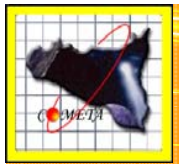
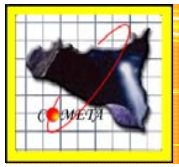
HPC Applications on the Sicilian Grid Infrastructure

Marcello Iacono-Manno (marcello.iacono@ct.infn.it)
Consorzio COMETA
OGF25 / 4th EGEE User Forum
Catania, March 5th, 2009



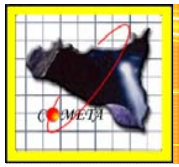
Summary

- Overview
- Sicilian Infrastructure
- Grid & HPC
- Middleware
- Scheduling Policies
- License Server
- Use Cases
  - FLUENT, FLASH, OpenFOAM, etc...



Why HPC on Grid

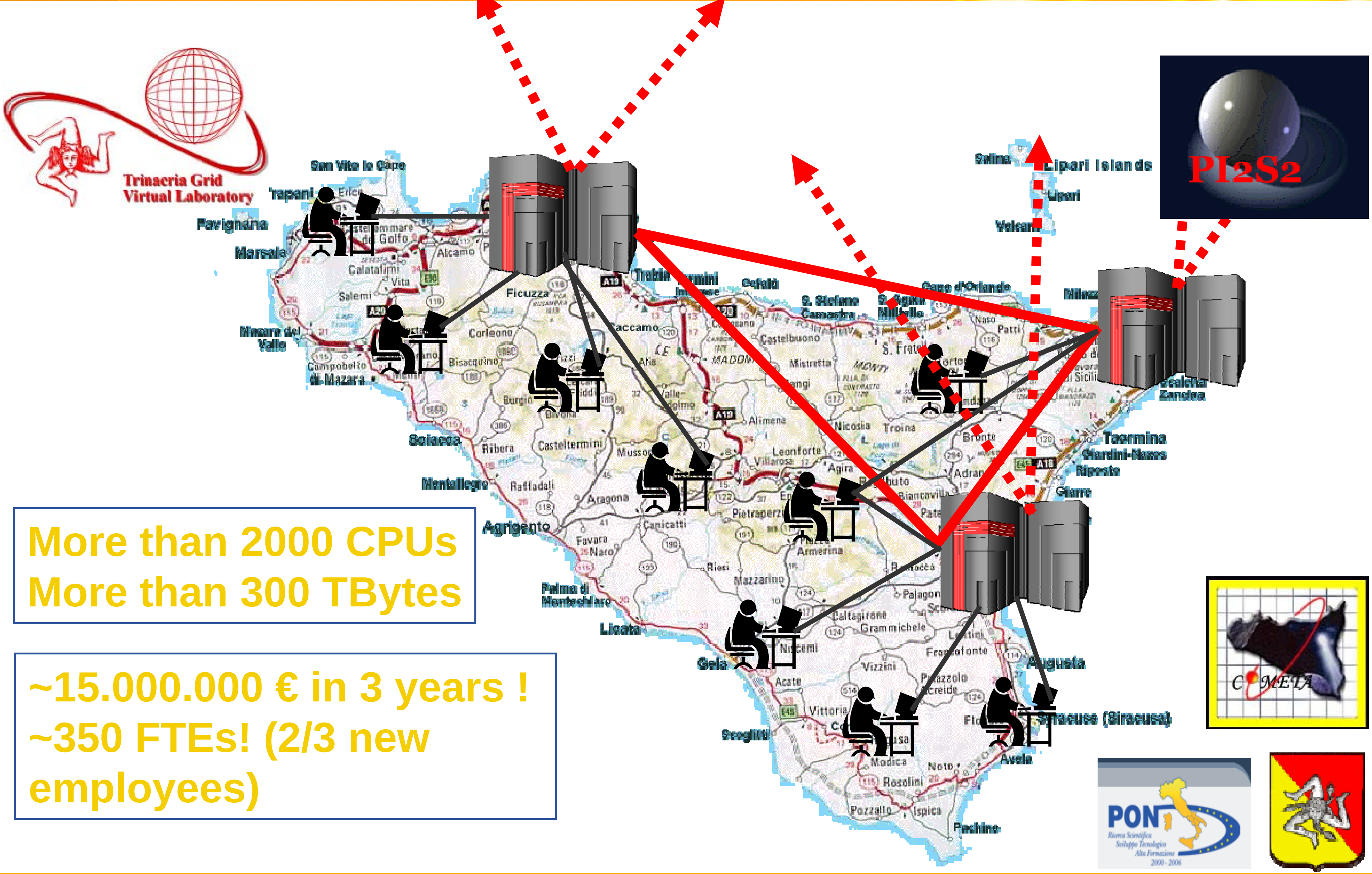
- **Provide a Grid supporting HPC is important to involve new communities**
  - Many HPC users will be happy to move to Grid and reduce the costs to achieve comparable performances
  - This is the main way towards **sustainability**
- **PI2S2 has pushed a big effort to enable its infrastructure for HPC applications**
  - Choosing infrastructure components satisfying the requirements of many HPC applications
    - es. server with multicore CPU, InfiniBand for node interconnection
  - Developing new solutions to integrate in the Grid middleware so to better manage HPC applications
  - Developing a license server to run on Grid so to allow the execution of commercial parallel applications
    - many users required commercial application such as FLUENT and others



Grid vs. HPC

- **HPC**
  - Clusters dedicated to a single user (or community)
    - One user at a time
  - Components optimised for one or a few applications
  - Ad-hoc network technologies (InfiniBand)
  - Cutting edge components to get the best performance on each single job
  - Too expensive for many small companies and/or research groups
  - Limited availability of calculus power and/or storage capacity
- **GRID**
  - Heterogeneous
    - Resources
    - Jobs
    - Users
  - Common network infrastructure (generally GigaBit Ethernet)
  - Focus on the overall infrastructure performance
    - Number of jobs
  - Allows resources sharing
    - Reduce the ownership of the infrastructure
    - Stability is more difficult to be attained
    - High overall calculus power and storage capacity available at a reasonable cost

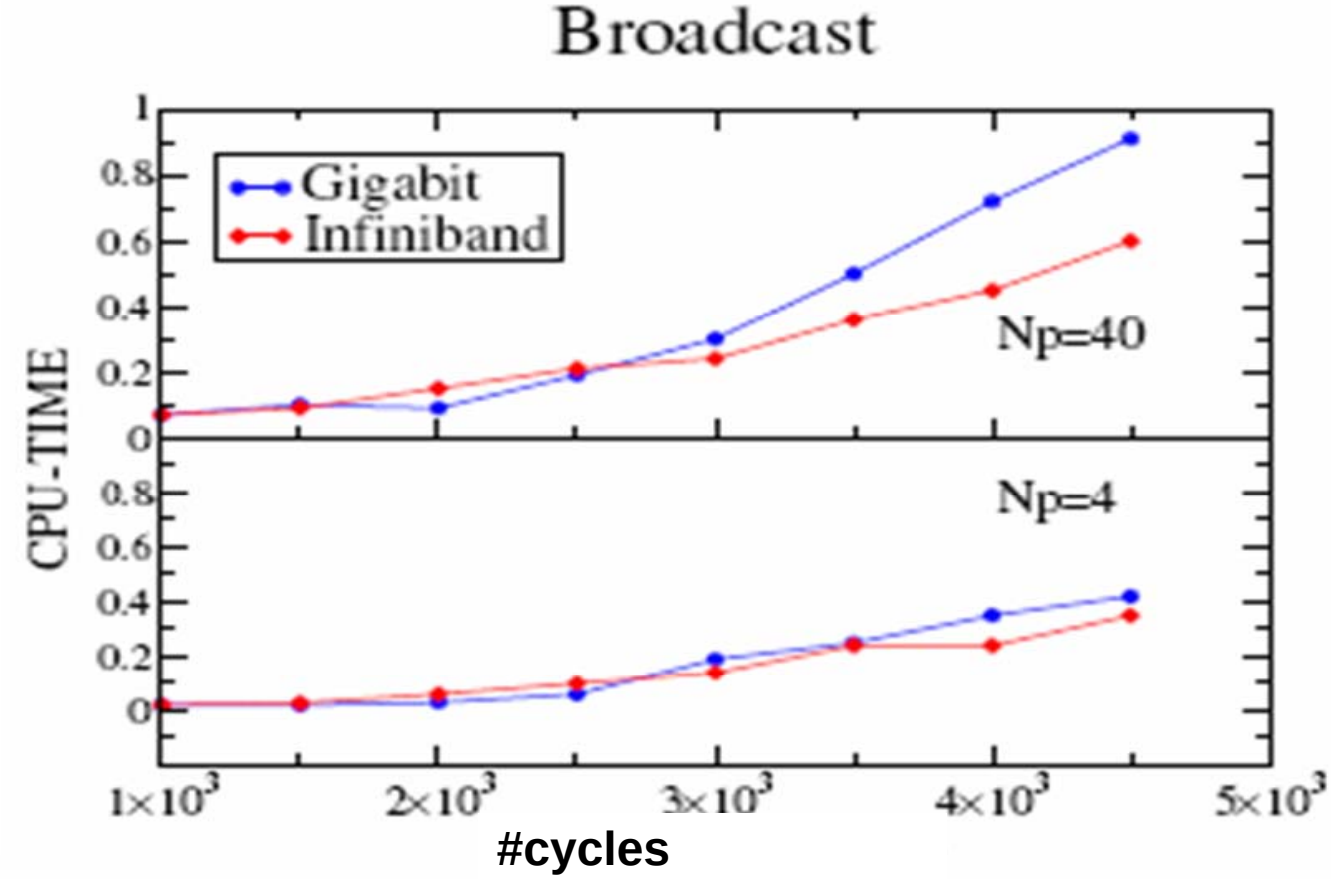
The Sicilian E-Infrastructure



More than 2000 CPUs
More than 300 TBytes

~15.000.000 € in 3 years !
~350 FTEs! (2/3 new employees)

GigaBit vs InfiniBand



The advantage of using a low – latency network becomes more evident the greater the number of nodes

MPI & Grid

- **An MPI program is an application including calls to MPI library functions that synchronise the execution flow among the cooperating nodes**
- **Official gLite supports only two MPI implementations:**
  - MPICH
  - MPICH2
- **Several patches enable MPI on “old” GigaBit Ethernet and “new” low-latency InfiniBand nets**
  - In COMETA infrastructure MPI jobs run on either GigaBit (MPICH, MPICH2) or InfiniBand (MVAPICH, MVAPICH2)
- **Currently, MPI parallel jobs can run only inside a single Computing Elements (CE)**
  - The possibility of executing parallel jobs spread on different CEs is under investigation
  - We are thinking about wiring the whole Catania Campus with IB

Submission of an MPI application

- MPI jobs are specified by setting the JDL JobType attribute to **MPICH**
- The implementation variant is specified with the attribute MPIType
  - <MPIvariant>_<compiler>, e.g. `MVAPICH2_PGI706`
- The NodeNumber defines the number of required cores
- Es.

```
Type = "Job";
JobType = "MPICH";
MPIType = "MVAPICH_gcc4";
NodeNumber = 12;
Executable = "mergesort-ib1-gcc4";
.....
```

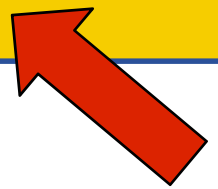
Matchmaking: the Resource Broker (RB) chooses a CE (if any!) with enough free Processing Elements (PE = CPU cores)

e.g.: free PE# ≥ NodeNumber

Matchmaking

- When these attributes are included in a JDL script the following requirements are automatically added by the WMS:

```
(other.GlueCEInfoTotalCPUs >= NodeNumber) &&
Member (<MPIType>,other.GlueHostApplicationSoftwareRunTimeEnvironment)
```

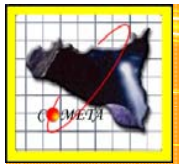


Replaced with the real value

- These requirements allow the WMS to find out the best resource where the job can be executed

Additional elements

- **Executable specifies the MPI executable**
- **Arguments specifies the WN command line**
  - Executable + Arguments form the command line on the WN
- **mpi.pre.sh is a special script file that is sourced before launching MPI executable**
- **mpi.post.sh is a special script file that is sourced after MPI executable termination**
  - warning: they run only on the master node
- **The mpirun command is issued by the middleware (… what if a proprietary script/bin?)**
- **MPIGranularity is a JDL attribute specifying the CPU core distribution**
  - *Foster the integration of shared memory applications into Grid*

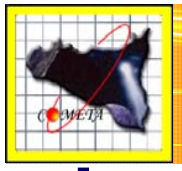


Execution

- **The WMS Job wrapper copies all the files indicated in the InputSandbox on the master and ALL of the “slave” nodes**

host based ssh authentication MUST BE well configured between all the WNs

- **If additional environment variables are needed ONLY on the “master” node, they can be set by the mpi.pre.sh**
  - If required ON ALL THE NODES a static installation is the only method (middleware extension is under consideration)
- **The WMS start the execution on the master**



NEW

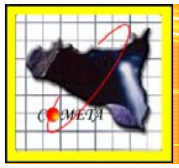
mpi.jdl

```
[
  Type = "Job";
  JobType = "MPICH";
  MPIType = "MPICH_GCC4";
  Executable = "MPIparallel_exec";
  NodeNumber = 2;
  Arguments = "arg1 arg2 arg3";
  StdOutput = "test.out";
  StdError = "test.err";
  InputSandbox = {"mpi.pre.sh","mpi.post.sh",
        "MPIparallel_exec"};
  OutputSandbox = {"test.err", "test.out",
        "executable.out"};
  Requirements =
     other.GlueCEInfoLRMSType == "PBS"
    || other.GlueCEInfoLRMSType == "LSF";
]
```

Executable

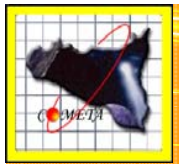
Pre e Post Processing Scripts

Local Resource Manager (LRMS) = PBS/LSF only



Scheduling Policy

- **Heterogeneous jobs running on the same infrastructure make Grids different from dedicated clusters**
  - Resources should be effectively shared among users having diverse constraint
- **PI2S2 sites have queues for different jobs duration**
  - increasing priority for shorter jobs
  - their policy is very complex due to the variety of requirements
- **Several kinds of jobs are identified**
  - emergency jobs need absolute priority so they perform pre-emption interrupting the execution of short jobs
    - emergency jobs are not very long, so pre-emption is acceptable
- **HPC job scheduling policy is based on resource reservation so they collect cores up to needed amount**
  - This policy is reasonably effective as the number of HPC jobs is far lower compared to the number of short jobs

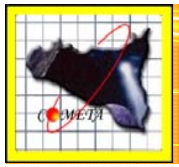


License Server

- **Many HPC programs require a software license**
- **A unique license must be shared among the users on different and geographically distant sites**
  - Currently licenses are generally connected to a user name, or a physical net address
    - Both these solution do not fit the distributed environment of Grid infrastructures
- **A float license delivered to remote sites is required**
- **Several tools allow the management of floating license**
  - FlexLM is the most used free software license server
  - The applications must know the address of the server and they ask at run-time for an execution license
- **Current license servers do not identify Grid users so it is not easily to define usage policy for commercial software**

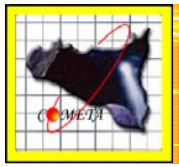
GridLM: Grid License Manager

- **GridLM has been developed by COMETA and INFN Catania to avoid problems with commercial software**
- **Execution permission is granted only for users belong to an authorised group inside a VO**
  - Users have to specify their group during the proxy creation
    - es. `voms-proxy-init --voms cometa:/cometa/example_sw`
  - Additionally, the software has to be specified in the JDL
- **The communication among sites use the Grid security mechanism**
  - Every message is encrypted with the proxy certificates
- **A further development aims to associate the license with delegated user proxy linked to a job**



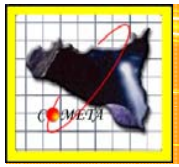
Use Case: FLUENT

- **FLUENT is a commercial software for Computer Fluid Dynamics (CFD)**
  - mainly used for for flow modelling, and heat and mass transfer simulations
- **Current distribution include the support for many MPI cluster configuration including the MPICH on InfiniBand**
- **No special effort required to execute FLUENT on Grid**
  - The MPI wrapper was by-passed since FLUENT use its own wrapper
  - The pre-processing script is responsible to run the application
  - The other JDL parameters are used to find the execution resources



Use Case: GAMESS, ABINIT

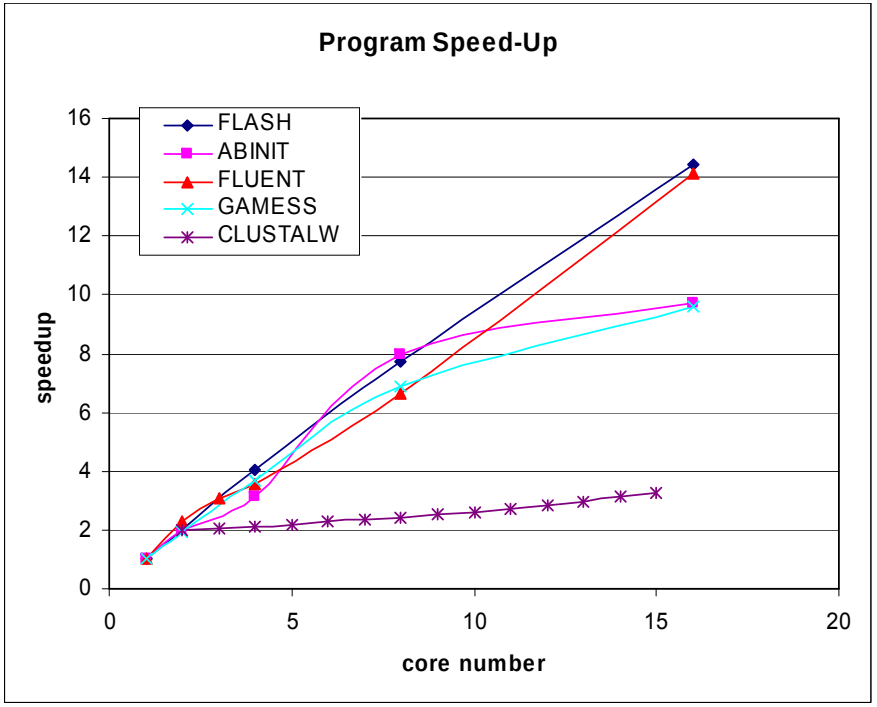
- FLASH is a 3D used in current
- Compiled
- Used Some
- Performance

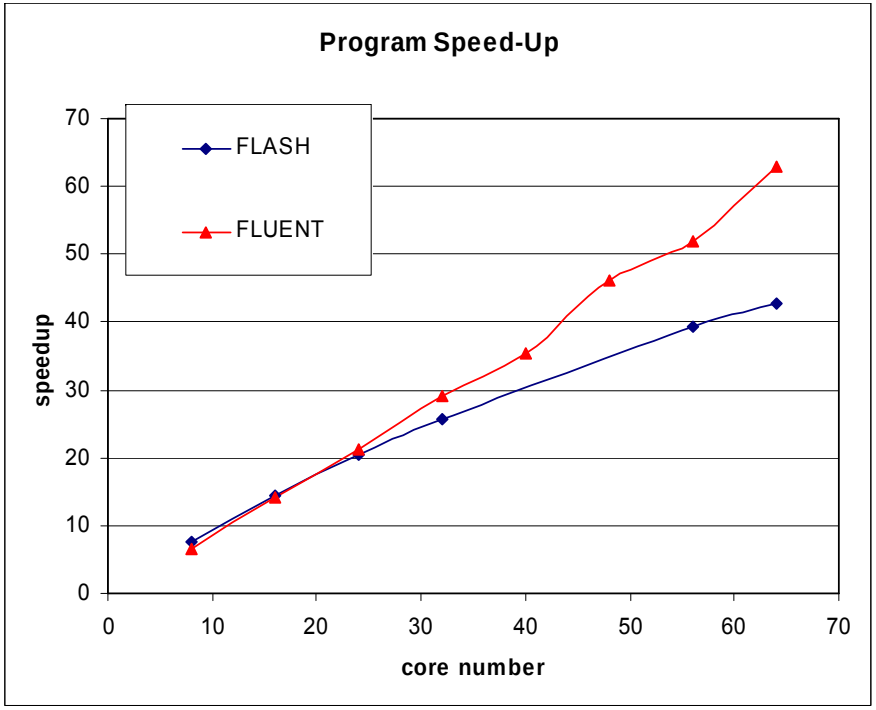


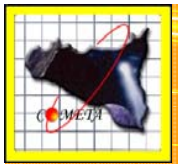
Use Case: FLASH

- **FLASH is a 3D astrophysical hydrodynamic code for supercomputers**
  - used in current astrophysical research
- **Compiled with PGI compiler and optimised for COMETA infrastructure**
- **Used in production with job using up to 128 cores and running for several days**
- **Some tests on performance enhancement after the optimisation have been executed**
- **Performance is comparable to CINECA (?)**

Speed-Up

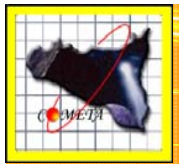

Program Speed-Up
FLASH
ABINIT
FLUENT
GAMESS
CLUSTALW
speedup
core number
0
2
4
6
8
10
12
14
16
0
5
10
15
20


Program Speed-Up
FLASH
FLUENT
speedup
core number
0
10
20
30
40
50
60
70
0
10
20
30
40
50
60
70



Use Case: OpenFOAM

- **OpenFOAM is a free simulation environment**
  - commonly used as an equation solver in solving CFD problems
- **Very complex interaction with gLite m/w**
  - a special set-up is required in any execution node but Grid mechanisms allow access only to the master
  - the MPICH support requires to recompile the code with a supported compiler
    - this is not a real limitation but need an extra effort to test the application
  - writing a solver requires to recompile the entire software so a mechanism to support new developer has to be investigated



MPI on the web...

GENERAL MPI WEB PAGES

- https://edms.cern.ch/file/454439/LCG-2-UserGuide.pdf
- http://oscinfo.osc.edu/training/
- http://www.netlib.org/mpi/index.html
- http://www-unix.mcs.anl.gov/mpi/learning.html
- http://www.ncsa.uiuc.edu/UserInfo/Training

PI2S2 RELATED MPI PAGES

- https://grid.ct.infn.it/twiki/bin/view/PI2S2/WikiConsorzioCometa

Thank you for your kind attention !

Any questions ?