C-DAC Four Days Technology Workshop

ON

Hybrid Computing – Coprocessors/Accelerators
Power-Aware Computing – Performance of Applications Kernels

hyPACK-2013
(Mode-1 : Multi-Core)

Lecture Topic:
Multi-Core Processors : Benchmarks (Part-III)

Venue : CMSD, UoHYD ; Date : October 15-18, 2013

Lecture Outline

Following Topics will be discussed

- ❖ Performance of Benchmarks
- ❖ Benchmark results on Multi Core Systems

Performance Characteristics

Approaches to measure performance

- Several Approaches exist to measure performance of a Multocre System
  - Summarize several key architecture of a given computer system and relate them in order to get measure of its performance
  - Most of the measures are based on some engineering or design considerations rather theoretical calculations
  - Define set of programs and observe the system's run times on those programs

Performance Characteristics: Peak Performance

Peak Performance

- Defined as the MFLOPS rate which the manufacturer guarantees the computer will never exceed
  - It is obtained by taking the clock rate of the given system an dividing it by the number of clock cycles a floating point instruction requires
- Peak Performance calculations assume the maximum number of operations that the hardware can execute in parallel or concurrently
- Peak Performance is a rough hardware measure; it essentially reflects the cost of the system
- There are some (rare) instances where peak performance can give a creditable idea of performance

Performance Characteristics: Sustained Performance

Sustained Performance

- It may be defined as the highest MFLOPS rate that can actual program achieved doing something recognizably useful for certain length of time

- It essentially provide an upper bound on what a programmer may be able to achieve

Efficiency rate = The achieved (sustained) performance divided by the peak performance

Note : The advantage of this number is its independent of any absolute speed.

Performance: Benchmarks Classification

Benchmark Classification

- Benchmarks can be classified according to applications
  - Scientific Computing
  - Commercial applications
  - Network services
  - Multi media applications
  - Signal processing
- Benchmark can also be classified as
  - Macro benchmarks and Micro benchmarks

Benchmarks on Multi Core Systems

(Contd…)

Micro/Macro Benchmarks for Multi Core Processors

Name	Area
SuperLU HPCC Suite (Top-500	Numerical Computing (Linear Algebra)
LMBENCH	System Calls & Data movement in Unix/Linux Environment
LLCBench	Low Level Cache Benchmarks
STREAM	Memory Bandwidth
I/O -Bench	I/O Benchmarks
TIO-Bench	Thread I/O Benchmarks
NAMD	Nanoscale Molecular Dynamics
NAS	Computational Fluid Dynamics

Micro/Macro Benchmarks : HPCC Benchmark Suite

HPCC Benchmark on Multi Cores

a) **Top-500 :** Peak /Sustained Performance : Matrix Solution of Linear System of Equations [**A] {x} = {b} .**

b) **Ptrans :** Transpose of a Matrix Algorithms

c) **STREAM :** COPY; SCALE; SUM; TRIAD **(OpenMP)**

d) **DGEMM :** Single DGEMM (Double Precision Matrix into Matrix Multiplication, part of **BlasBench**)

e) **Random Access** –Random Access - GUP/s

f) **Fast Fourier Transformations - FFT**

g) **(b_eff) :** MPI Benchmarks

Part-I : Results on

AMD Opteron (2/4/8 Socket 2 Dual Core)

Dual Core System : Configuration & Prog. Env

System Details	Multi Core : IWILL H205	SunFire 4600
CPU	Dual-Core AMD Opteron (tm) Processor 8218	Dual-Core AMD Opteron(tm) Processor 885
No of Sockets/Cores	4 Sockets (Total : 8 Cores)	8 Sockets (Total : 16 Cores)
Clock-Speed	2.6 GHz per core	2.6 GHz per core
Peak(Perf.)	41.6 Gflops	83.2 Gflops
Memory/Core	**1 GB per core**	**4 GB per core**
Memory type	DDR2	DDR2
Total Memory	**8 GB**	**64 GB**
Cache	L1 = 128 KB; L2 = 1 MB	L1 = 128 KB; L2 = 1 MB
OS	Cent OS 4.4 x86_64 (64 bit)	CentOS 4.4 (Final) x86_64 (64 bit)
Compilers	Intel 9.1(icc; fce; OpenMP)	Intel 9.1(icc; fce; OpenMP)
MPI	mpicc:Intel MPI 2 .0 7 gcc/gfortran mpiicc : Intel MPI 2.0 /icc, ifort	mpicc:Intel MPI 2 .0 7 gcc/gfortran mpiicc : Intel MPI 2.0 /icc, ifort
Math Libraries	ACML 3.5.0	ACML 3.5.0

Dual Core System : Configuration & Prog. Env

System Details	Multi Core : HP DL485	DELL PowerEdge 6950
CPU	Dual-Core AMD Opteron (tm) Processor 8200 SE	Dual-Core AMD Opteron™ Processor 8218
No of Sockets/Cores	4 Sockets (total 8 Cores)	4 Sockets (total 8 Cores)
Clock-Speed	2.8 GHz per core	2.6 GHz per core
Peak Performance	44.8 Gflops	41.2 Gflops
Memory/Core	2 GB per core	2 GB per core
Memory type	DDR2 667 MHz	DDR2 667 MHz
Total Memory	16 GB	16 GB
Cache	L1 = 64 KB; L2 = 1 MB	L1 =128 KB; L2 = 2 MB
OS	Cent OS 4.4 x86_64 (64 bit)	Cent OS 4.4 x86_64 (64 bit)
Compilers	Intel 9.1(icc; fce; OpenMP)	Intel 9.1(icc; fce; OpenMP)
MPI	mpicc:Intel MPI 2 .0 7 gcc/gfortran mpiicc : Intel MPI 2.0 /icc, ifort	mpicc:Intel MPI 2 .0 7 gcc/gfortran mpiicc : Intel MPI 2.0 /icc, ifort
Math Libraries	ACML 3.5.0	ACML 3.5.0

Dual Core : Prog. Env & Benchmark Rules

Rule 1 :The same Prog. environment is used for execution of Benchmark

Rule 2 :Process (Job) binding to particular CPU(s) is not considered

Rule 3 : Algorithmic parameters are not FULLY optimised

Computing System	Programming Environment Details	
IWILL SunFire HP DL485 DELL	Operating System	Cent OS 4.4 x86_64 (64 bit), Kernel 2.6.9
	Compilers	Intel 9.1.(icc; fce; OpenMP)
	Flags	- O3, -ip -funroll-loops,
	MPI	Intel MPI 2.0
	Libraries	ACML 3.5.0

Remarks 1 : For selective Benchmark suites (HPCC), the input problem parameters are same on the target systems. Tuning & Performance Optimisation is not carried out

Dual Core : Programming Environment

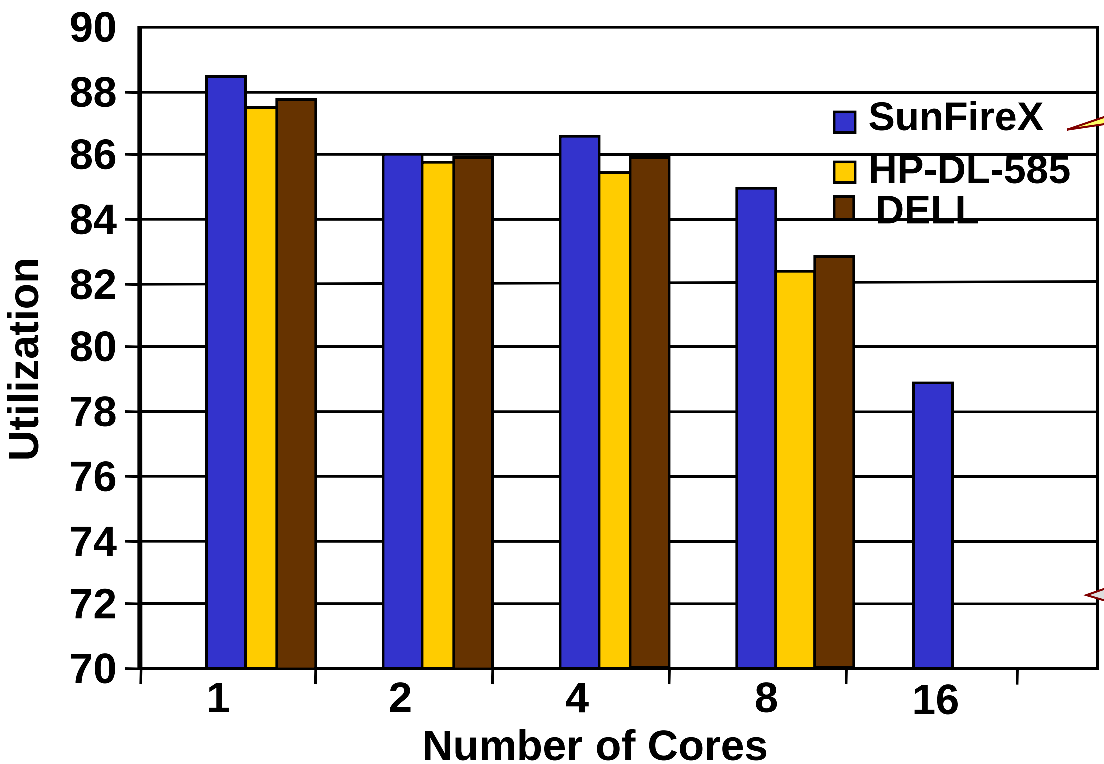
OS	Cent OS 4.4 x86_64 (64 bit), Kernel 2.6.9
Compilers	Intel 9.1.(icc; fce; OpenMP)
Flags	-O3, -ip -funroll-loops,
MPI	Intel MPI 2.0
Libraries	ACML 3.5.0

Dual Cores: HPCC – (Top-500) : Results & Performance

Comp. System	Multi Core (CPUs)	Matrix Size/ Block size/ (P,Q)	Peak Perf (Gflops)	Sust. Perf (Gflops)	Utilization (%)
SunFireX	**1**	**46080/128(1,1)**	**5.2**	**4.60**	**88.46**
	2	**46080/128(2,1)**	**10.4**	**8.94**	**86**
	4	**52000/160(2,2)**	**20.8**	**18.01**	**86.58**
	8	**56320/128(4,2)**	**41.6**	**35.37**	**85.02**
	16	**72000/192(8,2)**	**83.2**	**65.66**	**78.91**
HP-DL585	**1**	**24000/160(1,1)**	**5.6**	**4.95**	**87.5**
	2	**38400/160(2,1)**	**11.2**	**9.6**	**85.8**
	4	**38400/160(2,2)**	**22.4**	**19.2**	**85.5**
	8	**40448/160(4,2)**	**44.8**	**36.9**	**82.4**

Algorithm parameters, tuning & performance Compiler optimisations are not tried to extract the sustained performance.

Dual Cores: HPCC – (Top-500) : Results & Performance



Minimum and Maximum Performance differ by 2-4 %

Choice of HPL Problem Size parameters play an important role

Issues to be addressed for Performance Enhancements

- Memory per Core
- L1 Cache
- Tuned Compilers
- Mathematical Kernels
- Operative System
- Process Affinity

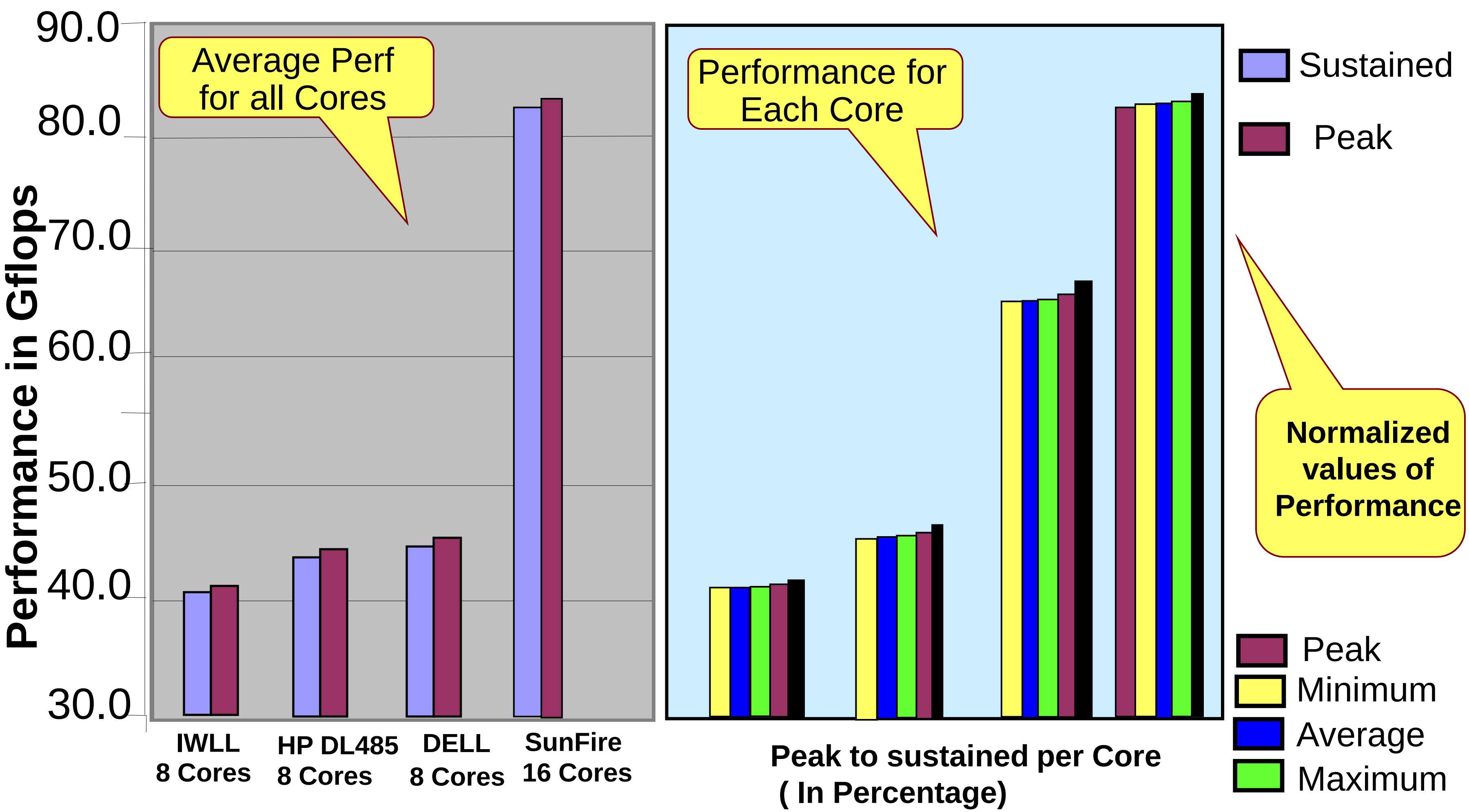
Algorithm parameters, tuning & performance of Compiler optimisations are not tried to extract the sustained performance.

Dual Cores: HPCC –Top 500 : Performance

Computing System	Multi Core (CPUs)	Matrix Size/ Block size/ (P,Q)	Peak Perf (Gflops)	Sust. Perf (Gflops)	Utilization (%)
IWILL	1	25600/128(1,1)	5.2	4.498	86.5
	2	25600/128/(2,1)	10.4	8.76	84.2
	4	25600 /120(4,1)	20.8	17.1	82.1
	8	30208/128(8,1)	41.6	31.7	76.7
SunFireX	1	25600/128(1,1)	5.2	4.60	88.5
	2	25600/128/ (2,1)	10.4	8.799	84.6
	4	25600 /120(4,1)	20.8	17.3	83.2
	8	30208/128(8,1)	41.6	32.0	76.9
HP-DL585	1	25600/128(1,1)	5.6	4.86	86.87
	2	25600/128 (2,1)	11.2	9.286	82.91
	4	25600 /120(4,1)	22.4	18.43	82.27
	8	30208/128(8,1)	44.8	33.84	75.54
DELL PowerEdge 6950	1	25600/128(1,1)	5.2	4.57	87.88
	2	25600/128 (2,1)	10.4	8.84	85.00
	4	25600 /120(4,1)	20.8	17.15	82.45
	8	30208/128(8,1)	41.6	29.6	71.15

For Top-500, algorithm parameters, tuning & performance of Compiler optimisations are not tried to extract the sustained Performance. Input Parameters on IWILL /SUNFIRE /HP DL585 /DELL are precisely the same.

Dual Cores: HPCC – (DGEMM) : Results & Performance



For DGEMM, algorithm parameters, tuning & performance of Compiler optimisations are not tried to extract the sustained Performance.

Dual Cores: HPCC – (DGEMM) : Results & Performance

System	Input Matrix Size	Clock Speed / Memory per Core / L1 Cache	Rating in Gflops for all CPUs (2/4/8 Cores)			Rating in Gflops per CPU
			Minimum	Average	Maximum	
IWILL	25600	**5.2**	4.762	4.764	4.765	4.767
SunFireX	25600	**5.2**	4.776	4.781	4.791	4.779
HP-DL585	38400	**5.6**	5.136	5.137	5.137	5.116
DELL	30208	**5.2**	4.701	4.710	4.739	4.745

For DGEMM, algorithm parameters, tuning & performance of Compiler optimisations are not tried to extract the sustained Performance.

Dual Cores: HPCC – (MPI) : Results & Performance

Computing System	Multi Core (CPUs)	*Ping-Pong*		*Naturally Ordered Ring*		*Randomly Ordered Ring*	
		Latency msecs	*Bandwidth Mbytes/Sec*	*Latency msecs*	*Bandwidth Mbytes/Sec*	*Latency msecs*	*Bandwidth Mbytes/Sec*
I WILL H4205	4	0.0099	809.725	0.0123	371.0213	0.0124	362.4078
Sun Fire X4600	4	0.0099	694.182	0.0138	306.8282	0.0137	300.5981
Dell Power Edge 6950	4	0.01023	837.442	0.0226	363.1904	0.0233	359.0649

HPCC - HLRS MPI suite used

For MPI Suite, algorithm parameters, tuning & performance of Compiler optimisations are not tried to extract the sustained Performance.

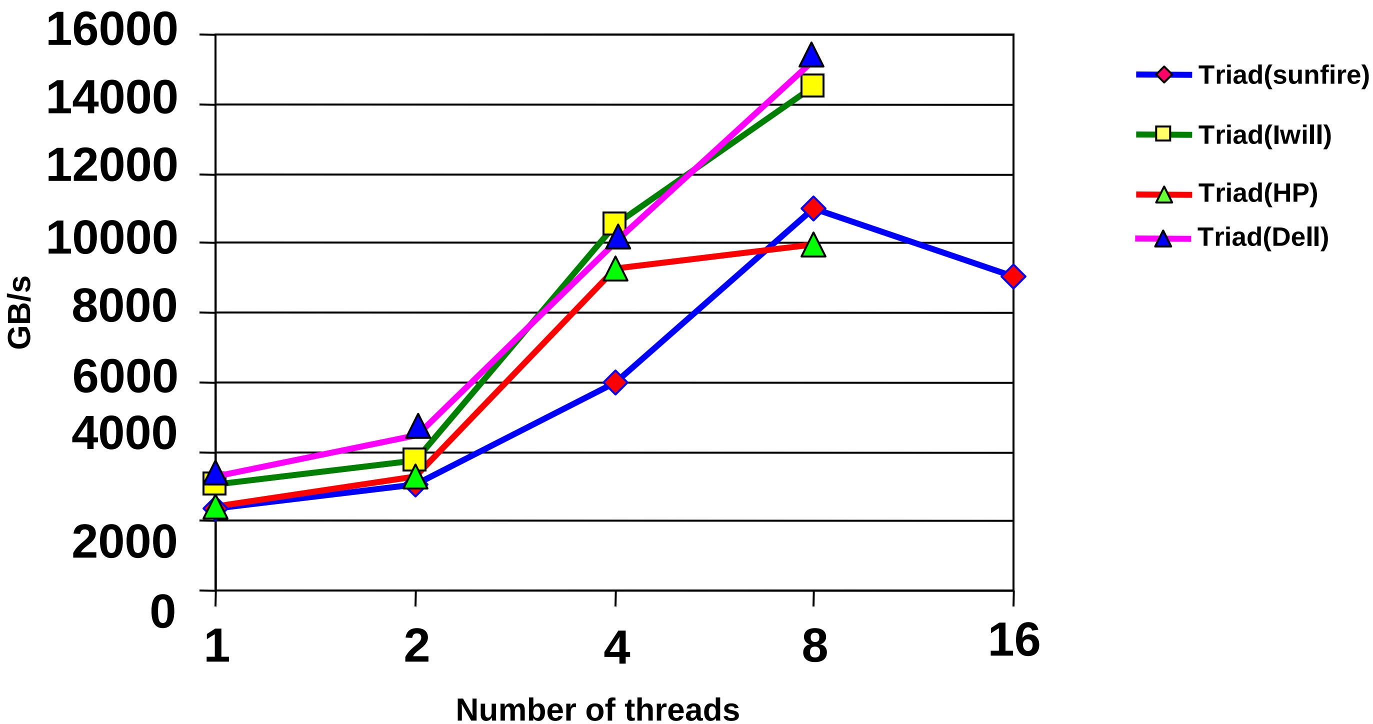
Dual Cores: HPCC – (MPI) : DELL Systems

Multi Core (CPUs) /Problem Size	***Ping-Pong***		***Naturally Ordered Ring***		***Randomly Ordered Ring***	
	Latency msecs	***Bandwidth Mbytes/Sec***	***Latency msecs***	***Bandwidth Mbytes/Sec***	***Latency msecs***	***Bandwidth Mbytes/Sec***
2 (2X1) 25600/120	0.0088	861.32	0.010	405.41	0.0100	401.35
4 (2X2) 38400/160	0.0128	849.73	0.012	377.35	0.0110	367.38
4 (4X1) 25600/160	0.0110	837.44	0.022	363.19	0.023	359.06
8 (4X2) 30720/128	0.0140	837.98	0.013	344.48	0.0137	331.95
8 (4X2) 30208/128	0.0147	837.97	0.013	344.48	0.014	331.95

HPCC - HLRS MPI suite used

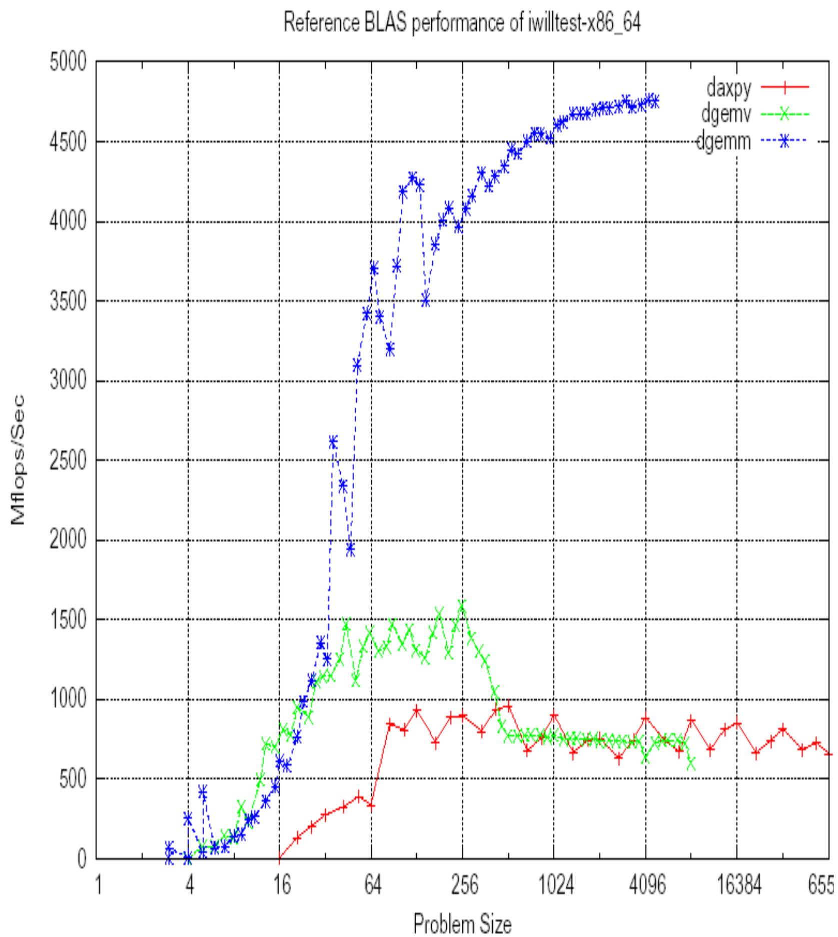
For MPI suite, algorithm parameters, tuning & performance of Compiler optimisations are not tried to extract the sustained Performance.

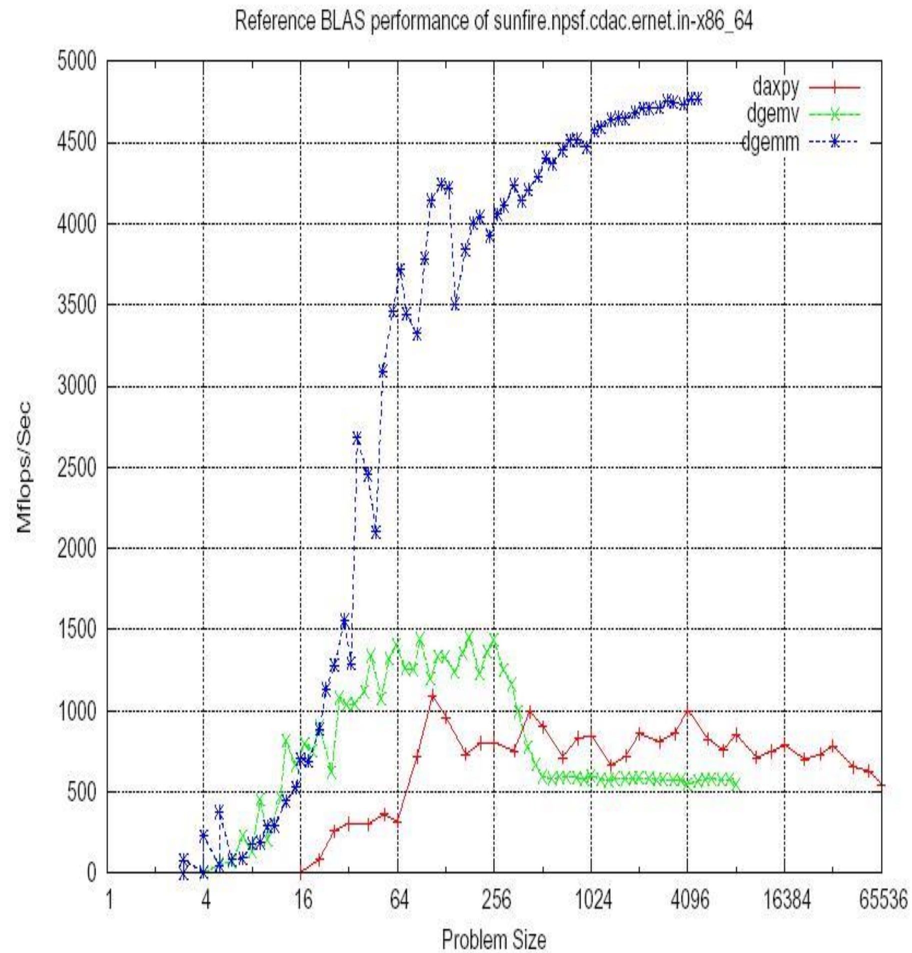
Dual Cores: STREAM (Triad) : Performance



For STREAM, algorithm parameters, tuning & performance of Compiler optimisations are not tried to extract the sustained Performance.

Dual Cores: BLASBench Performance





For LLCBench, algorithm parameters, tuning & performance of Compiler optimisations are not tried to extract the sustained Performance.

Dual Cores: LMBENCH Performance

a) **LMBENCH:** System Calls and data movement operations in Unix; to measure the Operating system overheads and the capability of data transfer between processor, cache, memory and network, disk on various Unix platforms.

- **Bandwidth (MB/s):** Memory Copy, File Read, Pipe, TCP (Results are available.)
- **Latency(μs) :**Memory Read , File Create, Pipe, TCP (Results are available.)
- **System Overhead(μs) :** Null system call , Process creation , Context Switching (Results are available)

Dual Cores: LLCBench Performance

LLCBench

– **BLASBench :** equivalent to HPCC-DGEMM. DGEMM (double precision Matrix into Matrix Multiplication) achieved performance of 4.7 Gflops, which is equivalent to 90 % of the peak performance (5.2 Gflops)

– **MpBench** – MPI benchmarks executed (Equivalent version of HPCC-HLRS Suites available)

– **CacheBench** – Low level Cache Benchmarks executed

C-DAC in-house developed Benchmarks : Suite III

Objective : Set of two test suites which perform Matrix into Vector and Matrix into Multiplication Algorithms on Dual Core Systems

Type of Computations : Integer Operations **Compiler flags Used** : -O3

Suite of Dense Matrix Computations (Integer Operations)				
Prog Env : Pthreads**; Matrix Size** :2048 **Memory : 48** MB**; Problem Size :** Class B				
Comp Systems	**CPUs (Process) Time in Seconds**			
	1	**2**	**4**	**8**
IWILL				
SunFireX	**154.87**	**78.06**	**44.93**	**23.51**
HP DL585	**165.38**	**81.78**	**51.33**	**21.05**
DELL-6950			**43.16**	**21.81**

For C-DAC In-house multi-threaded problems, algorithm parameters, tuning & performance of Compiler optimisations are not tried to extract the sustained Performance.

C-DAC in-house developed Benchmarks : Suite III

Objective : Set of two test suites which perform Matrix into Vector and Matrix into Multiplication Algorithms on Dual Core Systems

Type of Computations : Floating Point Operations

Compiler flags Used : -O3

Suite of Dense Matrix Computations (Floating Point Operations)				
Prog Env : Pthreads**; Matrix Size** :4096 **Memory :**384 MB**; Problem Size :** Class C				
Comp Systems	**CPUs (Process) Time in Seconds**			
	1	**2**	**4**	**8**
SunFireX	**1892.46**	**944.79**	**502.95**	**254.26**
HP DL585	**1829.77**	**903.69**	**458.18**	**250.74**

For C-DAC In-house multi-threaded problems, algorithm parameters, tuning & performance of Compiler optimisations are not tried to extract the sustained Performance.

C-DAC in-house developed Benchmarks : Suite III

Objective : Set of two test suites which perform Matrix into Vector and Matrix into Multiplication Algorithms on Dual Core Systems

Type of Computations : Floating Point Operations **Compiler flags Used** : -O3

Suite of Dense Matrix Computations (Floating Point Operations)				
Prog Env : OpenMP**; Matrix Size** :2048 **Memory :** 96 MB**; Problem Size :** Class B				
Comp Systems	**CPUs (Process) Time in Seconds**			
	1	**2**	**4**	**8**
IWILL				
SunFireX	**547.05**	**275.62**	**152.27**	**92.06**
HP DL585				
DELL-6950				

For C-DAC In-house multi-threaded problems, algorithm parameters, tuning & performance of Compiler optimisations are not tried to extract the sustained Performance.

Part-II : Results on

Intel Quad Socket /Quad Core Systems

Inte Caneland (Quad Socket /Quad Core)

Comp System Conf.	Intel Caneland (Quad Socket Quad Core)
CPU	Quad-Core Genuine Intel(R) CPU - Tigerton
No of Sockets /Cores	4 Sockets (Total : 16 Cores)
Clock-Speed	2.4 GHz per Core
Peak(Perf.)	153.6 Gflops
Memory/Core	4 GB per Core
Memory type	FBDIMM
Total Memory	16 GB
Cache	L1 = 128 KB; L2 = 8 MB Per socket shared
OS	Red Hat Enterprise Linux Server release 5 (Tikanga) x86_64 (64 bit)
Compilers	Intel 10.0(icc; fce; OpenMP)
MPI	Intel (/opt/intel/ict/3.0.1/mpi/3.0/bin64)
Math Libraries	Math Kernel Library 9.1

Inte Caneland (Quad Socket /Quad Core)

Benchmark Execution Run-Rules

Rule 1: The same programming environment is used for execution of Micro/Macro Benchmarks on all the Computing systems.

Rule 2: Process (Job) affinity to particular CPU(s) is not considered explicitly.

Rule 3: Algorithmic parameters for various Benchmarks/Kernels are not optimized. The Compiler flags are not fully explored to get the best sustained performance

Rule 4: The obtained performance interims of Gflop/s or the execution time or the values or parameters with respect to specific Benchmark/kernels is considered as a reporting metric.

Rule 5: For specific Benchmarks/Kernels, the problem data is fixed and measure the performance.

DGEMM : Benchmarks on Quad Core Processors

Multi Core (CPUs)	DGEMM Input Matrix Size (HPCC)	Rating in Gflops			Rating in Gflops/s per CPU
		Minimum	Average	Maximum	
1	36000; DGEMM_N =20480	8.255	8.255	8.255	8.272
2	40960; DGEMM_N =14481	8.411	8.418	8.426	8.507
4	40960; DGEMM_N =10240	8.109	8.118	8.123	8.255
8	42240; DGEMM_N = 7467	8.103	8.114	8.123	8.450
16	40960; DGEMM_N = 5000	6.760	7.036	7.188	8.543

Used Env : Intel 10.0(icc, MPI); Compiler Flag : -O3, -funroll-loops,- fomit-frame-pointer.

For DGEMM, algorithm parameters, tuning & performance of Compiler optimisations are not tried to extract the sustained Performance.

Top500 : Benchmarks on Quad Core Systems

Multi Core (CPUs)	HPL Matrix Size/ Block size/ (P,Q)	Peak Perf (Gflops)	Sust. Perf (Gflops)	Utilization (%)
1	36000/120(1,1)	9.6	8.6	89.58
2	40960/120 (2,1)	19.2	16.68	86.87
4	40960/120(2,2)	38.4	32.54	84.73
8	42240/120(4,2)	76.8	60.72	79.06
16	40960/200(4,4)	153.6	97.09	63.20
	83456/200(4,4) Used 56 GB	153.6$	116.2	76.0
	92000/200(4,4) * 64 GB can be used	153.6*	121.3	79.0

Used Env : Intel 10.0(icc, MPI); Compiler Flag : -O3, -funroll-loops,- fomit-frame-pointer

For Top-500, algorithm parameters, tuning & performance of Compiler optimisations are not tried to extract the sustained Performance.

Multi Cores: HPCC – (MPI) : Results & Performance

Benchmark Conditions	Multi Core (CPUs)	*Ping-Pong*		*Naturally Ordered Ring*		*Randomly Ordered Ring*	
		Latency msecs	*Bandwidth Mbytes/Sec*	*Latency msecs*	*Bandwidth Mbytes/Sec*	*Latency msecs*	*Bandwidth Mbytes/Sec*
Latency Measurements with 8 Bytes	2	0.0010	1073.6	0.0009	520.5	0.0009	521.309
	4	0.00134	873.49	0.0015	337.38	0.0015	303.90
Bandwidth Measurements with 2 MB Size	8	0.0015	811.79	0.0015	231.95	0.0017	221.20
	16	0.00143	726.300	0.0017	110.69	0.00198	106.14

HPCC - HLRS MPI suite used

Used Env : Intel 10.0(icc, MPI); Compiler Flag : -O3, -funroll-loops, -fomit-frame-pointer.

For HPCC MPI Suite, In-house multi-threaded problems, algorithm parameters, tuning & performance of Compiler optimisations are not tried to extract the sustained Performance.

Quad Cores: HPCC – (STREAM)

Performance for STREAM OpenMP (Triad).

Multi Core (CPUs)	Rate(MB/s) 1 GB Triad
1	3117.01
2	5869.17
4	5752.83
8	7175.5
16	9534.1

Used Env **:** Intel 10.0(icc, OpenMP); Compiler Flag : -O3,

For STREAM, algorithm parameters, tuning & performance of Compiler optimisations are not tried to extract the sustained Performance. The reported Results for fixed problem size i.e., 1 GB

EEMC Benchmarks

- **EEMBC**, the Embedded Microprocessor Benchmark Consortium,
  - Itis a non-profit corporation formed to standardize on real-world, embedded benchmark software to help designers select the right embedded processors for their systems.
- **EEMBC** is a collection of "algorithms" and "applications" organized into benchmark suites targeting telecommunications, networking, digital media, Java, automotive/industrial, consumer, and office equipment products.

Source : http://www.eembc.org/

EEMC Benchmarks

- **EEMBC**, the Embedded Microprocessor Benchmark Consortium,
- An additional suite of benchmarks, called MultiBench, specifically targets the capabilities of multicore processors based on an SMP architecture.
- These benchmarks may be obtained by joining EEMBC's open membership or through a corporate or university licensing program.
- The EEMBC Technology Center manages development of new benchmark software and certifies benchmark test results.

Source : http://www.eembc.org/

EEMC Benchmarks

❖ Benchmark Scores

- Automotive Consumer Digital Entertainment Java/CLDC

❖ Software Licensing and Membership

- AutoBench, ConsumerBench, DENBench, GrinderBench (Java)

❖ Hypervisors

- MultiBench, Networking, OABench, TeleBench, HyperBench

❖ Power/Energy

- EnergyBench

Source : http://www.eembc.org/

EEMC Benchmarks

❖ MultiBench™ 1.0 Multicore Benchmark Software

- Extends EEMBC benchmark scope to analyze multicore architectures, memory bottlenecks, OS scheduling support, efficiency of synchronization, and other related system functions.
- Measures the impact of parallelization and scalability across both data processing and computationally intensive tasks
- Provides an analytical tool for optimizing programs for a specific processor
- Leverages EEMBC's industry-standard, application-focused benchmarks in hundreds of workload combinations
- First generation targets the evaluation and future development of scalable SMP architectures
- MultiBench™ is a suite of embedded benchmarks that allows processor and system designers to analyze, test, and improve multicore architectures and platforms. MultiBench uses standardized workloads and a test harness that provides compatibility with a wide variety of multicore embedded processors and operating systems.

Summary

- Characteristics of Benchmarks are discussed

- Well known Benchmarks such as LLCBench, ScaLAPACK, LINPACK, NAS,HPCC suite are executed on Multi Core Systems.

- A set of in-house benchmarks are executed & obtained performance on Multi Core / Cluster of Multi Core Systems.

- EEMC Benchmarks for Multi-Core Performance

Dual/Quad Cores: Observations

- Performance from Dual to Quad Core Processors increases which merely depends upon the algorithm, memory access, bandwidth, latency at different levels…. However, it can be improved using Multi-threading Programming Paradigms.
- The HPL, DGEMM are sensitive to data size required from memory to Core, data locality and cache usage. The results for HPL and DGEMM may show improvement for various problem sizes from Scalability point of view.
- Benchmarks which are highly compute and memory intensive may give performance improvement for large problem sizes on Quad Core system from Scalability point of view.
- The performance of many benchmarks can be improved by at most 5% to 10% if memory per Core is increased beyond 4GB with fine tuning of Algorithms and Compliers.
- For all benchmarks, algorithm parameters, tuning & performance of Compiler optimisations are not tried to extract the sustained Performance of benchmarks on the Dual core /Quad core systems.

References

1. Andrews, Grogory R. **(2000),** Foundations of Multithreaded, Parallel, and Distributed Programming, Boston, MA : Addison-Wesley

2. Butenhof, David R **(1997),** Programming with POSIX Threads , Boston, MA : Addison Wesley Professional

3. Culler, David E., Jaswinder Pal Singh **(1999),** Parallel Computer Architecture - A Hardware/Software Approach , San Francsico, CA : Morgan Kaufmann

4. Grama Ananth, Anshul Gupts, George Karypis and Vipin Kumar **(2003),** Introduction to Parallel computing, Boston, MA : Addison-Wesley

5. Intel Corporation, **(2003),** Intel Hyper-Threading Technology, Technical User's Guide, Santa Clara CA : Intel Corporation Available at : http://www.intel.com

6. Shameem Akhter, Jason Roberts **(April 2006),** Multi-Core Programming - Increasing Performance through Software Multi-threading , Intel PRESS, Intel Corporation,

7. Bradford Nichols, Dick Buttlar and Jacqueline Proulx Farrell **(1996),** Pthread Programming O'Reilly and Associates, Newton, MA 02164,

8. James Reinders, Intel Threading Building Blocks – (**2007**) , O'REILLY series

9. Laurence T Yang & Minyi Guo (Editors), (**2006**) *High Performance Computing - Paradigm and Infrastructure* Wiley Series on Parallel and Distributed computing, Albert Y. Zomaya, Series Editor

10. Intel Threading Methodology ; Principles and Practices Version 2.0 copy right **(March 2003),** Intel Corporation

11. William Gropp, Ewing Lusk, Rajeev Thakur **(1999),** Using MPI-2, Advanced Features of the Message-Passing Interface, The MIT Press..

References

12. Pacheco S. Peter, **(1992),** Parallel Programming with MPI, , University of Sanfrancisco, Morgan Kaufman Publishers, Inc., Sanfrancisco, California

13. Kai Hwang, Zhiwei Xu, (**1998**), Scalable Parallel Computing (Technology Architecture Programming), McGraw Hill New York.

14. Michael J. Quinn (**2004**), Parallel Programming in C with MPI and OpenMP McGraw-Hill International Editions, Computer Science Series, McGraw-Hill, Inc. Newyork

15. Andrews, Grogory R. **(2000),** Foundations of Multithreaded, Parallel, and Distributed Progrmaming, Boston, MA : Addison-Wesley

16. SunSoft. Solaris multithreaded programming guide. SunSoft Press, Mountainview, CA, **(1996),** Zomaya, editor. Parallel and Distributed Computing Handbook. McGraw-Hill,

17. Chandra, Rohit, Leonardo Dagum, Dave Kohr, Dror Maydan, Jeff McDonald, and Ramesh Menon, **(2001)**,Parallel Programming in OpenMP San Fracncisco Moraan Kaufmann

18. S.Kieriman, D.Shah, and B.Smaalders **(1995),** Programming with Threads, SunSoft Press, Mountainview, CA. 1995

19. Mattson Tim, **(2002),** Nuts and Bolts of multi-threaded Programming Santa Clara, CA : Intel Corporation, Available at : http://www.intel.com

20. I. Foster **(1995,** Designing and Building Parallel Programs ; Concepts and tools for Parallel Software Engineering, Addison-Wesley (1995)

21. J.Dongarra, I.S. Duff, D. Sorensen, and H.V.Vorst **(1999),** Numerical Linear Algebra for High Performance Computers (Software, Environments, Tools) SIAM, 1999

22. OpenMP C and C++ Application Program Interface, Version 1.0". **(1998),** OpenMP Architecture Review Board. October 1998

23. D. A. Lewine. *Posix Programmer's Guide:* **(1991),** Writing Portable Unix Programs with the Posix. 1 Standard. O'Reilly & Associates, 1991

References

24. Emery D. Berger, Kathryn S McKinley, Robert D Blumofe, Paul R.Wilson, *Hoard : A Scalable Memory Allocator for Multi-threaded Applications* ; The Ninth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-IX). Cambridge, MA, November (**2000**). Web site URL : http://www.hoard.org/

25. Marc Snir, Steve Otto, Steyen Huss-Lederman, David Walker and Jack Dongarra, (**1998**) *MPI-The Complete Reference: Volume 1, The MPI Core, second edition* [MCMPI-07].

26. William Gropp, Steven Huss-Lederman, Andrew Lumsdaine, Ewing Lusk, Bill Nitzberg, William Saphir, and Marc Snir (**1998**) *MPI-The Complete Reference: Volume 2, The MPI-2 Extensions*

27. A. Zomaya, editor. Parallel and Distributed Computing Handbook. McGraw-Hill, **(1996)**

28. OpenMP C and C++ Application Program Interface, Version 2.5 (**May 2005**)", From the OpenMP web site, URL **: http://www.openmp.org/**

29. Stokes, Jon 2002 Introduction to Multithreading, Super-threading and Hyper threading *Ars Technica*, October **(2002)**

30. Andrews Gregory R. 2000, Foundations of Multi-threaded, Parallel and Distributed Programming, Boston MA : Addison – Wesley (**2000)**

31. Deborah T. Marr , Frank Binns, David L. Hill, Glenn Hinton, David A Koufaty, J . Alan Miller, Michael Upton, "Hyperthreading, Technology Architecture and Microarchitecture", Intel (**2000-01)**

32. Shay Gai-On, & Markus Levy EEMBC,Measuring Multi-core Performance**, Computer** pp.99-102, Nov **2008**

33. EEMBC, the Embedded Microprocessor Benchmark Consortium URL : http://www.eembc.org/

34. MultiBench™ 1.0 Multicore Benchmark Software, URL : http://www.eembc.org/benchmark/multi_sl.php

Thank You

Any questions ?