Enabling Grids for E-sciencE

Astronomy and Astrophysics - Requirements and experiences with use of MPI in Grid Infrastructures

S. Orlando[1] *and the COMETA Consortium*

[1] *INAF - Osservatorio Astronomico di Palermo*

www.eu-egee.org

Outline

- HPC vs. Grid
- Key requirements for HPC applications
- The PI2S2 Project and the Sicilian Grid Infrastructure
- MPI modifications to gLite middleware
- Other requirements:
  scheduling policy, job monitoring, long term proxy
- Priorities to make grid facilities competitive with traditional HPC facilities

REFERENCE: Iacono-Manno et al. 2009, IJDST in press

HPC vs. Grid

- **Grids maximize the overall infrastructure exploitation**

  quality policies address the performance of whole infrastructure over long periods

  (e.g. total number of jobs run over a month)

- **HPC users have in mind the time performance of their applications as the most relevant parameter**

BASIC DIFFERENCES

- **Concept**
  - **HPC clusters: dedicated to HPC applications (often MPI based)**
  - **Grid Infrastructures: multi-purpose**
- **Hardware:**
  - **HPC clusters: fastest processors and low latency net connection**
  - **Grid Infrastructures: largest number of processors**

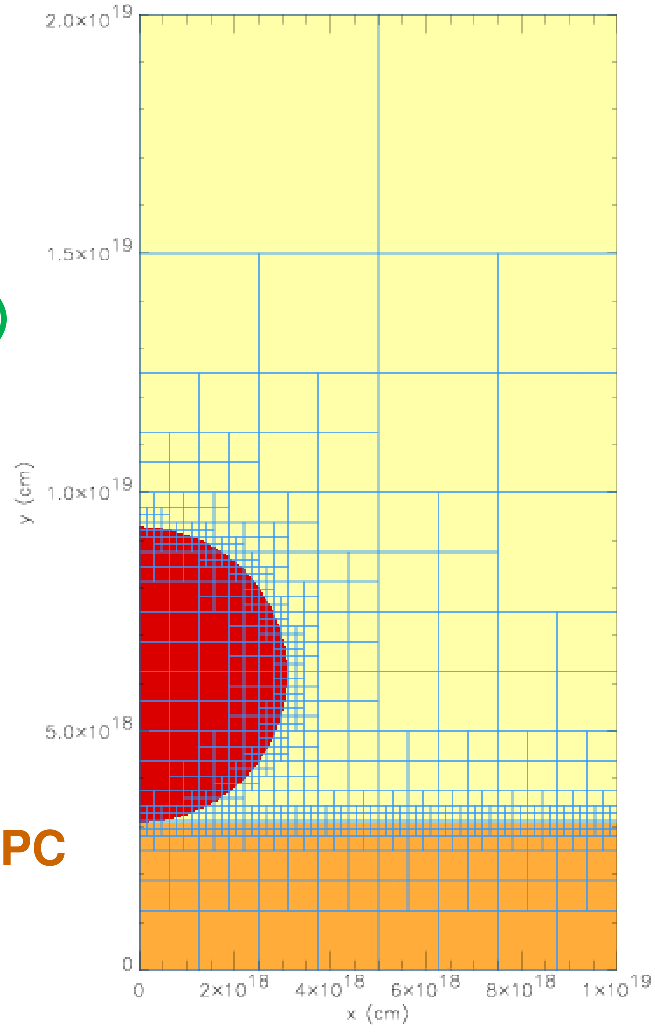
Key Requirements for HPC

- **High Performance Computing**
  (for a medium size application)
  - **N processors > 64**
  - **CPU time required > 10000 h**
    (i.e. ~7 days using 64 procs)
  - **60-100 GB of RAM**
  - **Output size ~ 100 GB**
  - **Low latency communication network**

Ex. of application: HD, MHD multi-D simulations

Ex. of platform: Mare Nostrum (BC JS21 Cluster; Barcellona Supercomputing Center)

- **In general, GRID Infrastructure not designed for HPC**

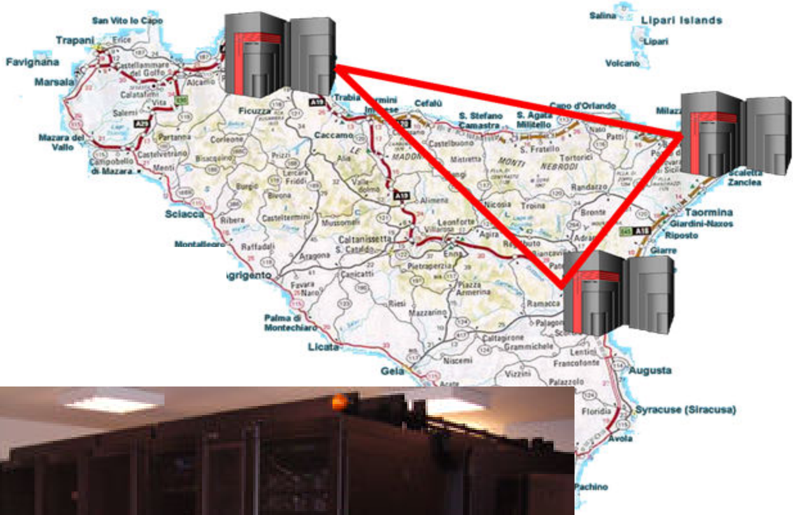


Technological challenge:
building Grid Infrastructure fully supporting HPC applications

The COMETA Consortium

INAF/OAPa among the promoters of the constitution of *COMETA* consortium and of the definition of the *PI2S2* project

Scope: Implementation and development of an e-infrastructure in Sicily based on the GRID paradigm



Facility: A number of HPC poles constituted in Palermo, Catania and Messina (May-Dec 2007)

- **Largest HPC system hosted in Palermo (616 AMD Opteron, Infiniband, 22TB disk storage)**
- **GRID infrastructure (about 2000 AMD Opteron)**

HPC Requirements

COMETA has produced a significant effort to support HPC applications in its Grid Infrastructure

- Each cluster of the infrastructure equipped with low latency communication network (InfiniBand)
- gLite 3.1 middleware extended to support MPI/MPI2
- Additional requirements:
  - Job monitoring during run → use of watchdog (*) "VisualGrid" tool
  - CPU time required > 10000 h → long term proxy: 21 days
  - run on > 64 procs → HPC queue resource reservation

(*) Watchdog utility more flexible than the Perusal file technique

MPI Deployment (1/2)

HPC requires full exploitation of WNs and communication capabilities

- HPC applications run on reserved executing nodes
- Concentration of job execution on the lowest N of physical processors (GRANULARITY)
  - New JDL TAG to select N cores of the same proc. to be used

Cooperating nodes running MPI programs tightly connected each other

- Ensure enough low latency for node-to-node communication (IB)
- Currently, MPI parallel jobs can run inside a single Computing Element (CE)

MPI: information exchange among cooperating nodes

Master node starts the processes on slave nodes

procedure based on SSH;

initial setup for the necessary key exchange

MPI Deployment (2/2)

Provide instruments to satisfy the requirements of HPC applications

patches for the LCG-2 Resource Broker; Workload Management System; User Interface; Computing Element

The patches support new tag types for MPI flavors

- **MPICH2** for MPI2
- **MVAPICH** for MPI with InfiniBand native libraries
- **MVAPICH2** for MPI2 with InfiniBand native libraries

Two scripts to be sourced before and after the exec. of MPI program

- **mpi.pre.sh**: prepare the execution environment
- **mpi.post.sh**: collect the final results

Extension of gLite3.1 middleware consists in a wrapper, able to collect the needed information from the Local Job Scheduler

Example JDL file

```
Type = "Job";
JobType = "MVAPICH2";
MPIType = “MVAPICH2_pgi706”;
NodeNumber = 128;
Executable = "flash2";
StdOutput = "mpi.out";
StdError = "mpi.err";
InputSandbox = {"watchdog.sh","mpi.pre.sh","mpi.post.sh","flash.par","flash2"};
OutputSandbox = {"mpi.err","mpi.out","watchdog.out",“flash_snr.log","amr_log"};
Requirements=(other.GlueCEUniqueId=="unipa-ce-01.pa.pi2s2.it:2119/jobmanager-lcglsf-hpc");
MyProxyServer = "grid001.ct.infn.it";
RetryCount = 3;
```

- Job wrapper copies all the files indicated in the InputSandbox on **ALL** “slave” nodes and cares about environment settings
- If some environment variables are needed ON ALL THE NODES, a **static installation** is *currently* required (middleware extension is under consideration)

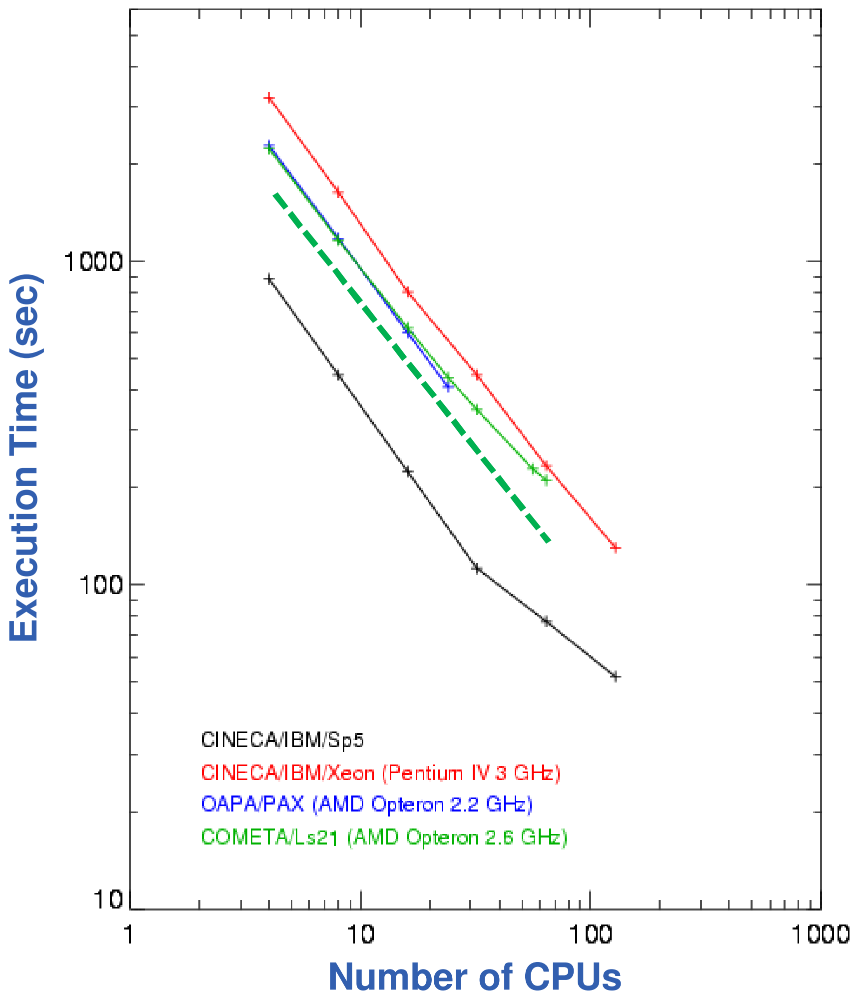
Test case: The FLASH Code

- **Framework:** Advanced Simulation and Computing (ASC) Academic Strategic Alliances Program (ASAP) Center (USA)
- **Main development site:** FLASH Center, The University of Chicago
- **Main features:** Modular, multi-D, adaptive-mesh, parallel code capable of handling general compressible flow problems in astrophysical environments
- **Collaboration OAPa/FLASH center:** to upgrade, to expand, and to apply extensively FLASH to astrophysical systems
  - New FLASH modules implemented @ OAPa (non-equilibrium ionization, Spitzer thermal conduction, Spitzer viscosity, radiative losses, etc.)

Scalability

- Problem: hydrodynamic 2-D
- Average number of grid points: 8×10^4
- Required 45 timesteps to cover 300 yr of physical time
- Tested the scalability up to 64 procs
- Parallel efficiency of
  - 80% on 32 procs.
  - 70% on 64 procs.

Working on optimization



Priorities and Future Perspectives

Making a Grid infrastructure competitive with traditional HPC facilities requires to improve the following points:

(from the point of view of a traditional HPC user)

- **STABILITY**
  - Currently, changes to system configuration may determine malfunctions of MPI applications

    → a continuous effort of system managers is necessary to support HPC applications

- **STURDINESS**
  - Currently, ~ 20% of HPC jobs fails for unknown reasons (lack of diagnostics)

- **TRANSPARENCY**
  - Improve job monitoring to allow users to check the status of the calculation *in real time*
  - Improve diagnostic capabilities in case of failure of the job