Code *TRIO-U*
Unsteady 3D code for general CFD purposes

02 / 2006

Prepared by
Benoit MATHIEU
CEA/DEN/DER/SSTH/LMDL

Protocol for tests

- Date of tests : 10-2005 → 03-2006
- Centre : ANL (1024 nodes ; ie 2048 cores)
  - Source code provided
  - Source transferred to ANL
  - Distant compilation and tests
  - Deletion of source code on ANL machine

Compilation

- Trio_U compiled without any code modification
- Numerical Platon not available (RPC not supported)

Job execution

- Occasional machine failure, often due to file-system problems (hardware and software), machine was otherwise very responsive (interactive tests up to 128 nodes)
- Hardware architecture does not support debugging
- No virtual memory and only 512 MB of memory per node

Computation

- 350.000 mesh elements per node (minimize communication/computation ratio)
- Structured mesh, incompressible 3D Navier-Stokes (implicit pressure solver)

Scaling tests

"Weak scaling" test

- problem size increased like the number of processors
  - maximum size treated : 700^3 (343 millions) pour 2048 cores
- parallel efficiency constant for any number of nodes between 16 and 512 (estimated figures: the test case does not fit in memory on a single cpu)
  - 55% (virtual node mode = two processors per node are used)
  - 65% (coprocessor mode = only one processor per node is used).

  (55% means that with N processors, the result is obtained 0.55*N times faster than with one processor. Speedup is 55 with 100 processors or 550 with 1000 processors)

"Strong scaling" test

- constant problem size, increasing processor number
- moderate performance due to communication overhead

Machine performance analysis

Performance analysis:

- Rather low network bandwidth (measured 130MB/s/processor), but very good scalability

- Parallel efficiency of 55% (dual core) or 65% (single core) is due to:

  - 10 - 20% of time spent in communication in optimal computation (maximize number of mesh elements per processor) => 20% less efficiency

  - parallel implicit solver is not perfectly scalable => 15% less efficiency

  - memory bandwidth shared between two processors in dual core mode => 10% less efficiency in “virtual node” mode.

- Compiler does probably not produce optimal code: sequential code runs two times slower than on a Pentium 4 2Ghz processor. The code could probably run two times faster on BGL with appropriate optimization.

Conclusions:

- If code is well optimized, network bandwidth will become a limiting factor. Using BGL optimally implies careful code and communication optimization.
- Simplified machine architecture: no debug, no complex code (no RPC, …)
- Network bandwidth and cpu speed are well balanced.
- Memory per processor (256MB in dual core operation) is quite low for complex codes (Trio_U executable alone uses 50MB).