Parallel execution of chemical software on EGEE Grid

M. Sterzel, T. Szepieniec, P. Lasoń
ACC CYFRONET AGH

EGEE User Forum 3,
Clermont-Ferrand, 11-14 February 2008

www.eu-egee.org

Outlook

- **Aim of this talk**
  - To demonstrate possibility of utilization of parallel versions of chemical software on EGEE Grid

- **Parts of the talk topics**
  - Commercial Software on the Grid
  - Gaussian & Turbomole
  - Parallel execution of chemical software on the Grid
  - Execution benchmarks
  - Final Remarks

Commercial Software

- **Commercial Software – why bother about it?**
  - Used by majority of scientists to conduct research
  - Faster implementation of new computational methods
  - Better user support
  - More frequent bug fixes

- **Why Gausian?**
  - Large number of computational methods implemented
  - One of the first *ab initio* codes
  - The most popular among communities
  - User friendly
  - Available on many platforms along with GUI

Turbomole – an alternative

Advantages:

- Probably the fastest B3LYP implementation
- Analytical gradients for excited states at DFT and CC2 levels
- Variety of fitting approaches speeding up calculations
- Very well scalability during parallel execution
- Extremely fast (ri)CC2 and (ri)MP2

Disadvantages:

- Limited number of DFT functionals (only “good” ones available)
- Lack of parallel version of analytical second derivatives
- Lack of parallel version of TDDFT
- Only NMR chemical shifts implemented, no Spin-Spin couplings

VOs in CYFRONET

Gaussian

- Invented and operated by ACC CYFRONET
- All license issues confirmed with Gaussian Inc,
- Open for every EGEE user
- Any computing centre with site Gaussian license may support it (4 supporting centres, another 3 in the line)
- 25+ users since the start in September 2006
- VO manager – Mariusz Sterzel (`m.sterzel@cyfronet.pl`)
- Recently enabled for parallel execution up to 8 processors

Turbomole

- Newly set-up
- Enabled for parallel execution as well

Gaussian VO – participation

As a user:

- Register at:

  `https://voms.cyf-kr.edu.pl:8443/vo/gaussian/vomrs`
- Accept Gaussian VO license requirements
- Wait for VOMRS admin acceptance
- voms-proxy-init --vo gaussian and you are ready to use the program...

Gaussian VO – participation

As a user:

- Register at:

  `https://voms.cyf-kr.edu.pl:8443/vo/gaussian/vomrs`
- Accept Gaussian VO license requirements
- Wait for VOMRS admin acceptance
- voms-proxy-init --vo gaussian and you are ready to use the program...

As a participating centre:

- Just sent an e-mail concerning participation to VO manager
- After confirmation of the license status at your centre with Gaussian Inc, detailed information concerning set-up will be sent back to you

gaussian VO Registration

[-] gaussian Registration Home
- Registration (Phase I)
- Groups and Group Roles
- Institutions & Sites
- Required Personal Info
- Certificate Authorities

Registration (Phase I)

Welcome to the gaussian VO user registration phase I page.

All fields on this page are required. After submitting this form, a confirmation email will be sent within 24 hours with further instructions. If you fail to follow the instructions within 10 days, your registration will be discarded and you will have to re-register.

If you don't receive the confirmation email, please check your email address in VOMRS and change it if necessary. If it was correct, contact ***the VO administrator*** .

Email address : jon.doe@important.domain

Grid job submission rights : full

Personal Information

First name: Jon

Last name: Doe

Phone: +00 00 123 4567

Submit Cancel

You are logged in as /C=XX/O=GRID/O=Important/CN=Jon Doe - Gaussian

/C=XX/O=GRID/CN=Important Grid CA

Grid and parallel execution

Past:

- Serial jobs only
- Job of MPICH type always enforced execution of `mpirun`

Present

- `mpirun` no longer enforced
- Instead a wrapper script can be executed which will automatically set up environment for required MPI flavour
- No possibility to request desired # of processors on a WN

Grid and parallel execution

Past:

- Serial jobs only
- Job of MPICH type always enforced execution of `mpirun`

Present

- `mpirun` no longer enforced
- Instead a wrapper script can be executed which will automatically set up environment for required MPI flavour
- No possibility to request desired # of processors on a WN

… Unfortunately not all sites are set up…

Chemical software

- **“Old codes” – mostly written in FORTRAN**
- **Serial – parallel execution added later**
  (with exceptions)
- **Different parallelization models used**
- **Low scalability in many cases**
- **Only selected computational methods parallelized**

Chemical software

- **“Old codes” – mostly written in FORTRAN**
- **Serial – parallel execution added later**
  (with exceptions)
- **Different parallelization models used**
- **Low scalability in many cases**
- **Only selected computational methods parallelized**

All that makes parallel grid ports of chemical software even more complicated

Selected cases

- **Gaussian**
  - Parallelization via OpenMP or Linda
  - OpenMP – SMP machines or multiprocessor/core clusters up to # of processors/cores on WN
  - Linda – allows the parallel execution between nodes. Requires equal # of processors for each WN
  - For Linda additional expenses required (commercial package), available only for specific platforms

- **Turbomole**
  - Uses MPI – currently HPMPI
  - No specific requirements

Selected cases

- **NAMD**
  - Charmm model (`charmrun`), requires an information about WNs in specific format
- **Gamess**
  - Uses DDI but execution via MPI also possible
- **ADF**
  - Uses MPI (MPICH, OpenMPI, HPMPI, …)
  - One of the best parallelized QC codes // to my knowledge ;-)

Selected cases

- **NAMD**
  - Charmm model (`charmrun`), requires an information about WNs in specific format
- **Gamess**
  - Uses DDI but execution via MPI also possible
- **ADF**
  - Uses MPI (MPICH, OpenMPI, HPMPI, …)
  - One of the best parallelized QC codes // to my knowledge ;-)

In addition above mentioned packages require a shared directory to store file(s), password-less connection between nodes, etc.

Grid solution

- **Gaussian**
  - Parallel execution via OpenMP on a single WN up to # of processors/cores available on that worker node
  - Necessarily queue system set-up requires a Site admin help
  - Torque set-up:
    - Modification of `/var/spool/pbs/torque.cfg`

      to: SUBMITFILTER /var/spool/pbs/submit_filter.pl
  - Other settings -- typical
    - Job has to be of MPICH type
    - # of processors controlled via `NodeNumber` variable
    - Gaussian `%Nproc` route is automatically set-up by script executing Gaussian
  - Execution with 8 processors per job possible.

Sample script

```
Executable    = "/bin/sh";
Arguments     = "$GAUSSIAN_SW_DIR/gaussian.run myfile.com";
JobType       = "MPICH";
NodeNumber    = 8;
InputSandbox  = {"myfile.com"};
StdOut        = "myfile.out";
StdErr        = "myfile.err";
OutputSandbox = {"myfile.log", "myfile.chk",
                 "myfile.out", "myfile.err"};
Requirements  = other.GlueCEUniqueID=="ce.cyf-kr.edu.pl"
```

Grid Solution cont.

- **Turbomole**
  - No special set up except shared directory needed, # of processors automatically discovered by Turbomole scripts
- **NAMD**
  - Similar to Turbomole. If the NAMD executing script was set-up properly during installation the necessarily “node file” is created every time program is executed
- **GAMESS**
  - Depends on Grid port
  - In case of MPI no additional input needed
  - DDI case – may require WN reconfiguration especially if large DDI memory is requested by a job

Execution benchmarks

Scheduling time

- MPI jobs
  - 4 proc./job -- usually less than hour
  - 8 proc./job -- waiting time even 3-4 hour
- OpenMP jobs
  - Job waiting time much longer, heavily depends on site overload
    - *4 proc./job -- from less than hour up to 6 hours*
    - *8 proc./job -- in some cases job waiting time exceeds 12 hours*

Parallel job execution can be inefficient in case of short (less than 24 h) jobs

Summary

- A parallel execution of chemical software can be made as simple for the user as a serial execution is now
- It is always possible to find solution for parallel execution even if computational platform does not directly support certain execution model
- Parallel execution of OpenMP jobs can be inefficient in a case of short time jobs