Performance Evaluation and Prediction on a Clustered SMP System for Aerospace CFD Applications with Hybrid Paradigm

Yuichi Matsuo*[1], Naoki Sueyasu[2] and Tomohide Inari[2]

[1] Japan Aerospace Exploration Agency, JAXA's Engineering Digital Innovation Center
7-44-1 Jindaijihigashi, Chofu-shi, Tokyo 182-8522, Japan
E-mail:matsuo.yuichi@jaxa.jp - Web page: http://censs.chofu.jaxa.jp

[2] Fujitsu Limited, Makuhari Systems Laboratory
1-9-3, Nakase, Mihama-ku, Chiba-shi, Chiba 261-8588, Japan

Key Words: Performance evaluation and prediction, Aerospace CFD, Hybrid paradigm, SMP

ABSTRACT

Japan Aerospace Exploration Agency has introduced a new terascale clusterd SMP system as a main compute engine of Numerical Simulator III for aerospace science and engineering research purposes. The system is using Fujitsu PRIMEPOWER HPC2500; it has computing capability of 9.3Tflop/s peak performance and 3.6TB of user memory, with about 1,800 scalar processors for computation. In this paper, we first present the performance evaluation results for aerospace CFD applications with hybrid programming paradigm used at JAXA. Next we propose a performance prediction formula for hybrid codes based on a simple extension of Amhhal's law, and discuss about the predicted and measured performances for some typical hybrid CFD codes.

1. Introduction

Remarkable developments in microprocessor technology are pushing up computing power everyday. Parallel processing is becoming inevitable for large-scale numerical simulations such as used in Computational Fluid Dynamics (CFD) in aerospace. Meanwhile, shared-memory architecture is becoming widespread because of its programming ease.

From 1993 through the third quarter of 2002, the Japan Aerospace Exploration Agency (JAXA), formerly known as the National Aerospace Laboratory of Japan (NAL), operated a distributed-memory parallel supercomputer system called *Numerical Wind Tunnel* (NWT) [1] that consists of 166 vector processors with 280Gflop/s of peak performance. In October 2002, it was replaced with a large clustered SMP system(i.e., a distributed parallel system consisting of SMP nodes) with approximately 1,800 scalar processors, peak performance of 9.3Tflop/s, and 3.6TB of user memory. The new system is called *Numerical Simulator III* (NS-III).

In this paper, we first show the results of performance evaluation for our current parallel aerospace CFD applications. Next, we discuss about the performance prediction for the parallel CFD applications on the clustered SMP system by proposing a simple prediction formarism besed on Amdhal's law.

2 System Overview

The computing subsystem in NS-III is called *Central Numerical Simulation System* (CeNSS). We have 18 cabinets; each is Fujitsu PRIMEPOWER HPC2500 [2], where a cabinet is the physical unit of hardware. Each cabinet has 128 CPUs with 256GB of shared memory and can act as a 128-way symmetric-multi-processor (SMP) system in its maximum configuration. The CPU is the SPARC64 V scalar chip with 1.3GHz clock. Thus, the theoretical peak performance

per CPU becomes 5.2Gflop/s and 665.6Gflop/s per cabinet. L2 cache is 2MB on chip. For computation, 14 cabinets are dedicated, giving a total peak computing performance of 9.3Tflop/s and 3.6TB of memory. A cabinet can be partitioned into either 2 or 4 nodes according to need, where a node is the logical unit from the operating system's point of view. In the CeNSS, each compute cabinet is partitioned into 4 nodes where each node is a 32-way SMP with 64GB shared memory, giving a total of 56 compute nodes. For the remaining cabinets, 3 are partitioned into 2 nodes each, for a total of six 64-way SMP nodes and used for service, login and I/O purposes. All nodes are connected to a crossbar interconnect network with high 4GB/s bi-directional bandwidth and low 5μsec latency through one data transfer unit per node.

The programming environment is crucial for any parallel system. In principle, we adopt the so-called hybrid programming paradigm; that is, we use the 'thread parallel' model within a node, with automatic parallelism or OpenMP, whereas between nodes, we use the 'process parallel' model with MPI or XPFortran, as shown in Fig. 1. Codes that use 'process parallel' only can also be run. Since we have already been accustomed to our own data parallel language NWT-Fortran (similar to HPF), we use its successor XPFortran. This programming style is very convenient for us, because if we think one NWT processor is mapped onto a node of CeNSS, the transition in programming from NWT to CeNSS is quite natural, particularly in parallel coding. For more details on the system description, see Ref.[3].

3 Performance evaluation for JAXA aerospace CFD applications on the CeNSS

We firstly measured the performance of several parallel CFD applications written with the hybrid programming paradigm actually being used at JAXA. As indicated in Table 1, six parallel CFD applications with different memory-access/data-transfer features are chosen. In Fig 2, each code's characteristics are plotted schematically where the horizontal line corresponds to the memory access ratio and the vertical line to the data transfer ratio. Linpack and NPB (NAS Parallel Benchmark) are also plotted for reference. Here, all the values are from the same job configuration (process*thread). As shown in Fig. 2, the selected codes can be classfied into the three types, that is, Type 1; CPU intensive with moderate memory access and light communication but with large amounts of floating point operations, Type 2; data transfer intensive with moderate memory, and Type 3; with heavy memory access. For more details on the applications, see Ref. [4].

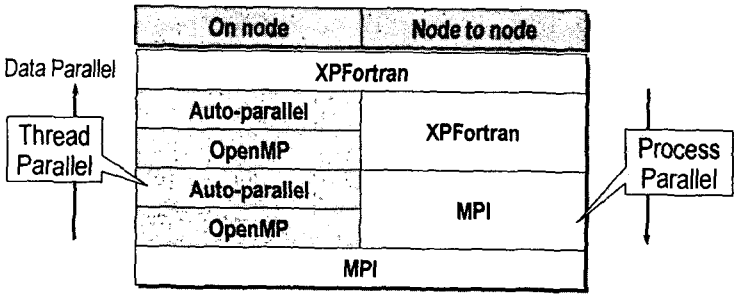

Figure 1: Programming paradigm in CeNSS.

Table 1: Specifications of JAXA parallel CFD applications.

Code: Name	Application	Simulation model	Numerical method with features	Parallel strategy	Language
P1:LES	Aircraft component	LES	FDM	OpenMP + MPI	F77
P2:HJET	Combustion	DNS	FDM with chemistry	OpenMP + MPI	F77
P3:CHANL	Turbulence	DNS	FDM with FFT	OpenMP + XPF	F77
P4:HELI	Helicopter	URANS	FDM with overlapped mesh	Auto-parallel + XPF	F77
P5:UPACS	Aeronautics	RANS	FVM with multi-block mesh	OpenMP + MPI	F90
P6:JTAS	Aeronautics	RANS	FVM with unstructred mesh	OpenMP + MPI	F77

LES: Large-Eddy Simulation, DNS: Direct Numerical Simulation, URANS: Unsteady RANS, RANS: Reynolds-Averaged Navier-Stokes, FDM: Finite Difference Method, FVM: Finite Volume Method

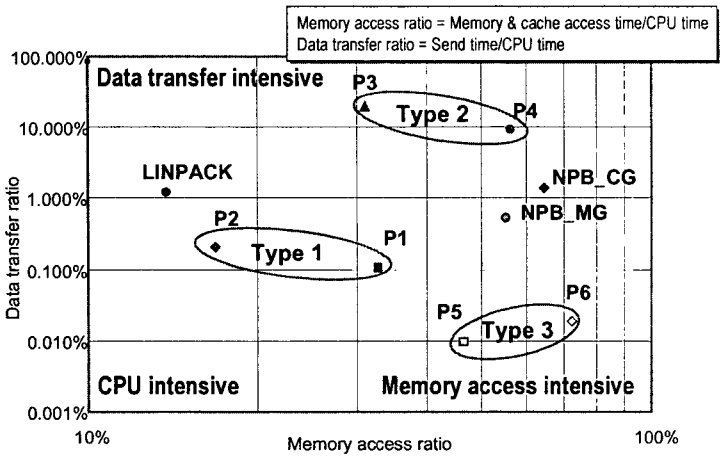

Figure 2: Features of JAXA hybrid CFD codes.

The measured speed-up performances for codes P3 and P5 are shown as examples in Fig. 3. At code P3, a large grid of 2048*448*1536 is used, and it can be seen that the process scalability tends to saturate for larger number of CPUs. At code P5, meanwhile, a 40*20*80*512 grid is used, and the thread scalability tends to saturate. To see the hybrid performances, the speed-up ratios are plotted in Fig. 4 with the number of CPUs constant where (*,*) means (process, thread). We found that in code P3 even with the same number of CPUs, the performance is better for the job with 4 threads by 56 processes than that with 1 thread by 224 processes. This is a typical example on the performance of the hybrid model while generally pure MPI is faster than hybrid [5] as in Fig. 4(b). For more details on the benchmark results, see Ref. [4].

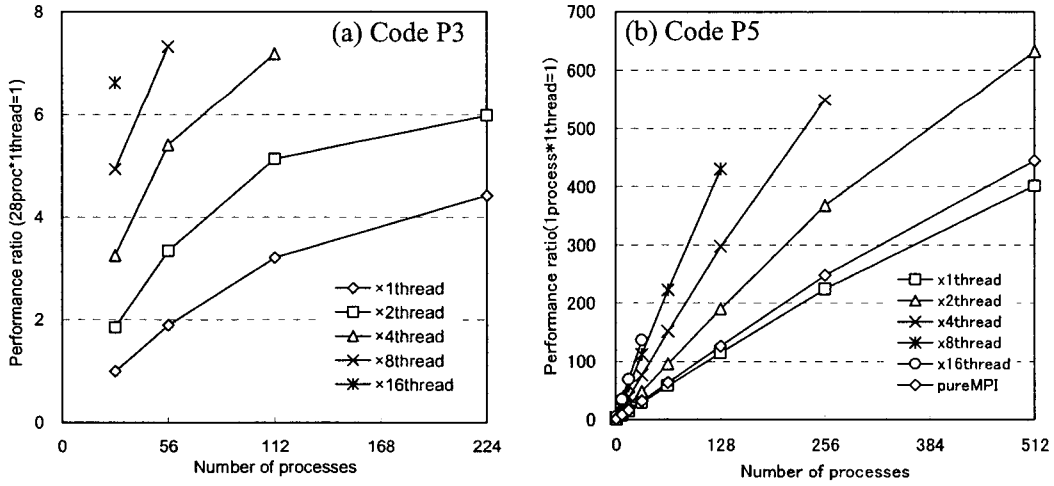

Figure 3: Speed-up performance for codes P3 and P5.

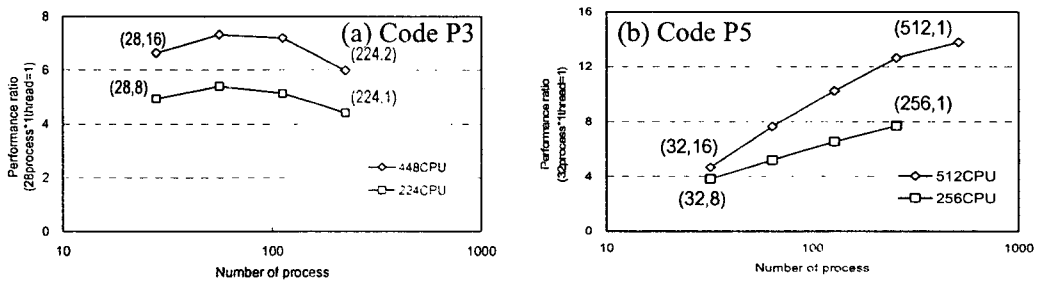

Figure 4: Speed-up performance for codes P3 and P5 shown with constant CPUs.

4 Performance prediction for the JAXA CFD applications with hybrid programming

It is well known that the speed-up performance of a parallel code is predicted by Amdhal's law, saying that the speed-up ratio is written by $S(n) = T_{serial}/T_{parallel}$, with $T_{parallel} = T_{serial} \times \{(1 - a) + a/n\}$...(1) where $T_{serial/parallel}$ is serial/parallel CPU time, a parallel ratio, and n number of CPUs. For the hybrid parallel programming paradigm, the Amdhal's law can be extended with a straightforward manner and the speed-up is written by $S(n_p,n_t) = T_{serial}/T_{hybrid}$, with $T_{hybrid} = T_{serial} \times \{(1 - a_p) + a_p/n_p\} \times \{(1 - a_t) + a_t/n_t\}$...(2) where $T_{serial/hybrid}$ is serial/hybrid CPU time, a_p process parallel ratio, n_p number of processes, a_t thread parallel ratio, and n_t number of threads. Also we find that when the communication volume is not negligible, the speed up can be written $S(n_p,n_t) = T_{serial}/T_{hybrid}$, with $T_{hybrid} = T_{serial} \times [\{(1 - a_p - c_t - c_n) + a_p/n_p\} \times \{(1 - a_t) + a_t/n_t\} + (c_t + n_p \times c_n)]$...(3) where $T_{serial/hybrid}$ is serial/hybrid CPU time, a_p process parallel ratio, n_p number of processes, a_t thread parallel ratio, n_t number of threads, c_t ratio of communication independent to the number of processes , and c_n ratio of communication proportional to the number of processes. In Fig. 5, the predicted and measured performances are compared for codes P3 and P5 with the number of CPUs constant. We found that for code P5 the performance can be predicted well with Eq. (2) while for code P3 the performance can be predicted well with Eq. (3).

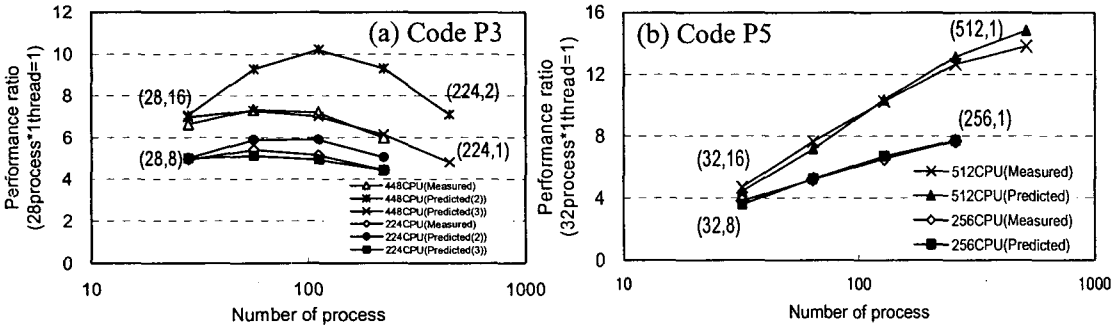

Figure 5: Speed-up performance for codes P3 and P5 shown with constant CPUs.

5 Summary

In this paper, we discussed about the performance evaluation and prediction results for JAXA hybrid CFD applications by selecting some typical codes. We found that the benefit of hybrid programming was visible for the application with large communication cost, e.g. code P3, and also found that by using the extended Amdhal's law the speed-up performances of the hybrid CFD codes can be predicted well even when the communication cost is high. This shows that the best process*thread combination of a code could be determined without a close benchmark if the memory-access and communication cost is known. But the present formalism is based on empiricism so we plan to confirm it theoretically.

References

[1] Miyoshi, H., et al., "Development and Achievement of NAL Numerical Wind Tunnel (NWT) for CFD Computations," In *Proceedings of SC1994,* November 1994.

[2] Fujitsu Limited, http://www.fujitsu.com.

[3] Matsuo, Y., et al., "Numerical Simulator III – Building a Terascale Distributed Parallel Computing Environment for Aerospace Science and Engineering," *Parallel Computational Fluid Dynamics, New Frontiers and Multi-Disciplinary Applications*, Elsevier (2003), 187-194.

[4] Matsuo, Y., et al., "Early Experience with Aerospace CFD at JAXA on the Fujitsu PRIMEPOWER HPC2500," In *Proceedings of SC2004,* November 2004.

[5] Cappelo, F. and Etiemble, D.,"MPI versus MPI+OpenMP on IBM SP for the NAS Benchmarks," In *Proceedings of SC2000,* November 2000.