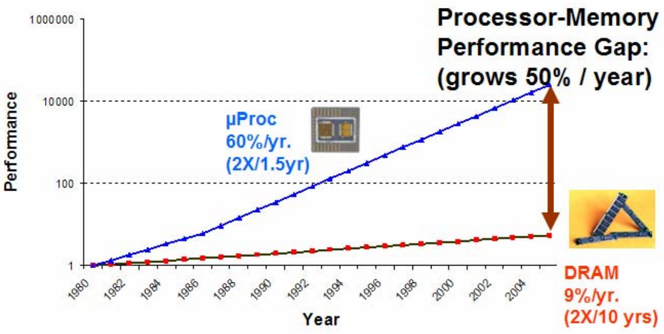
HPC Challenge Benchmarks and the TOP500

Jack Dongarra (UTK/ORNL)

At The Time The Linpack Benchmark Was Created …

- If we think about computing in late 70's
- Perhaps the LINPACK benchmark was a reasonable thing to use.
- Memory wall, not so much a wall but a step.
- In the 70's, things were more in balance
  - The memory kept pace with the CPU
    - n cycles to execute an instruction, n cycles to bring in a word from memory

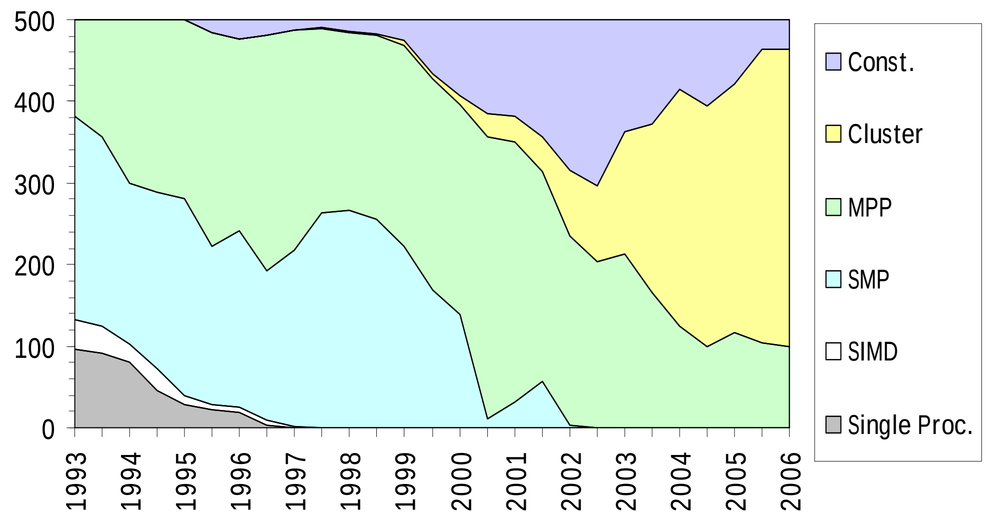


- Showed compiler optimization
- Today provides a historical base of data

Many Changes

- Many changes in our hardware over the past 30 years
  - Superscalar, Vector, Distributed Memory, Shared Memory, Multicore, …

Top500 Systems/Architectures



- While there has been some changes to the Linpack Benchmark not all of them reflect the advances made in the hardware.

- Today's memory hierarchy is much more complicated.

Motivation for Additional Benchmarks

Linpack Benchmark

- Good
  - One number
  - Simple to define & easy to rank
  - Allows problem size to change with machine and over time
  - Stresses the system with a long running job
- Bad
  - Emphasizes only "peak" CPU speed and number of CPUs
  - Does not stress local bandwidth
  - Does not stress the network
  - Does not test gather/scatter
  - Ignores Amdahl's Law (Only does weak scaling)
- Ugly
  - Benchmarketeering hype

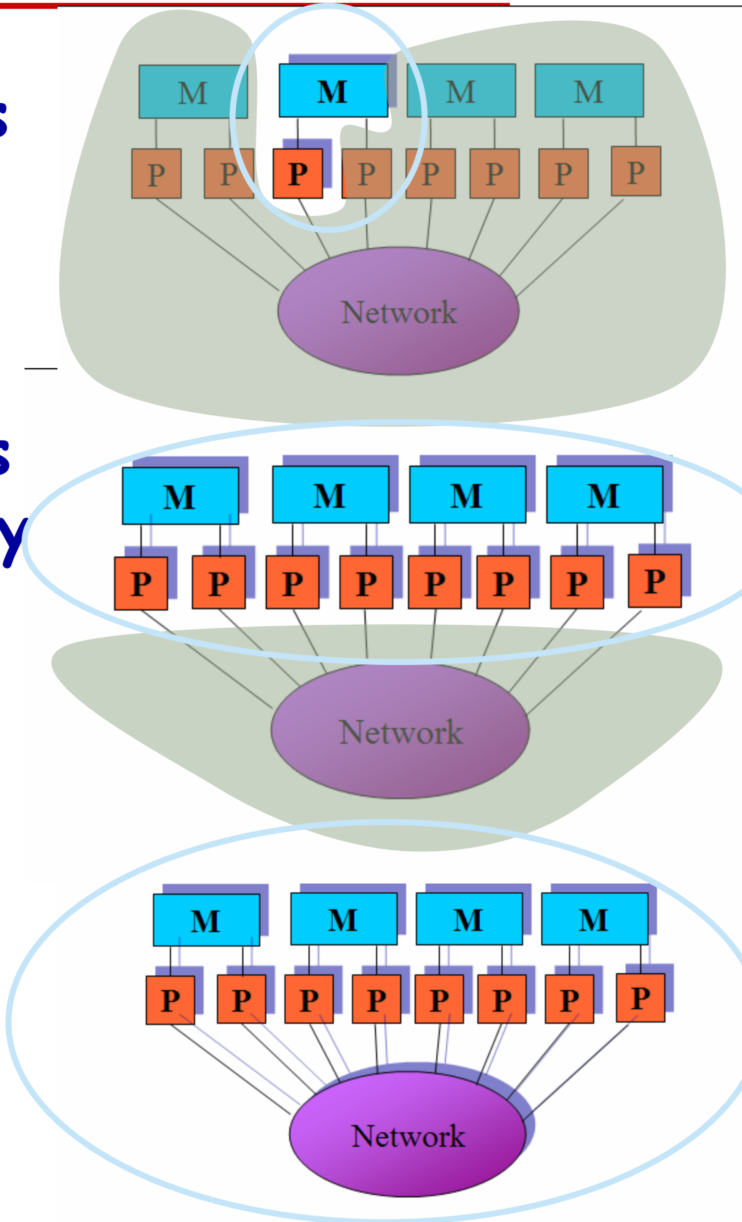
- Perhaps there was a time when this was adequate.
- From Linpack Benchmark and Top500: "no single number can reflect overall performance"
- Clearly need something more than Linpack
- HPC Challenge Benchmark
  - Test suite stresses not only the processors, but the memory system and the interconnect.
  - The real utility of the HPCC benchmarks are that architectures can be described with a wider range of metrics than just Flop/s from Linpack.

Goals HPC Challenge Benchmark

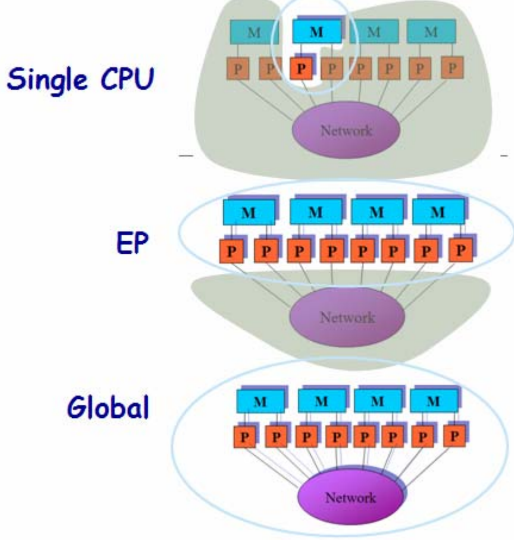
- Stress CPU, memory system, interconnect
- To *complement* the Top500 list
- To provide benchmarks that *bound* the performance of many real applications as a function of memory access characteristics
  - e.g., spatial and temporal locality
- Allow for optimizations
  - Record effort needed for tuning
  - Base run requires MPI and BLAS
- Provide verification of results
- Archive results

Tests on Single Processor and System

- Local - only a single processor is performing computations.
- Embarrassingly Parallel - each processor in the entire system is performing computations but they do no communicate with each other explicitly.
- Global - all processors in the system are performing computations and they explicitly communicate with each other.



HPC Challenge Benchmark



Consists of basically 7 benchmarks;

- Think of it as a framework or harness for adding benchmarks of interest.

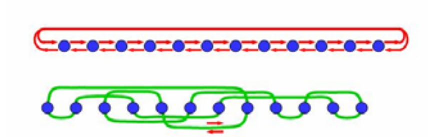
1. HPL (LINPACK) — MPI Global (Ax = b)
2. STREAM — Local; single CPU
   *STREAM — Embarrassingly parallel

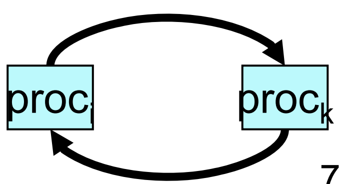
name	kernel	bytes/iter	FLOPS/iter
COPY:	a(i) = b(i)	16	0
SCALE:	a(i) = q*b(i)	16	1
SUM:	a(i) = b(i) + c(i)	24	1
TRIAD:	a(i) = b(i) + q*c(i)	24	2

3. PTRANS ($A \leftarrow A + B^T$) — MPI Global
4. RandomAccess — Local; single CPU
   *RandomAccess — Embarrassingly parallel
   RandomAccess — MPI Global

Random integer read; update; & write

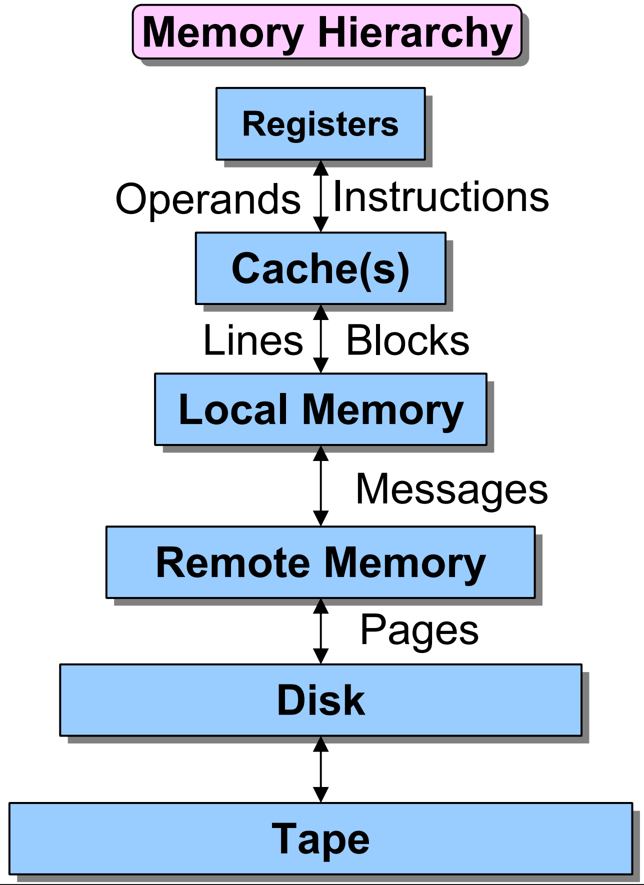
5. BW and Latency – MPI





6. FFT - Global, single CPU, and EP
7. Matrix Multiply – single CPU and EP

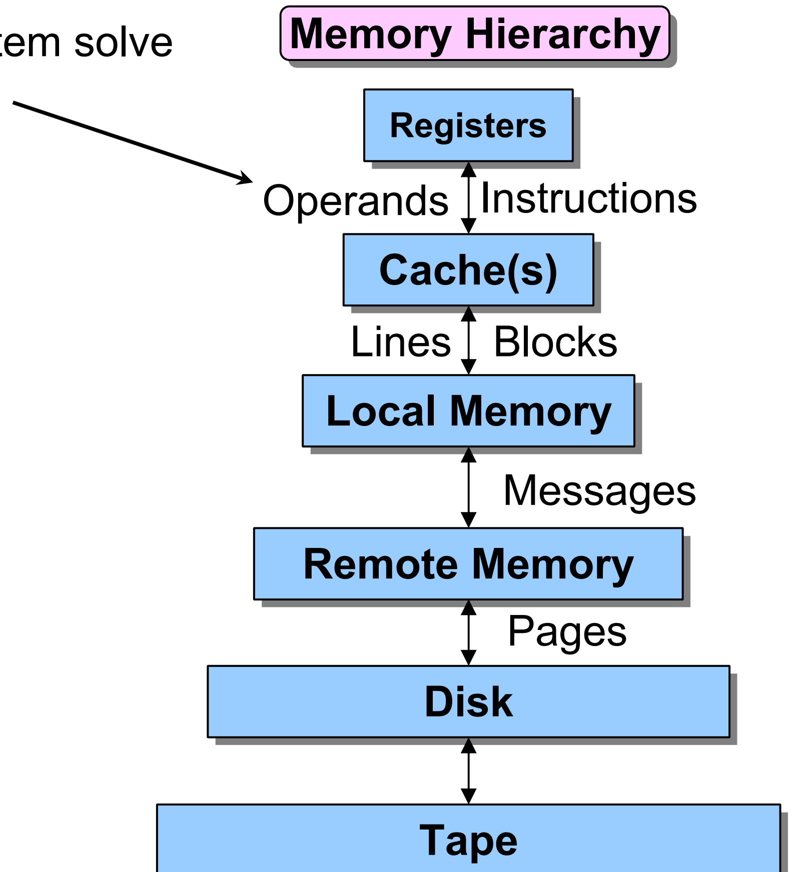
HPCS Performance Targets



- HPCC was developed by HPCS to assist in testing new HEC systems
- Each benchmark focuses on a different part of the memory hierarchy
- HPCS performance targets attempt to
  - Flatten the memory hierarchy
  - Improve real application performance
  - Make programming easier

HPCS Performance Targets

- LINPACK: linear system solve
  Ax = b



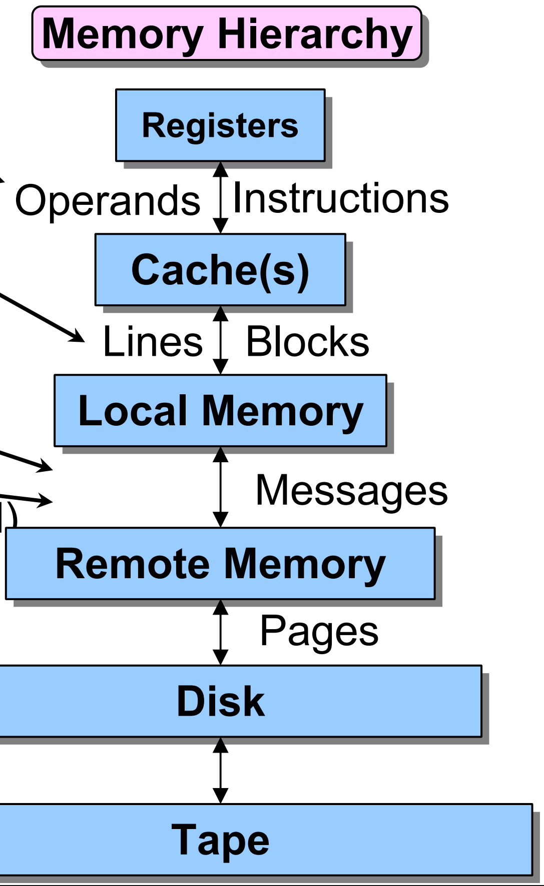
- **HPCC was developed by HPCS to assist in testing new HEC systems**
- **Each benchmark focuses on a different part of the memory hierarchy**
- **HPCS performance targets attempt to**
  - **Flatten the memory hierarchy**
  - **Improve real application performance**
  - **Make programming easier**

HPCS Performance Targets

- LINPACK: linear system solve
  $Ax = b$
- STREAM: vector operations
  $A = B + s * C$
- FFT: 1D Fast Fourier Transform
  $Z = fft(X)$
- RandomAccess: integer update
  $T[i] = XOR(T[i], rand)$

HPC Challenge

HPCC

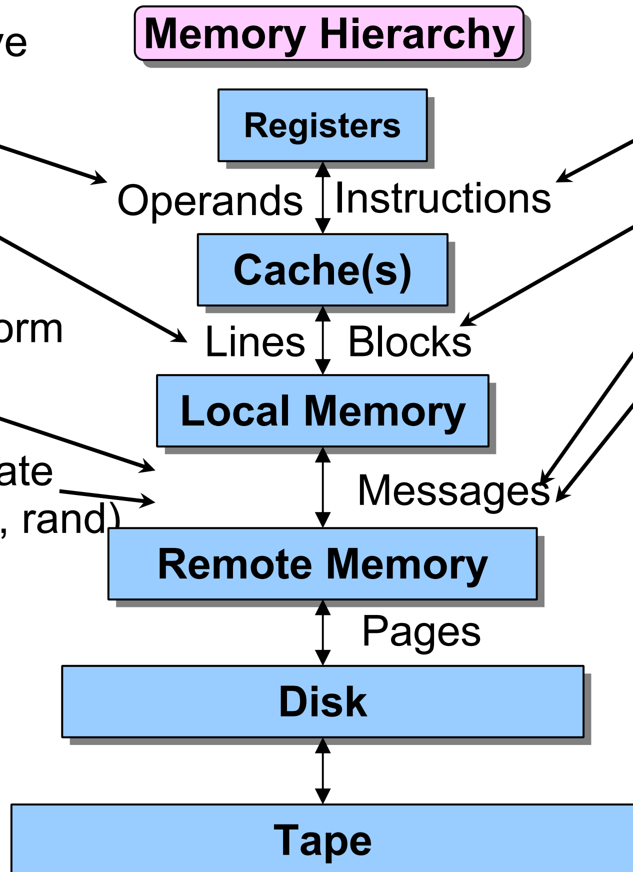


- **HPCC was developed by HPCS to assist in testing new HEC systems**
- **Each benchmark focuses on a different part of the memory hierarchy**
- **HPCS performance targets attempt to**
  - **Flatten the memory hierarchy**
  - **Improve real application performance**
  - **Make programming easier**

HPCS Performance Targets

- LINPACK: linear system solve
  Ax = b
- STREAM: vector operations
  A = B + s * C
- FFT: 1D Fast Fourier Transform
  Z = fft(X)
- RandomAccess: integer update
  T[i] = XOR(T[i], rand)

HPC Challenge

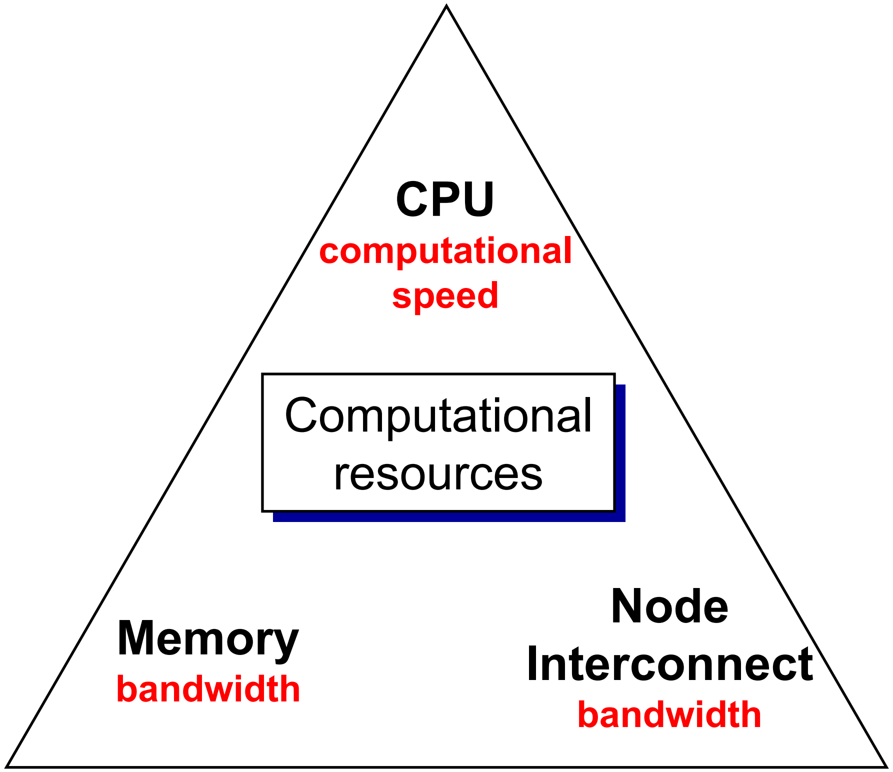


Max	Relative
2 Pflop/s	8x
6.5 Pbyte/s	40x
0.5 Pflop/s	200x
64000 GUPS	2000x

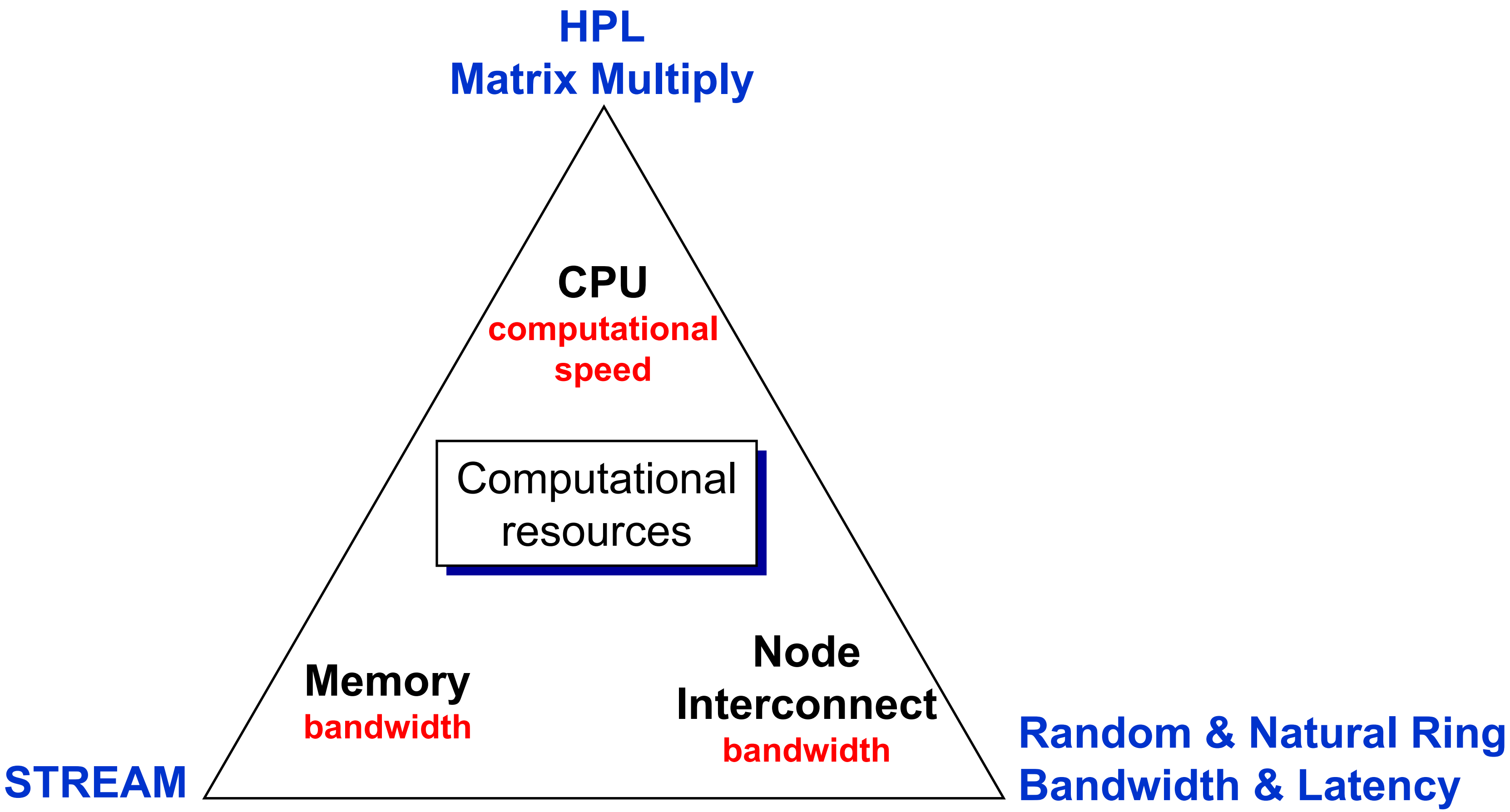
Performance Targets

- **HPCC was developed by HPCS to assist in testing new HEC systems**
- **Each benchmark focuses on a different part of the memory hierarchy**
- **HPCS performance targets attempt to**
  - **Flatten the memory hierarchy**
  - **Improve real application performance**
  - **Make programming easier**

Computational Resources and HPC Challenge Benchmarks



Computational Resources and HPC Challenge Benchmarks



How Does The Benchmarking Work?

- Single program to download and run
  - Simple input file similar to HPL input
- Base Run and Optimization Run
  - Base run must be made
    - User supplies MPI and the BLAS
  - Optimized run allowed to replace certain routines
    - User specifies what was done
- Results upload via website
- html table and Excel spreadsheet generated with performance results
  - Intentionally we are not providing a single figure of merit (no over all ranking)
- Goal: no more than 2 X the time to execute HPL.

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

HPC CHALLENGE

HPCS

- Home
- Rules
- News
- Download
- FAQ
- Links
- Collaborators
- Sponsors
- Upload
- Kiviat Diagram
- Results

HPC Challenge Benchmark

The HPC Challenge benchmark consists of basically 7 benchmarks:

1. HPL - the Linpack TPP benchmark which measures the floating point rate of execution for solving a linear system of equations.
2. DGEMM - measures the floating point rate of execution of double precision real matrix-matrix multiplication.
3. STREAM - a simple synthetic benchmark program that measures sustainable memory bandwidth (in GB/s) and the corresponding computation rate for simple vector kernel.
4. PTRANS (parallel matrix transpose) - exercises the communications where pairs of processors communicate with each other simultaneously. It is a useful test of the total communications capacity of the network.
5. RandomAccess - measures the rate of integer random updates of memory (GUPS).
6. FFTE - measures the floating point rate of execution of double precision complex one-dimensional Discrete Fourier Transform (DFT).
7. Communication bandwidth and latency - a set of tests to measure latency and bandwidth of a number of simultaneous communication patterns; based on b_eff (effective bandwidth benchmark).

Condensed Results - Base Runs Only - 114 Systems - Generated on Tue Nov 14 17:13:55 2006

System Information System - Processor - Speed - Count - Threads - Processes					G-HPL	G-PTRANS	G-Random Access	G-FFTE	EP-STREAM Sys	EP-STREAM Triad	EP-DGEMM	RandomRing Bandwidth	RandomRing Latency
MA/PT/PS/PC/TH/PR/CM/CS/IC/IA/SD					TFlop/s	GB/s	Gup/s	GFlop/s	GB/s	GB/s	GFlop/s	GB/s	usec
Cray Inc. Red Storm/XT3 AMD Opteron	2.4GHz	12960	1	25920	91.0350000	2356.9700	1.7401500	1554.0700	54840.499	2.1158	4.39939	0.05911	16.29
IBM Blue Gene/L PowerPC 440	0.7GHz	65536	1	65536	80.6830000	339.2840	0.0657312	2178.1100	53555.888	0.8172	1.85619	0.01084	8.84
IBM p5-575 Power5	1.9GHz	10240	1	10240	57.8670000	553.0090	0.1693440	842.5000	55184.179	5.3891	7.08562	0.11015	118.59
IBM p5-575 Power5	1.9GHz	8192	1	8192	45.7019000	2626.1700	0.3239760	908.6920	44455.936	5.4268	7.06423	0.08871	11.05
Cray Inc. XT3 Dual-Core AMD Opteron	2.6GHz	10404	1	10404	43.4033000	778.3850	0.8235630	1107.2100	25774.557	2.4774	4.78995	0.06937	14.32
IBM Blue Gene/L PowerPC 440	0.7GHz	65536	1	65536	37.3540000	4665.9100	0.1648600	1762.8200	62889.787	0.9596	2.47017	0.01039	8.62
IBM p5-575 Power5	1.9GHz	8192	2	8192	33.3175000	575.8230	0.2066390	966.6680	43802.460	5.3470	6.08616	0.07698	51.99
Cray Inc. XT3 AMD Opteron	2GHz	10350	1	10350	32.9865000	1813.0600	1.0176500	1118.2900	43581.780	4.2108	3.66719	0.16188	10.32
IBM Blue Gene/L PowerPC 440	0.7GHz	32768	1	32768	31.2581000	87.7818	0.2780090	1112.8100	29913.678	0.9129	2.17447	0.01197	9.51
Cray Inc. XT3 AMD Opteron	2.4GHz	5200	1	5200	20.5270000	874.8990	0.2685830	644.7300	26020.800	5.0040	4.39535	0.14682	25.80
Cray Inc. XT3 AMD Opteron	2.4GHz	5208	1	5208	20.4086000	944.2270	0.6724120	761.7290	24268.447	4.6598	4.41173	0.20636	9.20
Cray Inc. XT3 AMD Opteron	2.6GHz	4096	1	4096	16.9752000	302.9790	0.5330720	905.5690	20656.456	5.0431	4.78166	0.16896	9.44
Cray Inc. XT3 AMD Opteron	2.6GHz	4128	1	4128	16.6421000	674.7860	0.6767580	821.6770	19295.676	4.6743	4.75946	0.22245	8.23
Cray Inc. XT3 AMD Opteron	2.4GHz	3744	1	3744	14.7040000	608.5060	0.2202960	417.1720	18146.382	4.8468	4.41330	0.16164	25.32
Cray Inc. X1 Cray E	1.13GHz	1008	1	1008	12.0263000	108.0190	0.0861199	82.3884	15522.091	15.3989	14.50000	0.15667	16.30
HP XC Intel Itanium 2	1.6GHz	2048	1	2048	10.5616000	306.4190	0.2633320	339.8360	4000.031	1.9531	6.26484	0.15920	6.04
System Information System - Processor - Speed - Count - Threads - Processes					G-HPL	G-PTRANS	G-Random Access	G-FFTE	EP-STREAM Sys	EP-STREAM Triad	EP-DGEMM	RandomRing Bandwidth	RandomRing Latency
MA/PT/PS/PC/TH/PR/CM/CS/IC/IA/SD					TFlop/s	GB/s	Gup/s	GFlop/s	GB/s	GB/s	GFlop/s	GB/s	usec
SGI Columbia 2048 Intel Itanium 2	1.6GHz	2024	1	2024	9.3196500	18.1901	0.0491621	45.7760	3998.493	1.9755	6.23637	0.12271	6.98
NEC SX-8	2GHz	576	1	576	8.0085800	312.7070	0.0193617	160.9480	23555.750	40.8954	15.22320	0.82924	22.27
IBM p655 Power4+	1.5GHz	2048	1	2048	6.2094300	103.8040	0.1417230	156.5440	3631.636	1.7733	3.99073	0.06960	13.36
SGI Altix 3700 Bx2 Intel Itanium 2	1.6GHz	1008	1	1008	5.1383200	105.6660	0.0325982	15.6619	1907.509	1.8924	5.88404	0.20288	6.82
HP XC Intel Itanium 2	1.6GHz	1024	1	1024	4.8532700	154.1290	0.1615660	184.1270	2002.422	1.9555	6.24250	0.15978	5.94
Cray Inc. XT3 AMD Opteron	2.6GHz	1100	1	1100	4.7823400	217.9230	0.1370020	266.6600	5274.698	4.7952	4.81050	0.28638	25.94
Cray Inc. XT3 AMD Opteron	2.6GHz	1100	1	1100	4.7276600	253.3460	0.3035680	328.2860	5161.134	4.6919	4.77440	0.39964	7.29
IBM P5 P575+ Power5+	1.9GHz	512	1	512	3.4100800	219.5560	0.0848539	52.2538	2997.745	5.8550	7.62342	0.05243	36.14
Cray Inc. X1E Cray X1 MSP	1.13GHz	252	1	252	3.1940900	85.2040	0.0148684	15.5352	2439.985	9.6825	14.18470	0.36024	14.93
IBM p655 Power4+	1.5GHz	1024	1	1024	3.1102600	51.4248	0.0983533	82.1055	1733.284	1.6927	3.72701	0.09024	10.13

Future Directions

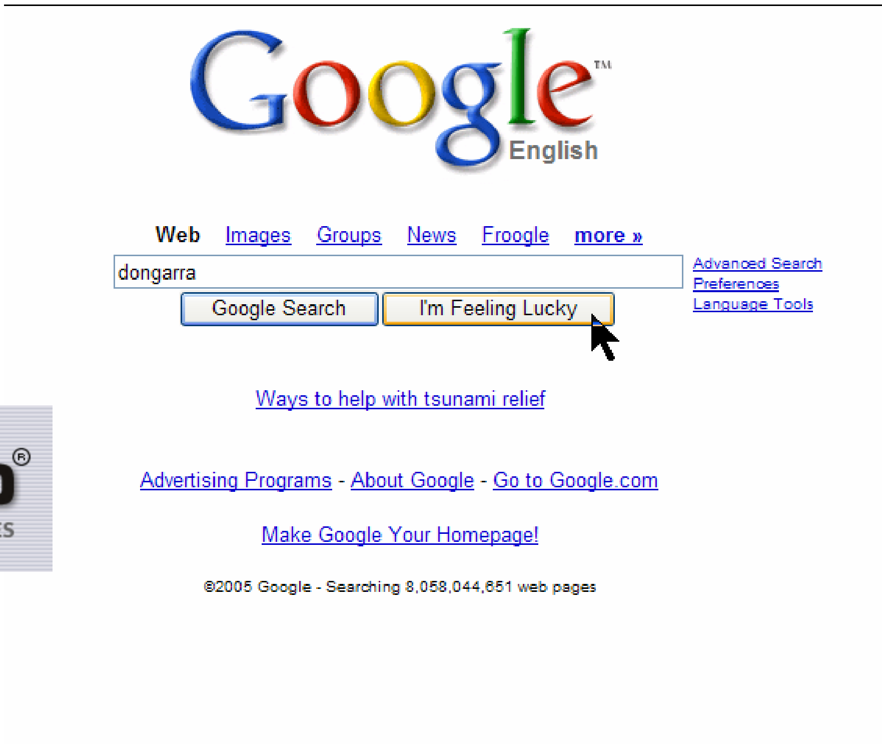
- Constructing a framework for benchmarks
- Developing machine signatures
- Plans are to expand the benchmark collection
  - Sparse matrix operations

- Port to new systems
- Provide more implementations
  - Languages (Fortran, UPC, Co-Array)
  - Environments
  - Paradigms

HPC CHALLENGE HPCS

Collaborators

- HPC Challenge
  - Piotr Łuszczek, U of Tennessee
  - David Bailey, NERSC/LBL
  - Jeremy Kepner, MIT Lincoln Lab
  - Bob Lucas, ISI/USC
  - Rusty Lusk, ANL
  - John McCalpin, IBM, Austin
  - Rolf Rabenseifner, HLRS Stuttgart
  - Daisuke Takahashi, Tsukuba, Japan
  - Jeff Vetter, ORNL



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

Office of Science
U.S. Department of Energy