52-61

393875

358039

1997

NASA/ASEE SUMMER FACULTY FELLOWSHIP PROGRAM

MARSHAL SPACE FLIGHT CENTER

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

OPTIMIZATION OF SUPERCOMPUTER USE ON EADS II SYSTEM

Prepared By:	**Ardsher Ahmed**
Academic Rank:	**Assistant Professor**
Institution:	**University of Massachusetts Dartmouth**
Department:	**Electrical and Computer Engineering**

NASA/MSFC:	
Office:	**AI41**
Division:	**Information Systems**
MSFC Colleagues:	**Amy S. Epps**
	Debby Bowerman

Introduction

The main objective of this research was to optimize supercomputer use to achieve better throughput and utilization of supercomputers. And to help facilitate the move non-supercomputing (inappropriate for supercomputer) codes to mid-range systems for better use of Government resources at Marshal Space Flight Center (MSFC). This work involved the survey of architectures available on EADS II and monitoring customer (user) applications running on CRAY T90 system.

The overall EADS II system is designed in a hierarchical fashion, with each higher level of the hierarchy designed to provide increased computing power or special computational facilities not economically justifiable at the desktop. As more speed, memory, and compute power is required, the user moves up the hierarchy.

EADS II Hardware configurations

a. **VMCS** - Virtual Memory Compute System
   SGI Power Challenge XL (8 processors, 1 GB memory, 87 GB RMDS)
   QIC 150 tape drive, 4mm DAT tape drive, 8mm DAT tape drive

b. **VPCS** - Vector Processor Compute System
   Cray T916/4-256 (4 processors, 256 MW memory,
   512 MW SSD, 170 GB RMDS)

c. **EDFS** - EADS DMF File Server
   Cray J916 (12 processors, 512 MW memory)
   510 GB disk
   8 STK tape drives

Performance Monitoring Tools on CRAY T90 & J90

All the CRAY platforms are loaded with hardware that can allow a user to monitor the performance of CRAY supercomputer while executing the code. The Cray Research Inc.'s publication "Guide to Parallel Vector Applications [1], includes a good number of decent performance monitoring and code tuning tools. Some of these useful tools are: HPM (Hardware Performance Monitor), ATExpert, flowview, jumpview, perfview, and perftrace, which are highly recommended for supercomputing.

Performance Monitoring on SGI Power Challenge XL

There is no performance monitoring tool available on the SGI platform. One can only obtain the elapsed, run, and system times of executables. There is no direct way to determine the throughput of the system in terms of Megaflops or MIPS.

Benchmark to compare the three systems

The matrix-multiplication algorithms was designed as a benchmark to compare system performance of CRAY T90, CRAY J90, and SGI. One of the objective was to achieve the "advertised" peak performance and other was to see the affects of migrating the code from high-end supercomputers (CRAYs) to mid-range supercomputer (SGI Power Challenge XL).

Scalar Algorithm:

```
do i=1,n
 do j=1,n
  do k=1,n
     C(i,j) = C(i,j) + A(i,k)*B(k,j);
  end do
 end do
end do
```

Vector Algorithm:

```
do i=1,n
 do j=1,n
   C(:,j) = C(:,j) + A(:,i)*B(i,j);
 end do
end do
```

Brief Results of Benchmark

On SGI, the Split-C algorithm (designed specially to handle stack size problems) gave the best run time for n=256 and up, followed by vector code. The reason for vector code not producing the best run time is due to the fact that SGI Power Challenge XL is not equipped with a vector-processor facility. The run times (in seconds) are:

n=256;	real=1.1 ,	user=.73 ,	sys=.08 (split)
n=256;	real=2.94,	user=2.89,	sys=0.03 (scalar)
n=256;	real=2.56,	user=2.47,	sys=0.03 (vector)
n=512;	real=2.7 ,	user=5.83 ,	sys=.21 (split)
n=512;	real=32.85 ,	user=32.33 ,	sys=.14 (scalar)
n=512;	real=21.46 ,	user=20.96 ,	sys=.15 (vector)
n=1024;	real=28.07 ,	user=1:16.71 ,	sys=1.95 (split)
n=1024;	real=27:30,	user=21:53.04,	sys=1.9 (scalar)
n=1024;	real=3:24.25 ,	user=3:21.06 ,	sys=0.89 (Vector)
n=2*1024;	eal=10:09.61 ,	user=9:59.78 ,	sys=2.36 (split)
n=2*1024;	real 2:58:58.07,	user 2:55:33.53,	sys=12.40 (scalar)
n=2*1024;	real=33:26 ,	user=1620 ,	sys=1.9 (vector)
n=3*1024;	real=30:15.46,	user=29:34.93 ,	sys=9.87 (split)
n=3*1024;	real=10:03:22.77 ,	user=9:59:55.13 ,	sys=42.52 (scalar)
n=3*1024;	real=1:39.50.04 ,	user=1:37:10.76 ,	sys= 19.61(vector)
n=4*1024;	real=1:00:44.45 ,	user=1:53:32.11 ,	sys= 1:25.53(split)
n=4*1024;	real=23:37:20.49,	user=23:31:00.65 ,	sys=1:34.23(scalar)
n=4*1024;	real=3:38:57.95 ,	user=3:37:20.6 ,	sys=42.85 (vector)

Run Times Statistics for Matrix-Mult Benchmark on CRAY J90

n=64; Mega Flops = 49 MFlops, Exec.Time = 1.3 mSec
n=128; Mega Flops = 135.7 MFlops, Exec.Time = 8.5 mSec
n=256; Mega Flops = 181.5 MFlops, Exec.Time = 45.75 mSec
n=512; Mega Flops = 191.99 MFlops, Exec.Time = 345.88 mSec
n=1024; Mega Flops = 193.93 MFlops, Exec.Time = 2.78 Sec
n=2048; Mega Flops = 194.73 MFlops, Exec.Time = 22 Sec
n=3072; Mega Flops = 195.04 MFlops, Exec.Time = 74.28 Sec

Statistics show that peak performance 195 Mega Flops (as opposed to peak advertised rating of 200 Mega Flops) is achieved which is saturated from the n=512 and up. The run times are from 1.3 ms to 74.28 seconds. These fast times are due to pipelining and vectorization on J90 (not available on SGI).

Run Times Statistics for Matrix-Mult Benchmark on CRAY T90

n=4096;	1.637 GFLOPS,	VL = 127.97,	t = 22.24 s
n=3072;	1.64 GFLOPS,	VL = 127.97,	t = 8.83 s
n=2048;	1.628 GFLOPS,	VL = 127.95,	t = 2.89 s
n=1024;	1.633 GFLOPS,	VL = 127.89,	t = 0.33 s
n=512;	1.607 GFLOPS	VL=127.76,	t = 46 ms
n=256;	1.475 GFLOPS	VL=127.25,	t = 12.63 ms
n=128;	1.011 GFLOPS	VL=124.56,	t = 2.67 ms
n=64;	0.267 GFLOPS	VL=60.91,	t = 0.22 ms

Statistics show that peak performance 1.64 Giga Flops (as opposed to peak advertised rating of 1.8 Giga Flops) is achieved which is saturated from the n=1024 and up. The run times are from 0.22 ms to 22.24 seconds. The T90 has achieved the speed up of up to 8 on this problem compared to J90 system. The Vector Lengths (VL) achieved are nearly ideal (peak VL=128). This shows that code is highly vectorized.

Customer Code Performance Analysis

Various application packages and user codes were analyzed on CRAY T90 during this research. Due to 5 page space limitation of this report it is not possible to include the performance statistics except for the MSC-NASTRAN which is a widely used application at MSFC. A benchmark was run at Johnson Space Center to compare J90 and SGI platforms. We extended this study by running same model on T90 machine. Following tables indicate the performance, vector lengths, run times, and analysis/comments.

TABLE 1
MSC-NASTRAN Performance Measurements on CRAY T90

Test case	Mflops	Vector Length	S/V Ratio	Elapsed Time (sec)	User CPU Time (sec)	Sys CPU Time (sec)	Comment
test0	110.52	63.78	4:1	8448	1598.37	384.66	scalar code
[illegible]	1.18	14.61	143:1	72	32.47	3.27	scalar code
test2b	1.03	14.63	49:1	20	5.62	0.68	scalar code
test2c	0.49	28.38	25:1	9	1.39	0.37	scalar code
test2d	0.37	32.90	23:1	9	1.2	0.39	scalar code
test2e	0.99	14.57	37:1	13	3.56	0.31	scalar code
test2f	1.02	14.06	38:1	14	3.97	0.32	scalar code

TABLE 2
Summary of Run Times for "test0" on 3 platforms

Time in (hr:min:sec)	SGI Power Challenge	CRAY T90	CRAY J90
Elapsed	1:43:11.72	2:20:48	5:46:1.46
User	1:23:42.39	26:29.37	1:57:50.92
System	12:44.47	6 24.66	11:19.27

Source: SGI Power Challenge and CRAY J90 data from report by Kevin Partins at JSC
CRAY T90 data compiled by running same model by A. Ahmed and Ric Moore (7/18/97).

Analysis of MSC-NASTRAN software

- The elapsed time is the most essential of all times as it is the turn-around time for the job. The SGI Power Challenge is faster than both CRAY systems!! SGI is 37 minutes faster than T90, and 4 hr 3 minutes faster than J90.

- The user time is best on T90 system due to fast I/O and memory banks but is overshadowed by T90's huge elapsed time. This may be due to the load on the T90 but depicts the realistic situation as the standalone CRAY T90 is almost impossible to find.

- The system times are pretty comparable from 6 to 12 minutes across all the three platforms.

- The vector lengths utilized and the scalar/vector (hold issue) ratios indicate that MSC-NASTRAN is predominantly a scalar code. It must be noted that the fortran90 compiler provides the best code vectorization capability and is strongly recommended for future releases.

- Based on this analysis, it is strongly recommendation that either:

  MSC-NASTRAN be rewritten and optimized in vector form to take full advantage of vector facility on CRAY platforms, or

  MSC-NASTRAN be migrated to the SGI Power Challenge system in its current version.

Conclusions and Recommendations

Most of the MSFC computing community is relying on the software developed by a third party code developer, to which they have no control over to modify or optimize. And most of this licensed software is written in FORTRAN 77 or FORTRAN IV without any vectorization or performance tuning for supercomputers. Basically it is "dusty-deck" software and no matter how powerful the supercomputer is it cannot speed up the code beyond a factor of 2, which can be attributed to a faster clock rate on CRAY T90. In order to attain the performance close to the peak one has to rewrite the code. Following are some of the recommendations for code developers and supercomputer users

- Just running a program on a supercomputer does not guarantee "supercomputing" performance
- Use best algorithm for solving a problem
- Optimize the code by hand (dependence analysis)
- Use compiler optimization flags
- Know the details of your Supercomputer hardware and limitations
- Use FORTRAN90 or HPF
- Use performance monitoring tools to tune the code for supercomputing

Acknowledgments: I would like to thank Dr. Jerry Karr, and my MSFC colleagues Ms. Debby Bowerman, Ms. Eppy Epps, Jerry Gray, Mike Knowels, Tom Brister, Ed Reske, Ric Moore, Ms. Vertayn Spencer, Ms. Vu, Ms. Patty Montgomery, who all contributed to my work and made my experience at MSFC an enjoyable one.

Reference:

[1] Guide to Parallel Vector Applications, SG-2182 2.0, Cray Research Inc. publication.