Performance of OVERFLOW-D Applications based on Hybrid and MPI Paradigms on IBM Power4 System

M. Jahed Djomehri
Computer Sciences Corporation
NASA Ames Research Center, Moffett Field, CA 94035
djomehri@nas.nasa.gov

Abstract

This report briefly discusses our preliminary performance experiments with parallel versions of OVERFLOW-D applications. These applications are based on MPI and hybrid paradigms on the IBM Power4 system here at the NAS Division. This work is part of an effort to determine the suitability of the system and its parallel libraries (MPI/OpenMP) for specific scientific computing objectives.

1 Introduction

The IBM Power4 system at the NAS Division is composed of two 32-way symmetric multiprocessors (SMPs), with 32 GB each of central memory. Each SMP is contained within a single cabinet. This system is temporarily installed at NAS for a preliminary assessment of its suitability for high performance scientific computing. Specifically here, we describe the performance of NASA's overset grid CFD application, OVERFLOW-D [2], on the IBM Power4 testbed. OVERFLOW-D has been specialized for moving-body (dynamic) grid applications, and is based on a version of the NASA aerodynamic flow solver, OVERFLOW [3], developed mainly for static overset grid systems. Two parallel versions of OVERFLOW-D have been considered for our experiments. One is based on the MPI paradigm, referred to here as "overd-mpi" [4], and the second is based on the hybrid (MPI+OpenMP) paradigm, referred to here as "Overd-hybrid" [1]. Both versions have already been tested on an SGI O2K platform.

2 Performance Results

The test case used with the above applications has a grid system of about 8 million grid points with 41 curvilinear grid blocks covering the flow domain.

Several runs have been made with both overd-mpi and overd-hybrid applications on the IBM P4 system using this same test case. The performance data below has been reported over a period of 20 time-steps. At the time of our experiments one cabinet of the test bed, named ibm02, retained its original configuration, (i.e. one 32-way node), but the second cabinet, named ibm01, was reconfigured as four 8-way nodes. All four nodes of ibm01, named ibm1-0{1,2,3,4}sa, were coupled with "Colony Switches". Ibm02 and ibm01 were coupled over a gigabit ethernet.

Our performance experiments on the ibm test-bed were conducted interactively, and would fluctuate somewhat depending upon the concurrent usage of the system by other users. To minimize these effects, We have frequently monitored the system occupancy via the program, "topas", and in some cases have had to repeat the runs several times. The best data has been reported here. Nevertheless, because of the time constraint during these runs, a possible small margin of correction should be kept in mind.

2.1 MPI Application

The overd-mpi application was used for the experiments here. The code was compiled in 64-bit mode with following options:

- `F77 = mpxlf_r`
- `CC = xlc_r`
- `LINK = ld`
- `FFLAGS = -O3 -g -q64 -qhot -qnosave -qtune=pwr4 -qcache=auto`
- `CFLAGS = -O -g -q64`

Table 1 shows performance results of overd-mpi on IBM P4. The execution runtime (in seconds per time-step), denoted by T_{exec}, consists of computation and communication timings and are averaged over the total number of MPI processes, N_{MPI} used. Runtimes are reported on ibm02 and ibm01. Some runs are split across the two systems,indicated by ibm02+ibm01, and some are split across the nodes of ibm01. All the split jobs are characterized by the number of nodes used; the number of MPI processes are split equally between the nodes. The total number of processors, N_{procs}, is equal to N_{MPI}. Whenever possible, the runtimes on IBM P4 are compared with similar runs on the tightly coupled SGI O2K machine. Results for O2K runs are taken from table 2 in reference [1].

As seen on the table, for the same number of processors, runtimes on the single node of ibm02 are shorter (i.e. more efficient) than similar runs on the multiple nodes of ibm01 and/or ibm02+ibm01 combined. The difference in the performances is mainly a reflection of the latency and communication time across processors. The inter-processors communication time on ibm02 is shorter due to the stronger interconnection, as compared with the communication time through the "colony switches" used in ibm01 or through the gigabit ethernet

Table 1: Comparison of OVERFLOW-D runtimes (in seconds) on IBM P4 and SGI O2K based on MPI programming model, using 8 million grid point test case

MPI Application					
		IBM P4			SGI O2K
Total N_{Procs}	N_{MPI}	Machine	N_{nodes}	T_{exec}	T_{exec}
2	2	ibm02	1	15.0	-[a]
.	.	ibm01	2	15.8	-
.	.	ibm02+ibm01	2	16.3	-
4	4	ibm02	1	8.5	31.4
.	.	ibm01	4	9.3	-
.	.	ibm02+ibm01	2	10.0	-
8	8	ibm02	1	4.3	15.4
.	.	ibm01	4	4.8	-
.	.	ibm02+ibm01	2	6.1	-
16	16	ibm02	1	3.7	9.0
.	.	ibm01	4	5.5	-
.	.	ibm02+ibm01	2	4.2	-
32	32	ibm02	1	3.4	5.3
.	.	ibm01	4	3.8	-
.	.	ibm02+ibm01	2	4.5	-

[a]Dash "-" denotes"Not Applicable", or data was not available.

between ibm02 and ibm01. Runs on ibm02 are 2.5 to 3.5 times faster than the runs on the O2K for 2 to 16 processors, but only about 1.5 times faster with 32 processors. It should be noted that the latter runs suffer significantly from poor load balancing caused by assigning 41 grid blocks onto 32 MPI processes. The parallel scalablity on O2k is slightly better than on the IBM P4.

2.2 Hybrid Application

The overd-hybrid application was used for the following experiments. This code was similarly compiled in a 64-bit mode using the following compiler options:

- `F77 = mpxlf_r`
- `CC = xlc_r`
- `LINK = mpxlf_r -q64`
- `FFLAGS = -O3 -g -q64 -qsmp=omp -qfixed -qnosave`
- `CFLAGS = -O -g -q64`
- `LINKFLAGS = -qsmp`

It should be noted that the compilation of the code in a 64-bit mode with the compiler optimization option "qhot", together with the OpenMP option "qsmp=omp", failed on one of our subroutines, while it compiled successfully

when "qhot" was turned off. Furthermore, it was found that compilation of two other subroutines with "qsmp" resulted in runtime unstable soluions, while without "qsmp" the solutions were stable. The code was compiled without "qhot", but with "qsmp" for all runs except for those two subroutines using the options specified in the above list.

Table 2: Part1; Comparison of OVERFLOW-D runtimes (in seconds) on IBM P4 and SGI O2K based on the hybrid programming model, using the 8 million grid point test case, with a total of 2 to 16 processors

Hybrid Application						
			IBM P4			SGI O2K
Total N_{Procs}	N_{MPI}	N_{thrd}	Machine	N_{nodes}	T_{exec}	T_{exec}
2	2	1	ibm02	1	18.2	-
.	.	.	ibm01	2	18.7	-
.	.	.	ibm02+ibm01	2	19.0	-
4	4	1	ibmo2	1	10.1	24.6
.	.	.	ibmo1	4	10.6	-
.	.	.	ibm02+ibm01	2	11.5	-
.	2	2	ibm02	1	10.5	-
.	.	.	ibm01	2	10.1	-
.	.	.	ibm02+ibm01	1	10.6	-
8	8	1	ibm02	1	6.0	14.2
.	.	.	ibm01	4	5.8	-
.	.	.	ibm02+ibm01	2	6.7	-
.	4	2	ibm02	1	6.2	17.2
.	.	.	ibm01	4	6.4	-
.	.	.	ibm02+ibm01	2	6.8	-
.	2	4	ibm02	1	5.9	-
.	.	.	ibm01	2	6.1	-
.	.	.	ibm02+ibm01	2	6.0	-
16	16	1	ibm02	1	4.5	9.6
.	.	.	ibm01	4	5.6	-
.	.	.	ibm02+ibm01	2	4.7	-
.	8	2	ibm02	1	3.9	10.5
.	.	.	ibm01	4	3.6	-
.	.	.	ibm02+ibm01	2	4.2	-
.	4	4	ibm02	1	3.7	12.8
.	.	.	ibm01	4	3.8	-
.	.	.	ibm02+ibm01	2	4.2	-
.	2	8	ibm02	1	4.0	14.6
.	.	.	ibm02+ibm01	2	4.2	-

The performance results of the overd-hybrid application on IBM P4 are presented in two parts on Tables 2 and 3. The former shows results for N_{procs} = 2, 4, 8, *and* 16, and the latter for N_{procs} = 32 *and* 64. These tables consist of similar data, as in Table 1, with an additional column entitled, N_{thrd}, that lists variations in the number of OpenMP threads used per each MPI process. The following relation holds, $N_{procs} = N_{MPI} * N_{thrd}$.

For a given value of N_{procs}, variations of runs based on $N_{MPI} * N_{thrd}$ have been reported, each reflects a different distribution and access of data in the memory. Again here, similar to the MPI results in §2.1, for multiple nodes, (i.e. $N_{nodes} > 1$), the number of MPI processes is equally split between the nodes. In comparison, runtimes on ibm02, N_{procs} up to 16, are two to three times faster than the similar runs on O2K, but not much faster for $N_{procs} \geq 32$. Similarly, runs on multinodes are slower than the corresponding runs on a single node, and their pertinent runtimes are 7 to 20

In table 3, timing data which was unexpectedly slow for some runs is marked with "?" on their right side. The exact cause of the problem could not be verified, but the conjecture is that the computational node was overloaded. It should be noted that these runs are all of the split type. For instance, for the run on ibm01 with $N_{nodes} = 2$ to $N_{procs} = 32$, and $N_{MPI} = 2$ to $N_{thrd} = 16$, two MPI processes are requested on ibm01, one on the node ibm1-01sa, and one on ibm1-02sa. There are only 8 processors assigned with each of these nodes. However, 16 OpenMP threads per each of these nodes are requested; the additional 8 threads can only be provided by overloading the node. More detailed analysis of timings pertinent to the runs marked with "?", show the order of magnitude of increase in computational time on these nodes, which supports the "overload" conjecture.

3 Conclusions and Future Work

We have conducted a priliminary performance analysis of a practical CFD application based on MPI and hybrid (MPI+OpenMP) paradigms on single and multiple IBM P4 nodes using an 8 million grid point test case. Our applications ran faster on IBM P4 relative to similar runs on O2K. Due to SMP's nodal configuration and inter-nodal connection, the runtimes of the applications on multiple IBM nodes is 7 t0 20on a single node. The scalability performance of the applicaions on O2K appears somewhat better than on the IBM P4, but could not be quantified based on one test case and dataset.

Future work should focus on the scaling performance of several multi-level hybrid applications based on multi-block structured overset grids, and also on unstructured grid systems. Application test cases should be selected from among various disciplines; CFD, climate modeling and molocular dynamics. Furthermore, larger datasets should be used and tested on a larger number of IBM P4 SMPs.

Acknowledgments. The author would like to appreciate the IBM staff Charles Grassl for his useful suggestions in compilation of our application with "qsmp" option.

Table 3: Part 2; Comparison of OVERFLOW-D runtimes (in seconds) on IBM P4 and SGI O2K based on the hybrid programming model, using the 8 million grid point test case and a total of 32 to 64 total processors

Hybrid Code						
			IBM P4			SGI O2K
Total N_{Procs}	N_{MPI}	N_{thrd}	Machine	N_{nodes}	T_{exec}	T_{exec}
32	32	1	ibm02	1	2.8	5.9
.	32	1	ibm01	4	3.0	-
.	32	1	ibm02+ibm01	2	4.0	-
.	16	2	ibm02	1	3.6	5.9
.	16	2	ibm01	4	4.5	-
.	16	2	ibm02+ibm01	2	5.2?	-
.	8	4	ibm02	1	2.7	6.4
.	8	4	ibm01	4	2.4	-
.	8	4	ibm02+ibm01	2	7.3?	-
.	4	8	ibm02	1	2.8	10.1
.	4	8	ibm01	4	2.6	-
.	4	8	ibm02+ibm01	2	15.5?	-
.	2	16	ibm02	1	3.4	-
.	2	16	ibm01	2	51.?	-
.	2	16	ibm02+ibm01	2	51.?	-
64	32	2	-	-	-	3.8
.	32	2	ibm02+ibm01	2	5.4?	-
.	16	4	-	-	-	3.8
.	16	4	ibm02+ibm01	2	12.3?	-
.	8	8	-	-	-	4.0
.	8	8	ibm02+ibm01	2	21.0?	-

References

[1] http://www.nas.nasa.gov/Research/Reports/Techreports/2002/nas-02-002-abstract.html, May 2002. "Hybrid MPI+OpenMP Programming of an Overset CFD Solver and Performance Investigations".

[2] R. Meakin. "On Adaptive Refinement and Overset Structured Grids". In *Proc. 13th AIAA Computational Fluid Dynamics Conf.*, 1997. Paper 97-1858.

[3] P. G. Buning and W. Chan and K. J. Renze and D.Sondak and I. T. Chiu and J. P. Slotnick and R. Gomez and D. Jespersen. *"Overflow User's Manual"*. NASA Ames Research Center, Mountain View, CA, version 1.6au edition, 1995.

[4] A. M. Wissink and R. Meakin. "Computational Fluid Dynamics with Adaptive Overset Grids on Parallel and Distributed Computer Platforms". In *Proc. Intl. Conf. on Parallel and Distributed Processing Techniques and Applications*, pages 1628–1634, Las Vegas, NV, 1998.