $NP -hostfile $HF mympi
```

openMP job

Wholenodes allows the submission of **multi-thread jobs on the grid**

- This example executes 8 openMP threads on a whole nodes -

- The user should be aware of potential memory affinity impact on performance -



JDL

```
Executable = "openmp.sh";
Requirements =(other.GlueCEInfoHostName == "gridce3.pi.infn.it");
CeRequirements = "wholenodes==\"true\"  && hostnumber==1";
```

openmp.sh

```
export OMP_NUM_THREADS=8
./myomp
```

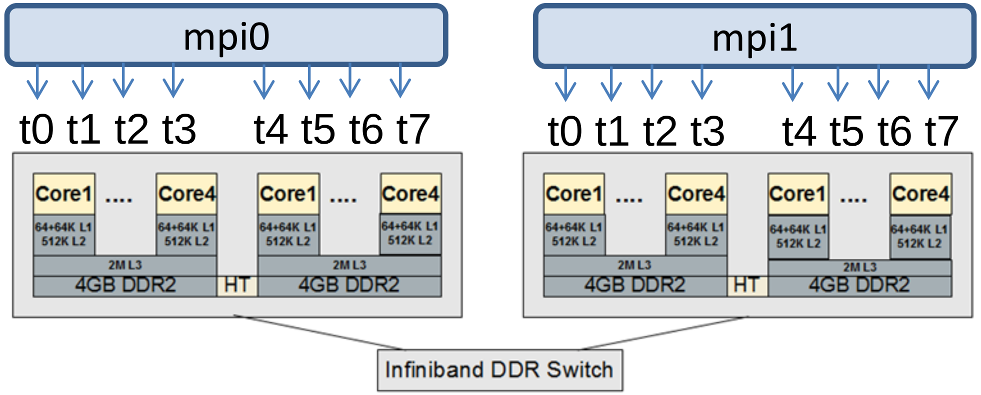
LSF hostfile

```
csn4wn110
csn4wn110
csn4wn110
csn4wn110
csn4wn110
csn4wn110
csn4wn110
csn4wn110
```

Hybrid MPI-openMP job

Hybrid parallel programming on the grid is enabled too.

- This example requires 2 MPI ranks. Each MPI process will launch 8 openMP threads -



JDL

```
Executable = "mpi_openmp.sh";
Requirements =(other.GlueCEInfoHostName == "gridce3.pi.infn.it");
CeRequirements = "wholenodes==\"true\"  && hostnumber==2";
```

New hostfile

```
csn4wn110
csn4wn111
```

mpi_openmp.sh

Generate new hostfile

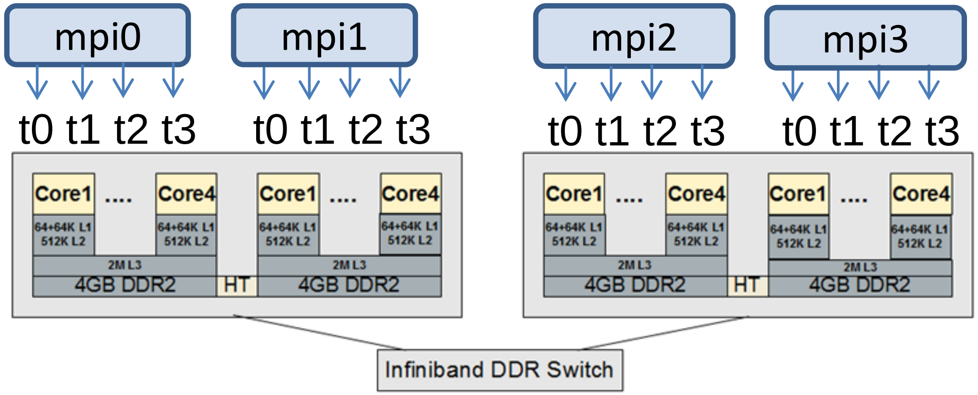
```
cat $HF | sort –u > $HF2.txt
NP=$[`cat $HF2  |  wc --lines`]
mpirun -np $NP  -x OMP_NUM_THREADS=8  --hostfile $HF2   mympiomp
```

Hybrid MPI-openMP job with affinity

The openMPI rankfile supports the **CPU affinity**

- binding of an MPI process to a specific core or range of cores –

- This example requires 4 MPI ranks. Each MPI process will launch 4 openMP threads -



rankfile

```
rank 0=csn4wn110 slot=0-3
rank 1=csn4wn110 slot=4-7
rank 2=csn4wn111 slot=0-3
rank 3=csn4wn111 slot=4-7
```

new hostfile

```
csn4wn110
csn4wn111
```

JDL

```
Executable = "mpi_openmp.bash";
Requirements =(other.GlueCEInfoHostName == "gridce3.pi.infn.it");
CeRequirements = "wholenodes==\"true\"  && hostnumber==2";
```

mpi_openmp.bash

Generate new hostfile

Generate rankfile

```
cat $HF | sort –u > $HF2.txt
awk '{print "rank " i++ "=" $1 " slot=0-3" "\n" "rank " i++"="$1 " slot=4-7"}' $HF2 > $RF
NP=$[`cat $RF |  wc --lines`]
mpirun -np $NP -x OMP_NUM_THREADS=4 --hostfile $HF2 --rankfile $RF mpiomp_exec
```

MPI job submission via MPI-start

If a **higher level of abstraction** is needed (i.e. don’t know where the MPI job will land) we have to use the **MPI-start** wrapper.

MPI-start is the submission method recommended by the EGEE MPI-WG.

The current version of **mpi-start is not able** to manage hybrid mpi-openMP application and memory/CPU affinity.

- This example executes 16 MPI ranks (2 whole nodes) -

JDL:

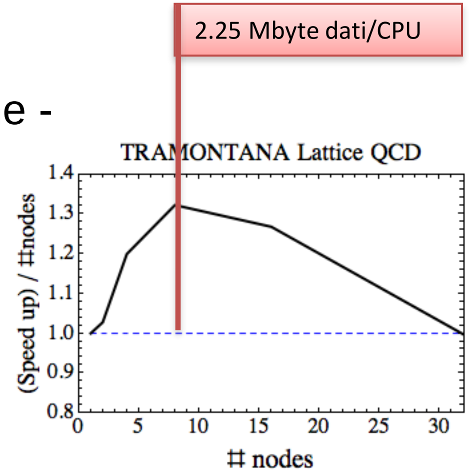
```
Executable = "mpistart-wrapper.sh";
Arguments = “mympi OPENMPI";
InputSandbox = {"mpistart_wrapper.sh","mpi-hooks.sh",“mympi.c"};
#Requirements =(other.GlueCEInfoHostName == "gridce3.pi.infn.it");
CeRequirements = "wholenodes==\"true\"  && hostnumber==2";
```

- **mpistart_wrapper.sh** is a standard script. Modification not needed
- **mpi-hooks.sh** includes pre and post execution scripts

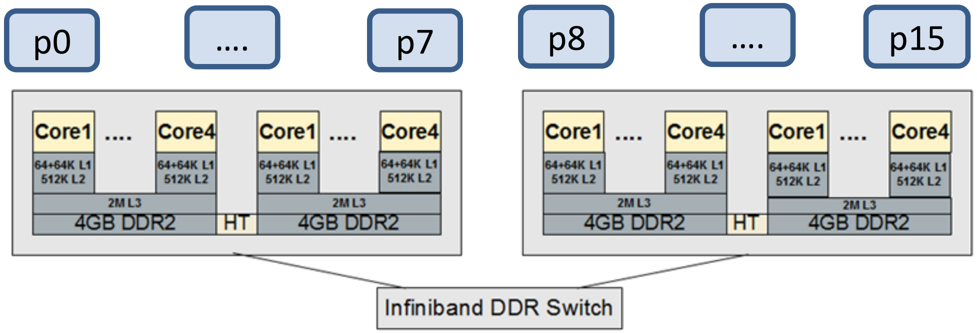
Results of real applications: USQCD

Hybrid-Montecarlo simulation of the Pure Gauge SU(3) on a 32x32x32x8 lattice (2000 sweep) using the publicly available **USQCD collaboration** “chroma” library (http://usqcd.jlab.org/usqcd-docs/chroma/). - Thanks to A. Feo, (Turin U.) -

- **Pure MPI code**
- Total memory occupation of the grid **~36MBytes**
- Importance of **memory affinity** - when all the data are not in cache -
- Cache effect - efficiency >1 -


2.25 Mbyte dati/CPU

Np	**8** (1x8)	**16** (2x8)	**32** (4x8)	**64** (8x8)	**128** (16x8)
Non-ranked	295 min	146 min	62 min	27 min	14 min
Ranked	287 min	139 min	59 min	27 min	14 min



Results of real applications : NumRel

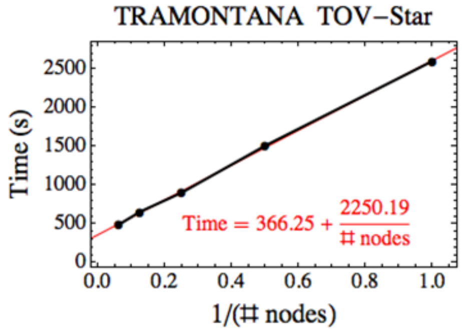
Evolution of a stable general relativistic TOV-Star using the **Einstein Toolkit consortium** codes (http://einsteintoolkit.org/). - Thanks R. De Pietri, Parma U. -

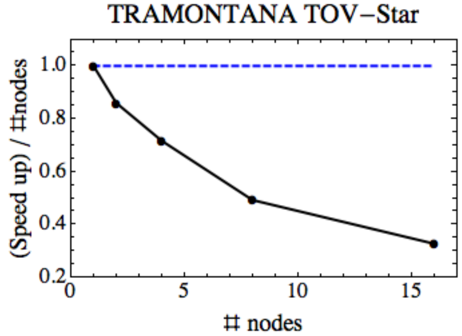
Hydro-dynamical Simulation of a perfect fluid coupled to the full Einstein Equations (dynamical space-time) on a 3-dimensional grid with 5-level of refinement spanning an octant of radius 177 *km* with a maximum resolution within the star of 370 *m*.

- Hybrid MPI-openMP code

- Total memory occupation of the grid **~8GByte**.
- The code is NOT full-parallelized but large memory request require parallelization.

#node	Np=8x#	Np=4x#	Np=2x#	Np=#	Np=2x#
	Nt=1	Nt=2	Nt=4	Nt=8	Nt=4 (rank)
1	2291.90	2934.21	3126.73	3360.96	**2608.08**
2	1438.72	1619.83	1797.30	2061.55	**1516.04**
4	1007.71	993.79	1007.71	1268.79	**909.36**
6	767.45	783.07	694.31	927.35	**745.63**
8	663.03	638.81	694.31	753.79	**661.37**
16	461.85	448.77	484.20	552.89	**497.78**





Conclusions and future work

The Cluster is ready for use. Now we need dissemination events/activity:

- Wiki page: http://wiki.infn.it/cn/csn4/calcolo/csn4cluster/
- Cluster inauguration (April 2011)
- Training course (date subject to fund availability from INFN Administration)

This model **can be extended to other MPI sites,** after the integration of the Granularity attributes in gLite.

MPI-start should support affinity:

The current version of MPI-start in not able to manage hybrid MPI-openMP application and memory/CPU affinity.

- Management of CPU/memory affinity is important: the migration of process with respect to allocated memory has an impact on performance.
- Hybrid MPI-openMP applications needs an adapted MPI-start able to suggest the right number of threads.

Thank you

for your attention!