Fluent Parallel Performance on Multi-Core System

Wang Junhong

Objective

This benchmark work studies the performance of the parallel Fluent solver on the new multiple core system. Fluent solver was parallelised using the MPI method for communication and message exchange among processor units.

Result and Findings

The benchmark testings were conducted on one thread, and two, four and up to eight threads in parallel. In order to compare the performance between threads on a single node and threads cross nodes, the parallel computations of the Fluent solver were performed on (i) a dual quad-core system, (ii) multi-core systems with two threads on each, and (iii) multi-core systems with one thread on each. Fig. 1 illustrates the three scenarios.

![Fig. 1 Execution scenarios of a four-thread parallel Fluent solver](image-url)
The clock speed for the multi-core system is 2.66 GHz. The testing case was a Fluent model with half a million cells.

Fig. 2 illustrates the benchmarking result. The figure shows that the parallel computing performance of the Fluent solver on two threads, either within one system or cross system, could reach about 1.9X. When running on four threads, the speedup on four threads within the quad-core system was about 3.3X, which was much better than the approximate 2.7X to 3X speedup rate obtained cross system.

![Fig. 2 Speedup Rate Comparison on Multi-Core System](image)

However, when running the solver on eight threads within the dual quad-core system, there was no gain of speedup, possibly due to the high overhead of data traffic among all eight threads inside the system. On the other hand, the speedup rate increased to 4.2X by running the solver cross four or eight systems. The benchmark testing was also extended to run the solver on 16 threads across 16 systems, and the speedup rate again increased to 6.2X, a rather small improvement in the performance.

It should be noted that the problem size and the interconnection among the cluster nodes are very crucial when using large number of systems for parallel computing. Large problems, which cannot be handled within a single system due to the memory limit, can be solved by running them cross multiple systems.

The performance of the new multi-core system was also compared with Atlas2 cluster, the latest parallel computing cluster in SVU with a CPU clock speed of 3.6 GHz. A single thread job easily ran 1.2X faster on the new system, though the multi-core clock speed was 2.66 GHz.
Summary

The benchmark results indicate that:

1. Fluent solver won’t scale well running with more than four threads within a single node
2. If there is a need to run more than 4 threads, it is recommended to run it across nodes
3. the parallel performance depends very much on the application, job size and interconnection among the cluster
4. the single core speed is 20% faster than that for the Atlas2 xeon EM64T processor

We recommend that users run the Fluent job submission command “fluent-job” to submit their parallel Fluent jobs. The command will take into account various issues, such as the performance characters of the new system, the system load and the queue status, and then suggest the most optimal LSF batch queue to run your Fluent simulation job.