HYP3D

Gilles Bourhis

Equipe SIMPA, laboratoire PALMS

Univ. Rennes 1 – CNRS

http://www.palms.univ-rennes1.fr/SIMPA

Presentation

- **Short description of the HYP3D code**
  - Application used in molecular physics, based on quantum mechanics theorie
  - Study reactive collisions
    - example : O + H2 --> OH + H
  - It uses a time-independant method based on hyperspherical coordinate to study the dynamic of the reaction
- **Author**
  - Jean-Michel Launay, head of the SIMPA team
- **History and usage**
  - Developed in 1990
  - Used mainly by local users
  - Already used to study 15 reactions (40-50 published articles)

Presentation

- **Type and structure of the application**
  - Sequential Fortran 77 code
  - A lot of linear algebra, matrix operations
  - Uses some local libraries and the BLAS/LAPACK libraries.
  - Has been ported to different kinds of Unix/Linux machines
  - Currently used on MacOS X (Xserve cluster) with the Intel Fortran compiler and the MKL package.
  - The application is a set of three binaries (TB, TJ, TK)
- **Special requirements**
  - Large in-core memory (up to 8 GB)
  - Large datasets (up to 8 GB) and storage (2TB)
  - Typical length of an execution : 48 hours

Workflow

- **Structure of the application**
  - TB -> TJ -> TK
  - TB runs only once to produce a "basis" file
  - TJ runs several times, one for each angular momentum (the J parameter, starting at 0, can iterate to 30, sometimes 100)
    - Takes the basis file and the value of J as input
    - Produces a J-Basis file
  - After each execution of TJ, TK is run once. For a given J, TK iterates internaly on the energy values (up to 1000).
    - Takes the J-Basis file and the set of energy values as input
    - Produces a K-matrix file, a K matrix for each energy
      - *the K-matrix is used to compute the scattering matrix (S-matrix) from which cross sections and rates coefficients can be computed*

Gridification

- **Why gridification ?**
  - Possibility to increase the code performance (parallel execution of some part)
    - TK can be parallelized on the energies (MPI)
    - several instances of TJ+TK can be executed in parallel with different angular momentum values (J values)
  - Making the code available to the growing community interested in reactive processes
  - Help progress of knowledge of binary collision dynamics in particular at thermal, low and ultra-low temperature

Porting session

- **Test case for porting : NA3**
  - Small case : runs with 1GB of memory and files of 150 MB, execution time (for J=0, only 10 energy values) about 30'
  - Requirement : a Fortran compiler and BLAS/LAPACK
- **What can be done**
  1. Compile our local libraries
  2. Compile and link the three binaries
  3. Decide what to do with the data files (use of the file management system)
  4. Change the 3 input parameter files, write the 3 JDL files and the 3 scripts jobs OR define a workflow in P-GRADE?
  5. Submit the jobs (can be interleaved with the previous step) ... and look at the results