A Short Report on Performance of PETSC-MUMPS solver on the Beowulf Cluster

A 2D Navier-Stokes equation was implicitly solved with finite volume based finite element method in conjunction with MUMPS direct solver (for the Cavity Test Case). In this approach, different orderings on different Beowulf Clusters (Systems A and B) were examined to solve the system of linear equations. The result will be presented in Tables (3-4).

Table (1)
Cluster System A specification

Software Specification	
OS	Linux(Redhat 7.3)
MPI	MPICH-1.2.4
PETSC	Linux-gnu(2.1.6)
Hypre	1.6.0
Hardware Specification	
CPU	Pentium AMD 1200 MHZ
Network	3Com(100 Mbit/sec)
RAM(Server Node)	768 Megabyte
RAM(Computational Node)	768 Megabyte

Table (2)
Cluster System B specification

Software Specification	
OS	Linux(Redhat 7.3)
MPI	MPICH-1.2.4
PETSC	Linux-gnu(2.1.6)
Hypre	1.6.0
Hardware Specification	
CPU	Pentium IV Intel 2.4
Network	3Com(100 Mbit/sec)
RAM(Server Node)	1 Gigabyte
RAM(Computational Node)	256 Megabyte

Some Notes and Conclusions:

1. The square cavity was discretized by structured triangular elements and a row order node numbering was used for the initial mesh.
2. The result showed that PORD ordering was the best for sequential runs from solution time point of view.
3. The result showed that PORD ordering performed better for large-scale problems in the sequential runs (For this initial ordering). System A was better than system B from Speedup point of view. As it was mentioned in Table (1) and Table (2), the cluster system A was the same as cluster system B except its CPUs was weaker than cluster system B. Based on the results presented in Tables (3-4) and the system specifications, the importance of $\frac{Communication\ \mathrm{T}ime}{\text{Computation Time}}$ parameter become clear. (Because the network for both clusters was the same (3Com (100 Mbit/sec)), therefore the Communication Time was the same. In this regard, the Computation Time plays the major role in parallel performance results. The weaker system from computation power point of view gave us better Speedup.)
4. The results showed that, AMD and AMDQUASI ordering performed better than the other orderings from Scalability and Speedup point of view for the parallel runs. In addition, AMD and AMDQUASI orderings performed better on System A than System B.

Table 3
Results obtained from system A

Pc Type(options)	KSP Type(options)	PETSC_ARCH	Mesh	NZ[1]	Time (Sec) Number Of processors				
					1	2	4	6	8
MUMP(AMD)	preonly	Linux-intel	129x129x3	799001	10.482	8.734	9.289	8.168	15.038
MUMP(AMF)	preonly	Linux-intel	129x129x3	799001	6.790	7.315	8.686	9.845	10.825
MUMP(PORD)	preonly	Linux-intel	129x129x3	799001	8.600	8.460	10.708	11.126	12.685
MUMP(METIS)	preonly	Linux-intel	129x129x3	799001	11.463	9.036	8.730	11.440	11.372
MUMP(AMDQUASI)	preonly	Linux-intel	129x129x3	799001	10.399	8.885	8.271	8.129	10.209
MUMP(AMD)	preonly	Linux-intel	183x183x3	1617629	47.986	29.191	22.416	25.340	25.035
MUMP(AMF)	preonly	Linux-intel	183x183x3	1617629	27.579	24.546	29.855	28.643	30.418
MUMP(PORD)	preonly	Linux-intel	183x183x3	1617629	26.257	26.019	26.335	27.516	30.289
MUMP(METIS)	preonly	Linux-intel	183x183x3	1617629	31.321	26.758	26.389	25.859	25.510
MUMP(AMDQUASI)	preonly	Linux-intel	183x183x3	1617629	48.046	29.204	22.415	25.861	24.959
MUMP(AMD)	preonly	Linux-intel	223x223x3	2408269	86.557	47.412	32.227	32.759	33.316
MUMP(AMF)	preonly	Linux-intel	223x223x3	2408269	55.843	46.308	36.516	37.597	42.117
MUMP(PORD)	preonly	Linux-intel	223x223x3	2408269	48.504	42.792	42.555	47.845	46.843
MUMP(METIS)	preonly	Linux-intel	223x223x3	2408269	64.742	44.017	38.514	42.964	45.951
MUMP(AMDQUASI)	preonly	Linux-intel	223x223x3	2408269	86.562	47.529	32.452	32.500	33.853
MUMP(AMD)	preonly	Linux-intel	257x257x3	3203609	139.962	62.018	50.883	42.221	52.352
MUMP(AMF)	preonly	Linux-intel	257x257x3	3203609	92.700	64.233	46.068	51.706	56.892
MUMP(PORD)	preonly	Linux-intel	257x257x3	3203609	78.237	66.521	52.160	54.227	56.500
MUMP(METIS)	preonly	Linux-intel	257x257x3	3203609	108.456	68.425	59.142	58.577	51.808
MUMP(AMDQUASI)	preonly	Linux-intel	257x257x3	3203609	136.987	62.163	50.984	42.816	51.703

[1] - NZ were obtained by –sles_view in the PETSC framework

Table 4
Results obtained from system B

Pc Type(options)	KSP Type(options)	PETSC_ARCH	Mesh	NZ	Time (Sec) Number Of processors				
					1	2	4	6	8
MUMP(AMD)	preonly	Linux-intel	129x129x3	799001	3.565	3.400	5.452	5.445	6.416
MUMP(AMF)	preonly	Linux-intel	129x129x3	799001	2.678	3.964	3.781	5.332	4.228
MUMP(PORD)	preonly	Linux-intel	129x129x3	799001	3.515	4.598	6.455	5.506	5.870
MUMP(METIS)	preonly	Linux-intel	129x129x3	799001	4.086	4.414	4.326	6.615	7.011
MUMP(AMDQUASI)	preonly	Linux-intel	129x129x3	799001	3.483	4.140	3.652	4.057	4.725
MUMP(AMD)	preonly	Linux-intel	183x183x3	1617629	13.664	10.734	9.588	9.776	9.774
MUMP(AMF)	preonly	Linux-intel	183x183x3	1617629	8.201	9.224	11.866	11.173	12.896
MUMP(PORD)	preonly	Linux-intel	183x183x3	1617629	9.280	10.108	10.832	11.463	12.387
MUMP(METIS)	preonly	Linux-intel	183x183x3	1617629	10.606	11.435	11.251	11.931	11.101
MUMP(AMDQUASI)	preonly	Linux-intel	183x183x3	1617629	13.534	11.335	9.905	9.499	9.971
MUMP(AMD)	preonly	Linux-intel	223x223x3	2408269	24.437	-*	14.068	15.246	15.004
MUMP(AMF)	preonly	Linux-intel	223x223x3	2408269	15.882	-*	16.327	16.239	17.334
MUMP(PORD)	preonly	Linux-intel	223x223x3	2408269	15.417	-*	18.009	18.873	19.116
MUMP(METIS)	preonly	Linux-intel	223x223x3	2408269	20.671	-*	16.848	18.711	18.324
MUMP(AMDQUASI)	preonly	Linux-intel	223x223x3	2408269	23.882	-*	13.861	15.514	14.775
MUMP(AMD)	preonly	Linux-intel	257x257x3	3203609	36.112	-*	21.276	20.252	19.818
MUMP(AMF)	preonly	Linux-intel	257x257x3	3203609	24.679	-*	22.024	23.201	24.981
MUMP(PORD)	preonly	Linux-intel	257x257x3	3203609	23.815	-*	23.125	23.984	23.788
MUMP(METIS)	preonly	Linux-intel	257x257x3	3203609	31.687	-*	23.365	28.250	22.312
MUMP(AMDQUASI)	preonly	Linux-intel	257x257x3	3203609	35.900	-*	21.330	18.987	20.999

[*] - Due to the lack of memory on the computational nodes, the test could not be performed