Benchmark Results

周朝宜

cychou@nchc.org.tw

2006/10/03

Outline

- Motivation
- HPC Challenge Benchmark Suite
  - Software
  - Installation guide
  - Fine Tune
- Results Analysis
- Summary

Motivation

- **效能評估程式之目的**
  - Evaluate, Compare, Characterize platforms
  - **讓機器發揮最佳的效能表現**
    - **量測機器之運算效能**
    - **評比系統效能**
  - **幫助預估自己程式在各平台之效能表現**
    - **幫助瞭解機器特性以適合程式之最佳效益**
  - **採購、組裝與驗收電腦之參考**
    - **更新系統配備參考以及驗證更新後系統**
- **效能評估程式分類**
  - **瞭解硬體**(CPU、Memory,…)**效能：**SPEC, STREAM,…
  - **整體運算效能：**HPL, HINT,…
  - **特殊應用領域程式：** NPB,MM5,TPC,…

NCHC Application Suite**選擇原則**

- **本土性**
- **代表性**
- **移動性**
- **穩定性**
- **完整性**
- **適中性**

HPC Challenge Benchmark Suite

- **美國國防部先進研究計畫署高效能技術電腦計畫**
  - DARPA HPCS (Defense Advanced Research Projects Agency, High Productivity Computing Systems, http://www.highproductivity.org)
- To Augment HPL (Top500 List)
- 2010 Procurements
- Simplify codes to catch application area in Spatial and Temporal locality.

HPC Challenge Benchmark Suite Components

- HPL
- DGEMM
- FFT
- STREAM
- Random Access
- PTRANS
- b_eff

Scenarios

- Local
- Embarrassingly Parallel
- Global

- One Executing File
- Based on the largest HPL matrix

High-Performance Linpack Benchmark(HPL)

- solving the system of equations
- LU factorization with partial pivoting
- the operation count for the algorithm must be 2/3 n^3 + O(n^2) floating point operations

http://www.netlib.org/benchmark/hpl/

HPL

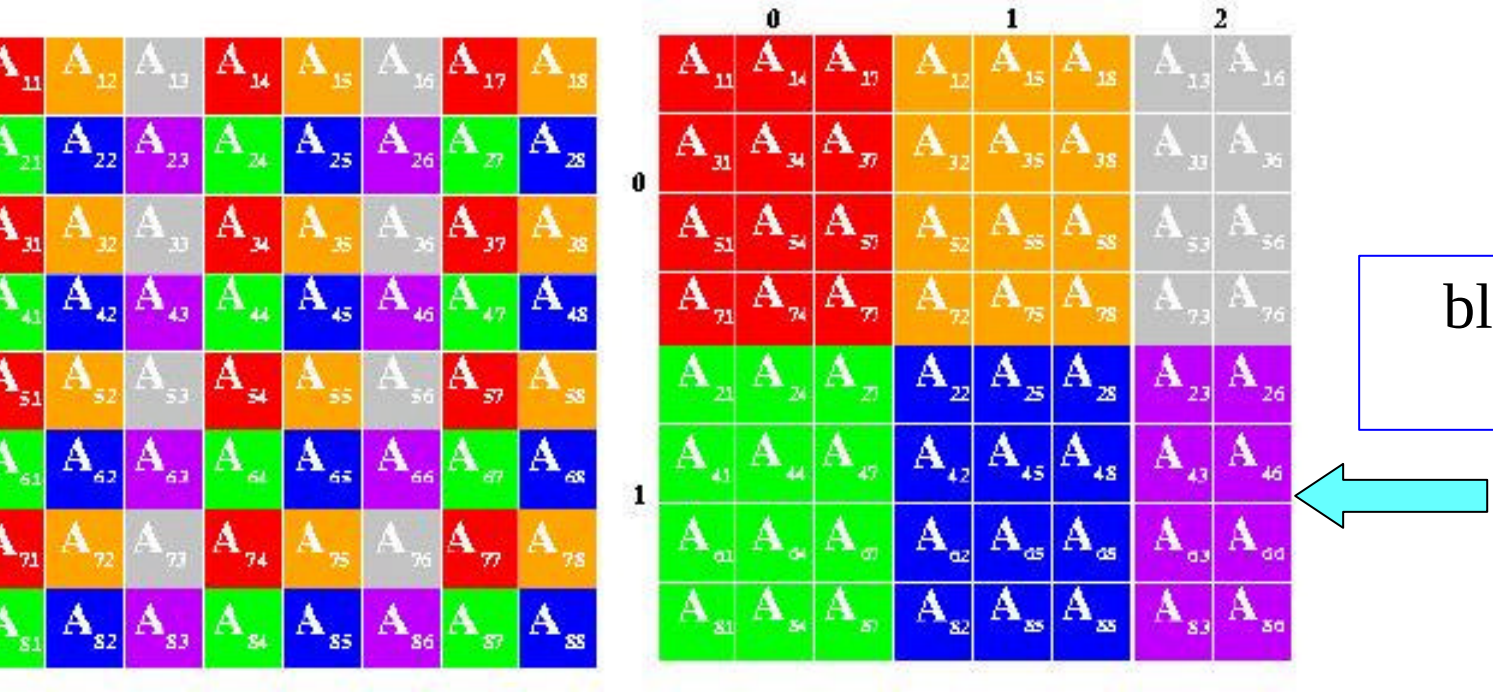
BLAS

HPL - A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers

Innovative Computing Laboratory
UNIVERSITY OF TENNESSEE
COMPUTER SCIENCE DEPARTMENT

Version 1.0a A. Petitet, R. C. Whaley, J. Dongarra, A. Cleary January 20, 2004 # Accesses

HPL Algorithm



block-cyclic scheme

Total Executing Time = 2 gam3 N^3 / (3 P Q) + beta N^2 (3 P + Q) / (2 P Q) + alpha N ((NB + 1) log(P) + P) / NB

HPCC Software

- **FFT**
  - One dimensional Discrete Fourier Transform

$$Z_k \leftarrow \sum_j^m z_j e^{-2\pi i \frac{jk}{m}}; \quad 1 \le k \le m$$

  - Operation Count is $5\, m\, log_2 m$
- **STREAM**
  - Measure sustainable Memory Bandwidth in GB/s

$$\text{COPY:}\quad c \leftarrow a \qquad \text{SCALE:}\quad b \leftarrow \alpha c$$

$$\text{ADD:}\quad c \leftarrow a + b \qquad \text{TRIAD:}\quad a \leftarrow b + \alpha c$$

- **Random Access**
  - Measure the rate of integer updates to random locations in GUPS
  - The MPI version generates the updating sequence locally and then distributes it using all-to-all collective communication.

$$x \leftarrow f(x) \qquad f : x \mapsto (x \oplus a_i); \qquad a_i - \text{pseudo-random sequence}$$

where $f : \mathbf{Z}_2^m \to \mathbf{Z}_2^m; \quad x \in \mathbf{Z}_2^m$

Benchmark for Massage Passing on HPCC

- **PTRANS**
  - Parallel Matrix Transpose, in GB/s

$$A \leftarrow A^T + B$$

  - Exercises the communications where pairs of processors exchange large messages simultaneous.
  - Test of the total communications capacity of the system interconnect.
- **b_eff**
  - Ping Pong
    - Maximal Latency in μs, Minimal Bandwidth in GB/s
    - 8bytes for Latency; 2MB for Bandwidth
  - Ring
    - Processor Order
      - **Naturally** model orders the processor in the ranks of **MPI_COMM_WORLD**
      - **Randomly** model is used the geometric mean of 10 different Randomly Chosen Ordering
    - Take the best in the following MPI Implement
      - MPI Standard non-blocking receive and send
      - Two calls to MPI_Sendrecv for both directions

Installation of HPCC

- **Download this benchmark suite**
  - hpcc-1.0.0.tar.gz on http://icl.cs.utk.edu/hpcc/
- **Setup**
  - unpack (eg, tar –zxvf hpcc-1.0.0.tar.gz)
  - Create a file Make.<arch> in the **hpl** directory, such as ***Make.Gnu_ATLAS*** (**可以參考** setup **目錄下的檔案**)
  - **修改幾個環境變數設定**, **例如**:

    MPdir = MPI directories
    LAdir = /work/chou/ATLAS/lib/Linux_P4ESSE2
    LAinc =
    LAlib = -L$(LAdir) -lcblas –latlas
    HPL_OPTS = -DHPL_CALL_CBLAS
    CC = /opt/mpich/gnu/bin/mpicc
- **Generate the executing file**
  - Return the top-level directory, type "make arch=<arch>" , such as, ***make arch= Gnu_ATLAS*** , and then one obtain the executing file, called “**hpcc**” in this directory.

Input File for HPCC

- hpccinf.txt

```
40000       Ns
1           # of NBs
88          NBs
0           PMAP process mapping (0=Row-,1=Column-major)
1           # of process grids (P x Q)
4           Ps
4           Qs
16.0        threshold
1       # of panel fact
2       PFACTs (0=left, 1=Crout, 2=Right
```

```
##### This line (no. 32) is ignored (it serves as a separator). ######
0                       Number of additional problem sizes for PTRANS
1200 10000 30000                values of N
0                       number of additional blocking sizes for PTRANS
40 9 8 13 13 20 16 32 64        values of NB
```

Output File for HPCC

- hpccoutf.txt

N : 40000
NB : 88
PMAP : Row-major process mapping
P : 4
Q : 4
PFACT : Right

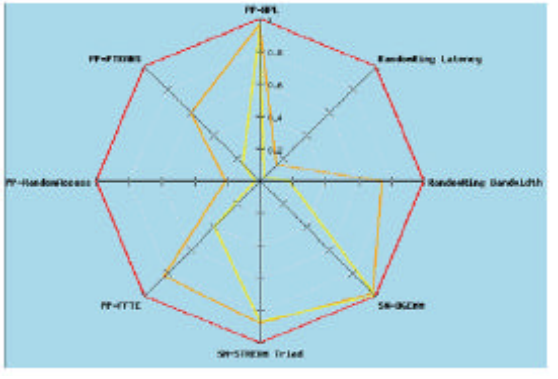
RFACT : Crout
BCAST : 1ringM
DEPTH : 1
SWAP : Mix (threshold = 64)
L1 : transposed form
U : transposed form
EQUIL : yes
ALIGN : 8 double precision words

```
================================================================================
T/V                N    NB     P     Q               Time             Gflops
--------------------------------------------------------------------------------
WR11C2R4       40000    88     4     4             877.09          4.865e+01
--------------------------------------------------------------------------------
||Ax-b||_oo / ( eps * ||A||_1  * N        ) =        0.0195192 ..... PASSED
||Ax-b||_oo / ( eps * ||A||_1  * ||x||_1  ) =        0.0178475 ..... PASSED
||Ax-b||_oo / ( eps * ||A||_oo * ||x||_oo ) =        0.0035047 ..... PASSED
================================================================================
gin of StarRandomAccess section.
in table size   = 2^26 = 67108864 words
mber of updates = 268435456
U time used  = 74.920000 seconds
al time used = 75.050249 seconds
003576743 Billion(10^9) Updates    per second [GUP/s]
und 0 errors in 67108864 locations (passed).
de(s) with error 0
nimum GUP/s 0.003510
erage GUP/s 0.003557
ximum GUP/s 0.003577
rrent time (1150139847) is Tue Jun 13 03:17:27 2006
```

MPI_Wtick=9.999999e-07
HPL_Tflops=0.0486484
HPL_time=877.09
HPL_eps=1.11022e-16
HPL_RnormI=8.77558e-10
HPL_Anorm1=10123.8
HPL_AnormI=10096.4
HPL_Xnorm1=43746.6
HPL_XnormI=5.58446

Analysis on HPCC Results

- Global Merit
  - G-HPL, G-PTRANS, G-Random Access, G-FFTE, G-STREAM,…
- Per Processor Merit
  - Kiviat Charts

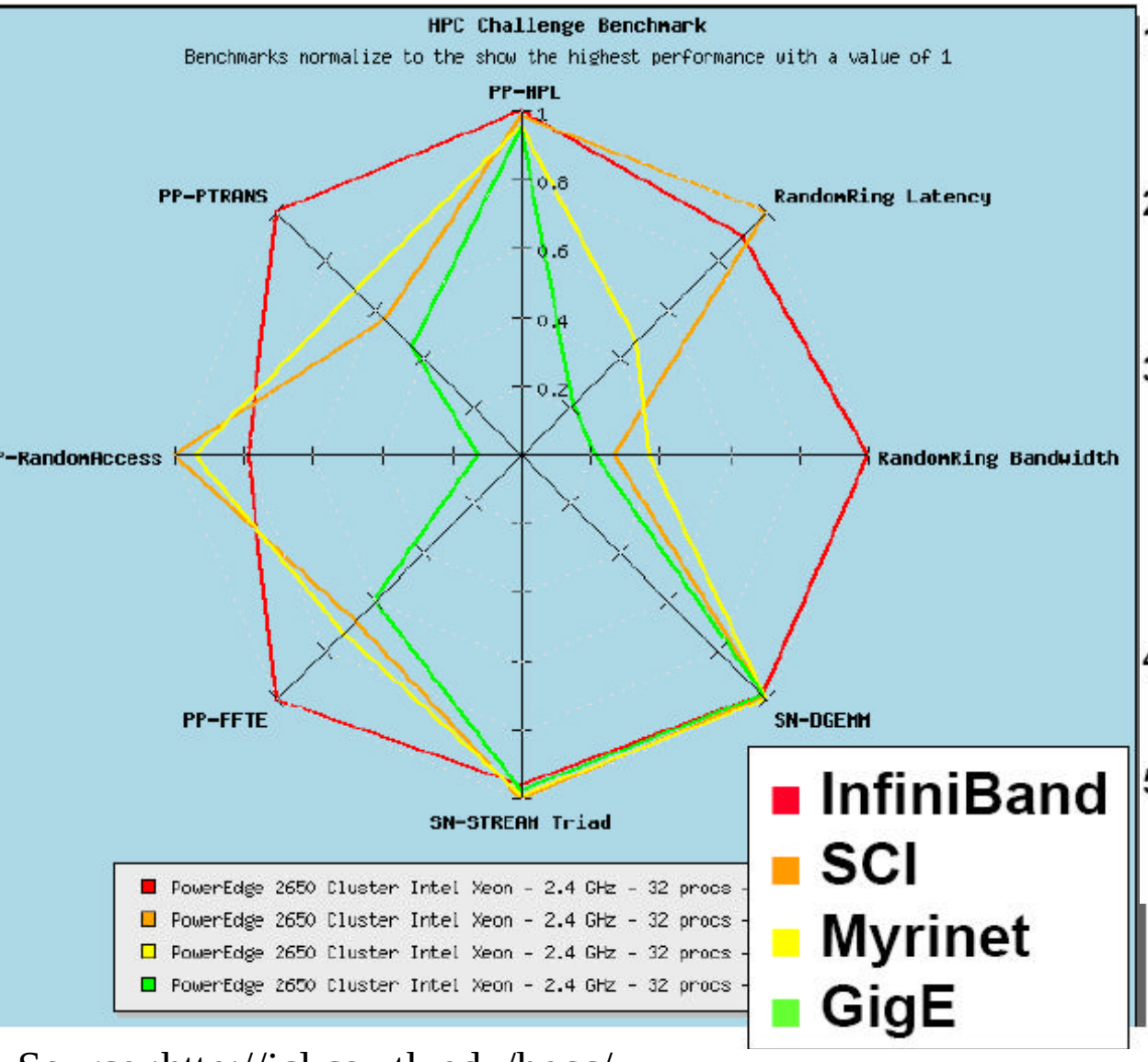


Computer	HPL	PTRANS	STREAM - single CPU				*STREAM				RandomAccess			Latency		Bandwidth		
			Copy	Scale	Add	Triad	Copy	Scale	Add	Triad	Single CPU	*	MPI	Ping Max.	Random Ring	Ping Min.	Random Ring	Natural Ring
System/Procs	TFlop/s	GB/s	GB/s	GB/s	GB/s	GB/s	GB/s	GB/s	GB/s	GB/s	Gup/s	Gup/s	Gup/s	usec	usec	GB/s	GB/s	GB/s

Source://icl.cs.utk.edu/hpcc/

System Information System - Processor	Speed	Count	Tds	Proc	G-HPL TFlop/s	G-PTRANS GB/s	G-Random Access Gup/s	G-FFTE GFlop/s	G-STREAM Triad GB/s	EP STREAM Triad GB/s	EP DGEMM GFlop/s	Random Ring Bandwidth GB/s	Random Ring Latency usec
Cray XT3 AMD Opteron	2.4GHz	5200	1	5200	20.527	874.899	0.268583	644.73	26020.8	5.004	4.395	0.14682	25.8
Cray mfeg8 X1E	1.13GHz	248	1	248	3.3889	66.01	1.85475	-1	3280.9	13.229	13.564	0.29886	14.58
Cray XT3 AMD Opteron	2.6GHz	4096	1	4096	16.9752	302.979	0.533072	905.57	20656.5	5.043	4.782	0.16896	9.44
NEC SX-7	0.552GHz	32	16	2	0.2174	16.34	0.000178	1.34	984.3	492.161	140.636	8.14753	4.85
NEC SX-8/6 SX-8	2GHz	6	1	6	0.0918	25.183	0.000769	3.19	370.6	61.773	15.944	13.5473	3.02
IBM pSeries 655 Power 4+	1.7GHz	256	4	64	1.0744	23.721	0.005502	10.46	411.7	6.433	17.979	0.72395	8.34
PathScale Inc. AMD Opteron	2.6GHz	32	1	32	0.1258	6.719	0.030367	10.35	134.3	4.197	4.775	0.26531	1.31

Kiviat diagram on different networks



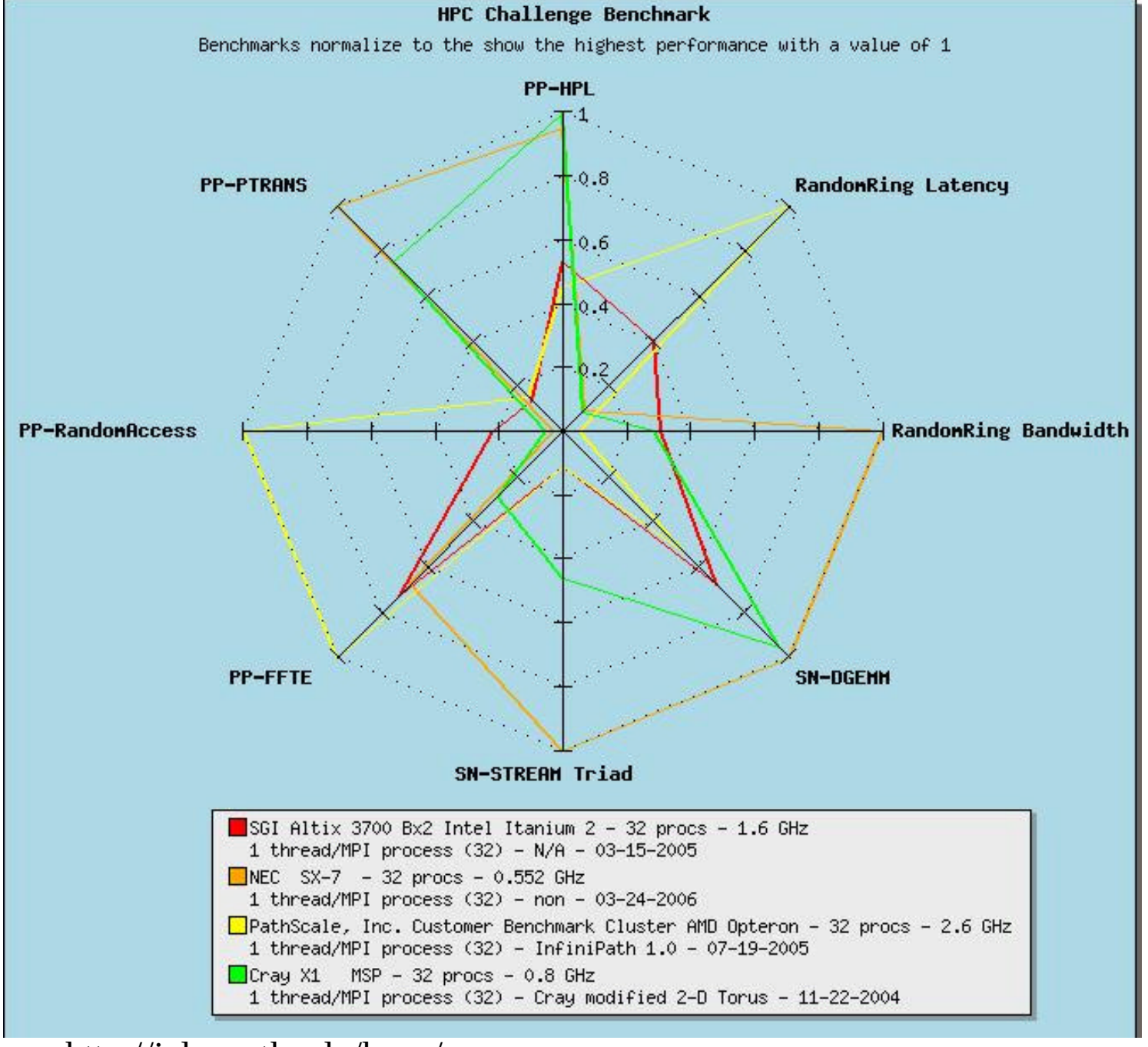
1. RandomRing Bandwidth
   InfiniBand has significantly greater bandwidth than other technologies
2. RandomRing Latency
   InfiniBand and SCI have significantly lower latencies than other technologies
3. STREAM, DGEMM, and HPL
   Interconnect technology doesn't matter
   STREAM and DGEMM have communications
   HPL scales well with respec to communications
4. RandomAccess
   Interconnect technology do matter! Latency sensitive
5. PTRANS and FFTE
   Interconnect technology do matter Bandwidth sensitive

Source :http://icl.cs.utk.edu/hpcc/

Case Study

System	CPU*32	OS	BLAS	MPI	network	Compiler
Cray X1 MSP	0.8 GHz	Unicos/MP 2.4	Libsci5.2	MPT2.4	2D Torus	Prg Env 5.3
NEC SX-7	0.55 GHz	SUPER-UX 13.1	Mathkeisan 1.3.0	Sx 7.06	-	-
SGI Altix	Itanium 2 1.6GHz	SGI ProPack	Intel MKL 7,2 DGEMM, SCSL 1.5.1.1	SGI MPT1.12	-	Intel 9.0
PathScale	AMD Opteron 2.6GHz	Fedora Core 3	Goto opt64r0.99	PathScale MPI 1.0	InfiniPath 1.0	PathScale EKO v.2.2

System	G-HPL	G-PTRANS	G-Random Access	G-FFT	EP-STREAM Sys	EP-DGEMM	Randomly Ring	
							BW	Latency
	GFLOPS	GB/s	Mup/s	GFLOPS	GB/s	GFLOPS	GB/s	μs
Cray X1 MSP	277	32.66	1.66	2.96	476	264	1.41	14.94
NEC SX-7	262	43.59	0.78	6.96	873	241	4.96	14.15
SGI Altix	147	6.05	6.60	7.48	64	192	1.52	3.26
PathScale	126	6.72	30.37	10.35	134	153	0.27	1.31


HPC Challenge Benchmark
Benchmarks normalize to the show the highest performance with a value of 1
PP-HPL
RandomRing Latency
RandomRing Bandwidth
SN-DGEMM
SN-STREAM Triad
PP-FFTE
PP-RandomAccess
PP-PTRANS
1
0.8
0.6
0.4
0.2
SGI Altix 3700 Bx2 Intel Itanium 2 - 32 procs - 1.6 GHz
1 thread/MPI process (32) - N/A - 03-15-2005
NEC SX-7 - 32 procs - 0.552 GHz
1 thread/MPI process (32) - non - 03-24-2006
PathScale, Inc. Customer Benchmark Cluster AMD Opteron - 32 procs - 2.6 GHz
1 thread/MPI process (32) - InfiniPath 1.0 - 07-19-2005
Cray X1 MSP - 32 procs - 0.8 GHz
1 thread/MPI process (32) - Cray modified 2-D Torus - 11-22-2004

R_{max} and Efficiency

System	Peak GFLOPS	G-HPL GFLOPS	Efficiency %	(flops/cycle) ×GHz ×procs
Cray X1 MSP	409.60	277	68	16 ×0.8 ×32
NEC SX-7	282.56	262	93	16 ×0.552 ×32
SGI Altix	204.80	147	72	4 ×1.6 ×32
PathScale	166.40	126	76	2 ×2.6 ×32

System	EP-DGEMM GFLOPS	Efficiency %
Cray X1 MSP	264	64
NEC SX-7	241	85
SGI Altix	192	94
PathScale	153	92

Summary

- HPL is the Linpack toward peak performance benchmark. [Jack J. Dongarra]
  - The benchmark is affected only slightly by cache size, memory speed and network performance. [Dr. Pase]
- The HPC Challenge Benchmark Suite
  - This benchmark test suite stresses not only the Processor, but Memory system and the Interconnect. [Jack J. Dongarra]