Benchmarking

Performance required on range of applications

Value of computer dependent on context

Context varies by workload application time

Evaluation valid for one site may not be valid for another

There is no universal metric

LINPACK
Livermore Loops
NAS Parallel Benchmarks
Perfect
SPEC
ParkBench

Twelve Ways to Fool the Masses
D. Bailey, NASA Ames

1. **Quote 32-bit performance results, not 64-bit results.**
2. **Present inner kernel performance figures as the performance of the entire application.**
3. **Employ assembly code and other low-level language constructs.**
4. **Scale up the problem size with the number of processors.**
5. **Quote performance results projected to a full system.**
6. **Compare parallel results against scalar, unoptimized code on conventional systems.**
7. **Compare with an old code on an obsolete system.**

Twelve Ways to Fool the Masses (Contd.)

8. Base Mflop/s operation counts on the parallel implementation instead of on the best sequential implementation.
9. Quote performance in processor utilization, parallel speedups or peak Mflop/s.
10. Mutilate the algorithm used in the parallel implementation to match the architecture.
11. Measure parallel run times on a dedicated system, but measure conventional run times in a busy environment.
12. If else fails, show pretty pictures and animated videos, and don't talk about performance.

LINPACK BENCHMARK

- Linpack Benchmark is really three benchmarks
  - Run the Benchmark code, all Fortran, no changes allowed to the software. User only to supply timing function. Benchmark runs the LINPACK routines DGEFA and DGESL which calls the BLAS (DAXPY). Time to factor and solve a system of equations of order 100.
  - Run the Benchmark code, solve a system of equations of order 1000. User can supply the software to solve the problem.
  - Run the largest system of equations problem on the machine. User can supply the software to solve the problem.

Interested in ...

R_{max}	Gflop/s for largest problem
N_{max}	size of largest problem
$N_{1/2}$	size where half the R_{max} achieved
R_{peak}	theoretical peak performance

Must get the "correct results".

The results were checked for accuracy by calculating a residual for the problem $||Ax - b||/(||A||||x||)$.

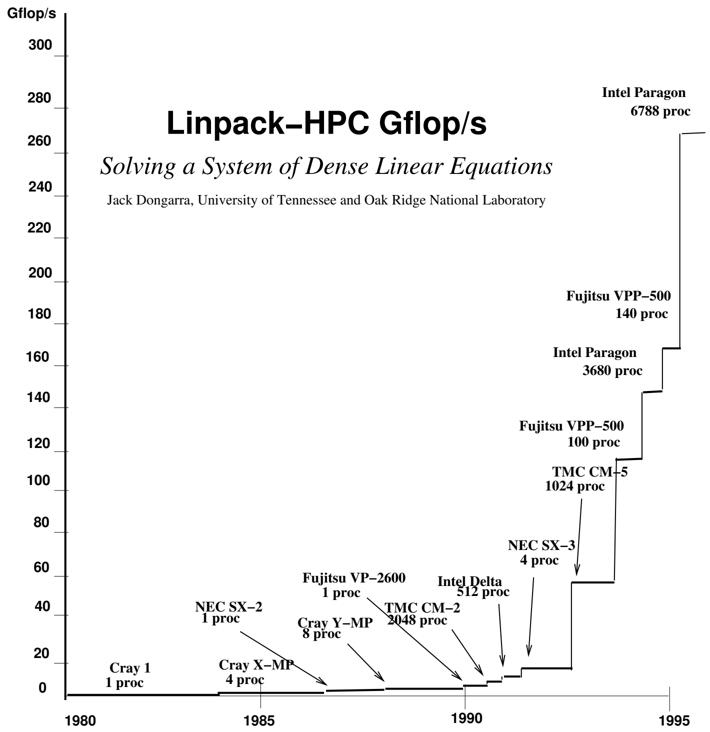
The executation rate is computed using $2/3n^3 + 2n^2$ operations.

Table 1: **Performance in Solving a System of Linear Equations**

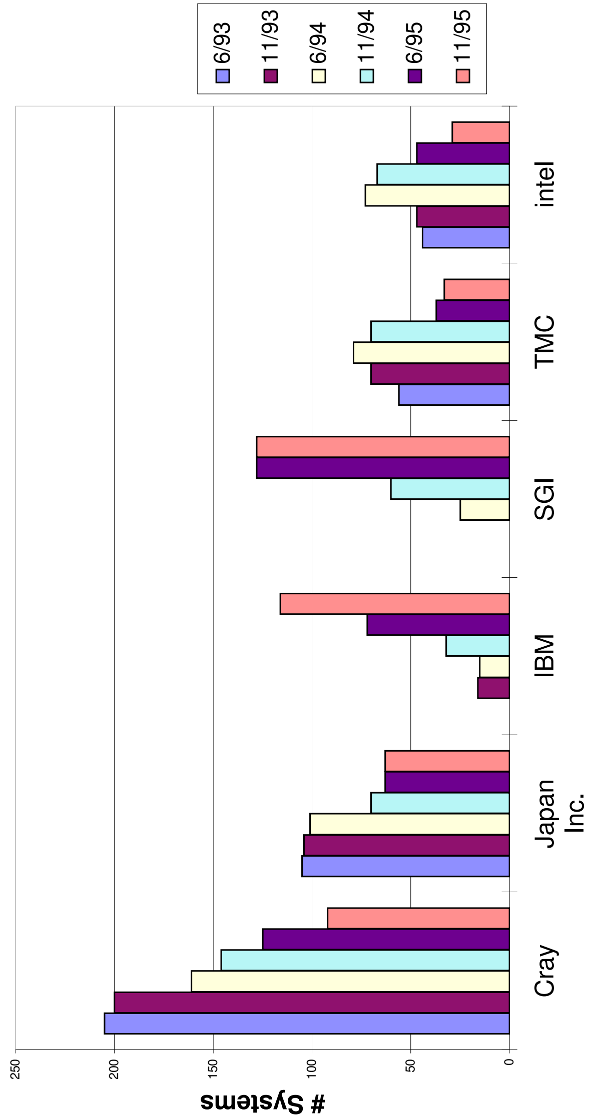
Computer	"LINPACK Benchmark" n = 100 OS/Compiler	Mflop/s	"TPP" Best Effort n=1000, Mflop/s	"Theoretical Peak" Mflop/s
Cray T932 (32 proc. 2.2 ns)	cf77 (6.0) -Zp -Wd-68		29360	57600
Cray T928 (28 proc. 2.2 ns)	cf77 (6.0) -Zp -Wd-68		28340	50400
Cray T924 (24 proc. 2.2 ns)	cf77 (6.0) -Zp -Wd-68		26170	43200
Cray T916 (16 proc. 2.2 ns)	cf77 (6.0) -Zp -Wd-68		19980	28800
Cray T916 (8 proc. 2.2 ns)	cf77 (6.0) -Zp -Wd-68		10880	14400
Cray T916 (4 proc. 2.2 ns)	cf77 (6.0) -Zp -Wd-68		5700	7200
Cray T916 (2 proc. 2.2 ns)	cf77 (6.0) -Zp -Wd-68		2937	3600
Cray T916 (1 proc. 2.2 ns)	cf77 (6.0) -Zp -Wd-68	522	1577	1800
Cray C90 (16 proc. 4.2 ns)	CF77 5.0 -Zp -Wd-e68	479	10780	15238
Cray C90 (8 proc. 4.2 ns)	CF77 5.0 -Zp -Wd-e68	468	6175	7619
Cray 3-128 (4 proc. 2.11 ns)	CSOS 1.0 level 129	421	2862	3792
Hitachi S-3800/480(4 proc 2 ns)			20640	32000
Hitachi S-3800/380(3 proc 2 ns)			16880	24000
Hitachi S-3800/280(2 proc 2 ns)			12190	16000
Hitachi S-3800/180(1 proc 2 ns)	OSF/1 MJ FORTRAN:V03-00	408	6431	8000
Cray 3-128 (2 proc. 2.11 ns)	CSOS 1.0 level 129	393	1622	1896
Cray C90 (4 proc. 4.2 ns)	CF77 5.0 -Zp -Wd-e68	388	3275	3810
Cray C90 (2 proc. 4.2 ns)	CF77 5.0 -Zp -Wd-e68	387	1703	1905
Cray C90 (1 proc. 4.2 ns)	CF77 5.0 -Zp -Wd-e68	387	902	952
NEC SX-3/44R (4 proc. 2.5 ns)			15120	25600
NEC SX-3/42R (4 proc. 2.5 ns)			8950	12800
NEC SX-3/41R (4 proc. 2.5 ns)			4815	6400
NEC SX-3/34R (3 proc. 2.5 ns)			12730	19200
NEC SX-3/32R (3 proc. 2.5 ns)			6718	9600
NEC SX-3/31R (3 proc. 2.5 ns)			3638	4800
NEC SX-3/24R (2 proc. 2.5 ns)			9454	12800
NEC SX-3/22R (2 proc. 2.5 ns)			5116	6400
NEC SX-3/21R (2 proc. 2.5 ns)			2627	3200
NEC SX-3/14R (1 proc. 2.5 ns)	f77sx 040 R2.2 -pi*:*	368	5199	6400
NEC SX-3/12R (1 proc. 2.5 ns)	f77sx 040 R2.2 -pi*:*	368	2757	3200
Cray 3-128 (1 proc. 2.11 ns)	CSOS 1.0 level 129	327	876	948
NEC SX-3/44 (4 proc. 2.9 ns)			13420	22000
NEC SX-3/24 (2 proc. 2.9 ns)			8149	11000
NEC SX-3/42 (4 proc. 2.9 ns)			7752	11000
NEC SX-3/22 (2 proc. 2.9 ns)			4404	5500
NEC SX-3/14 (1 proc. 2.9 ns)	f77sx 020 R1.13 -pi*:*	314	4511	5500
NEC SX-3/12 (1 proc. 2.9 ns)	f77sx 020 R1.13 -pi*:*	313	2283	2750
Cray Y-MP/832 (8 proc. 6 ns)	CF77 4.0 -Zp -Wd-e68	275	2144	2667
Fujitsu VP2600/10 (3.2 ns)	FORTRAN77 EX/VP V11L10	249	4009	5000
Cray Y-MP/832 (4 proc. 6 ns)	CF77 4.0 -Zp -Wd-e68	226	1159	1333
Fujitsu VPP500/1(1 proc. 10 ns)	FORTRAN77EX/VP V12L20	206	1490	1600
Cray Y-MP M98 (8 proc. 6 ns)	CF77 5.0 -Zp -Wd-e68	204	1733	2666
Fujitsu VP2200/10 (3.2 ns)	FORTRAN77 EX/VP V12L10	203	1048	1250
Cray 2S/4-128 (4 proc. 4.1 ns)	CSOS 1.0 level 129	202	1406	1951

Table 3: **Highly Parallel Computing**

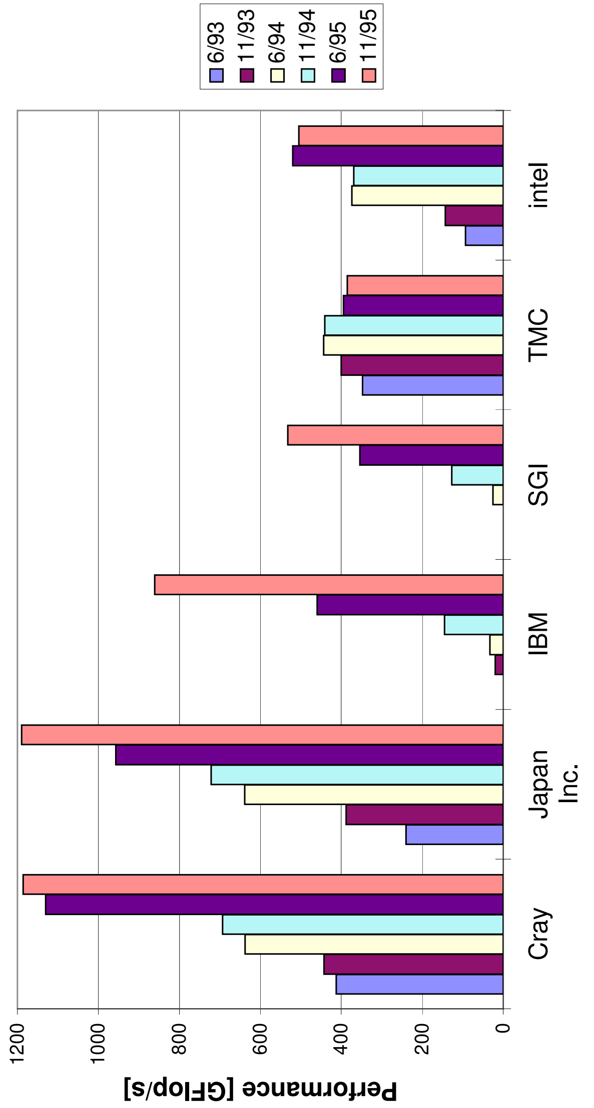
Computer (Full Precision)	Number of Processors	R_{max} Gflop/s	N_{max} order	$N_{1/2}$ order	R_{peak} Gflop/s
Intel Paragon XP/S MP (50 MHz OS=SUNMOS)	6768	281.1	128600	25700	338
Intel Paragon XP/S MP (50 MHz OS=SUNMOS)	6144	256.2	122500	24300	307
Intel Paragon XP/S MP (50 MHz OS=SUNMOS)	5376	223.6	114500	22900	269
Intel Paragon XP/S MP (50 MHz OS=SUNMOS)	4608	191.5	106000	21000	230
Numerical Wind Tunnel* (9.5 ns)	140	170.4	42000	13800	236
Numerical Wind Tunnel* (9.5 ns)	128	157.9	40960	13120	216
Intel Paragon XP/S MP (50 MHz OS=SUNMOS)	3648	151.7	95000	18100	182
Fujitsu VPP500/128(10nsec)	128	149.7	40960	13120	205
Intel Paragon XPS-140 (50 MHz OS=SUNMOS)	3680	143.4	55700	20500	184
Paragon XP/S MP(1024 Nodes, OS=SUNMOS S1.6)	3072	127.1	86000	17800	154
Fujitsu VPP500/100(10nsec)	100	120.4	36000	11450	160
Fujitsu VPP500/96 (10nsec)	96	116.2	35712	11200	154
Cray T3D 1024 (150 MHz)	1024	100.5	81920	10224	152
Fujitsu VPP500/80 (10nsec)	80	98.9	32640	10050	128
IBM SP2-T2 (66 MHz)	512	88.4	73500	20150	136
Fujitsu VPP500/64 (10nsec)	64	81.1	29440	8950	102
Hitachi S-3000 cluster/412 (3x4) (2 ns)	12	78.2	31120	4880	96
Intel Paragon XPS-140 (50 MHz)	1872	72.9	55000	17500	94
Paragon XP/S MP(512 Nodes, OS=SUNMOS S1.6)	1516	64.0	61000	12200	77
Cray T932 (2.2 ns) ***	32	61.8	16384	1280	58
Thinking Machines CM-5	1024	59.7	52224	24064	131
Hitachi S-3000 cluster/309 (3x3) (2 ns)	9	59.0	26940	3180	72
Hitachi S-3000 cluster/408 (2x4) (2 ns)	8	54.1	31200	3760	64
Cray T932 (2.2 ns) ***	24	51.1	16384	1000	43
Cray T3D 512 (150 MHz)	512	50.8	57856	7136	76
IBM SP2-T2 (66 MHz)	256	44.2	53000	13500	68
Fujitsu VPP500/32 (10nsec)	32	42.4	20736	4940	51
Hitachi S-3000 cluster/306 (2x3) (2 ns)	6	40.9	27000	2400	48
Hitachi S-3000 cluster/206 (3x2) (2 ns)	6	40.6	21600	2160	48
Cray T932 (2.2 ns) ***	16	36.6	16384	1000	29
Thinking Machines CM-5	512	30.4	36864	16384	66
Paragon XP/S MP(256 Nodes, OS=SUNMOS S1.6)	768	31.7	43500	8400	38
IBM SP2-T2 (66 MHz)	160	28.7	42200	10300	42
Hitachi S-3800/480 (2 ns)	4	28.4	15500	830	32
Hitachi S-3000 cluster/204 (2x2) (2 ns)	4	27.9	21600	1640	32
Hitachi S-3000 cluster/404 (1x4) (2 ns)	4	27.2	31200	2680	32
SGI POWER CHALLENGE (90 MHz)	128	26.7	53000	20000	46
Cray T3D 256 (150 MHz)	256	25.3	40960	4918	38
NEC SX-3/44R (2.5 ns)	4	23.2	6400	830	26
IBM SP2-T2 (66 MHz)	128	22.9	37000	9200	34
Fujitsu VPP500/16 (10nsec)	16	21.7	14592	3090	26
Hitachi S-3800/380 (2 ns)	3	21.6	15680	760	24
Hitachi S-3000 cluster/303 (1x3) (2 ns)	3	21.5	27000	1560	24
NEC SX-3/44 (2.9 ns)	4	20.0	6144	832	22
SGI POWER CHALLENGE (75 MHz)	96	18.5	53000	20000	29



Top500 - Manufacturers



TOP500 - Manufacturers



Commerical HPC Vendors

Status	Vendors
Out of business	Alliant, American Supercomputer, Ametek, Culler, Cydrome, Cray Comp. Denelcor, Elexsi, Kendall Square, Multiflow, Myrias, Prevec, Prisma, Saxpy, SCS, SSI(2), Trilogy, Wavetracer
Division closed	Astronautics, BBN, CDC/ETA Systems, E&S, FPS, Goodyear, Gould, Loral, Vitesse
Merged	Celerity, FPS, Key, Supertek, Ardent/Steller
Down, not out	AMT(Cambridge), CHoPP, Encore, Stardent/Kubota, Thinking Machines
Currently active	Convex/HP, Cray Research, Fujitsu, IBM, Intel, nCUBE, Meiko, NEC, Parsytec, SGI, Tera

Measure of the supercomputer density by ranking the top ten countries with the highest number of supercomputer per capita.

Population (in thousands) per TOP500 Supercomputer.

Country	Population (in 1K)	# of entries	Population (in 1K) per HPC
Switzerland	6,813	9	757
Singapore	2,769	3	923
USA	255,200	261	978
Denmark	5,158	4	1290
Norway	4,288	3	1429
Finland	5,008	3	1669
Germany	80,250	48	1672
Netherlands	15,160	9	1684
Japan	124,500	73	1705
Hong Kong	5,800	3	1933
Sweden	8,652	4	2163
France	57,180	25	2287
Austria	7,776	3	2592
UK	57,700	17	3394
Canada	27,370	8	3421

Highly Parallel Supercomputing: Where Are We?

1. Performance:

- Sustained performance has dramatically increased during the last year.
- On most applications, sustained performance per dollar now exceeds that of conventional supercomputers.

  But
- Conventional systems are still faster on some applications.

2. Languages and compilers:

- Standardized, portable, high-level languages such as HPF, PVM and MPI are available.

  But
- Initial HPF releases are not very efficient.
- Message passing programming is tedious and hard to debug.
- Programming difficulty remains a major obstacle to usage by mainstream scientist.

Scientific Computing: 1985 vs. 1995

- **1985:**
  1. **Minisupercomputers (1 - 20 Mflop/s): Alliant, Convex, DEC.**
  2. **Parallel vector processors (PVP) (20 - 2000 Mflop/s): CRI, CDC, IBM.**
- **1995:**
  1. **RISC workstations (10 - 600 Mflop/s): DEC, HP, IBM, SGI, Sun.**
  2. **RISC based symmetric multiprocessors (SMP) (0.5 - 7 Gflop/s): Convex, CRI, DEC, and SGI.**
  3. **Parallel vector processors (1 - 30 Gflop/s): Convex, CRI, Fujitsu, Hitachi, IBM, and NEC.**
  4. **Highly parallel processors (1 - 150 Gflop/s): Convex/HP, CRI, Fujitsu, IBM, Intel, MasPar, nCUBE.**

PARKBENCH
(Parallel Kernels and Benchmarks)

- **Founded at a special session of SC'92, convened by Dongarra & Hey**
- **Initial goal-Convergence on an agreed set of benchmarks and procedures within 12 months**
- **Membership open to all - voting rights acquired by attendance at meetings, a la HPF**
- **Three monthly meetings with interim email discussion**
  - **Mail reflector - parkbench-comm@cs.utk.edu**
- **Supported by most many benchmarking groups**
- **Supported by computer vendors - Cray Research, Intel, IBM, Meiko, Sun,...**

PARKBENCH OBJECTIVES

- **To establish a comprehensive set of parallel benchmarks that is generally accepted by both users and vendors of parallel systems.**
- **To provide a focus for parallel benchmark activities and avoid unnecessary duplication of effort and proliferation of benchmarks.**
- **To set standards for benchmarking methodology and result-reporting together with a control database/repository for both benchmarks and the results.**
- **To make the benchmarks and results freely available in the public domain.**

- **MOTIVATION**
  - Lack of distributed memory message passing benchmarks that were generally acceptable to both users and system vendors

PARKBENCH STRUCTURE

Chairman:	**Tony Hey**	**(Southampton)**
Vice Chair:	**Jack Dongarra**	**(Knoxville)**
Secretary:	**Mike Berry**	**(Knoxville)**

Work divided between 5 subcommittees:

- **Methodology (David Bailey)**
- **Low-Level Benchmarks (Roger Hockney)**
- **Kernel Benchmarks (Tony Hey)**
- **Compact Applications (David Walker)**
- **HPF Compiler Benchmarks (Tom Haupt)**

PARKBENCH OUTPUTS

- Draft paper presented at SC'93 and SC'94 Describes philosophy, methodology, current benchmark suite, procedure for submission of compact applications.
- First release of Low-level and Kernel benchmark suite assembled Compromises codes from Genesis, LAPACK & NAS Based on Fortran 77 + PVM for message-passing. Future versions will use MPI for message-passing, also HPF & Fortran 90
- Proposed applications areas and procedure for submission of compact applications identified
- Email discussion group set up - parkbench-comm@cs.utk.edu
- Prototype Performance Database Server set up on Xnetlib and the WWW via Mosaic
  - provides access to benchmarks results and literature.
- Finalized, first report available:
  - R.W. Hockney and M.W. Berry, "Public International Benchmarks for Paralle l Computers", PARKBENCH Committee report number 1, Scientific Programming 3(2), 1 01- 146, 1994.

METHODOLOGY David Bailey, NASA Ames

- Units
- Metrics
- Procedure
- Optimization
- Source
- Performance Database

LOW-LEVEL BENCHMARKS

Roger Hockney, Southampton University

Use of Linpack and Livermore Loops recommended to examine single node performanc e but are not part of suite. Low-Level benchmarks measure basic machine parameters, mostly derived from Genes is.

TICK1	**Measurement of clock resolution**
TICK2	**Measurement of clock correctness**
RINF1	**Measurement of vectorization parameters** r_∞ **and** $n_{1/2}$
POLY1	**In-cache memory bottleneck**
POLY2	**Out-of-cache memory bottleneck**
POLY3	**Communications-bottleneck**
COMMS1	**Unidirectional single message transfer ('pingpong' benchmark)**
COMMS2	**Bidirectional exchange of two messages ('pingping' benchmark)**
COMMS3	**Saturation Bandwidth**
SYNCH1	**Barrier synchronization cost**

KERNEL BENCHMARKS

Tony Hey, Southampton University

- **Matrix Kernels (ScaLAPACK/PVM)**
  - **Dense matrix multiply**
  - **Transpose**
  - **Dense LU factorization with partial pivoting**
  - **QR Decomposition**
  - **Matrix tridiagonalization**
- **FFT Kernels - 3-D Transforms (NAS)**
- **PDE Kernels**
  - **Multigrid (NAS)**
- **Miscellaneous**
  - **Embarrassingly Parallel (NAS)**
  - **Large Integer Sort (NAS)**
  - **Conjugate Gradient (NAS)**
  - **Input/Output**

PARKBENCH KERNEL BENCHMARKS

- **All Available (Input/Output - pencil and paper style)**
- **NAS Benchmarks - NASA in-house parallel implementations available for the Intel iPSC and the CM-2 and PVM3.3**

PARKBENCH COMPACT APPLICATIONS

- SEIS1.2
  - Seismic processing performance evaluation suite from ARCO
  - PVM and NX versions available
- POLMP
  - Proudman Oceanographic Laboratory Multiprocessing Program from the UK Natural Environment Research Council sequential, HPF, and PVM versions available
- PSTSWM
  - Parallel Spectral Transform Shallow Water Model from ORNL
  - PVM and MPI versions available
- NAS
  - BT, SP and LU
  - PVM version in testing, MPI in progress

Anticipated applications areas include:

- Climate and meteorological modelling
- Computational Fluid Dynamics
- Molecular Dynamics
- Finance, e.g. portfolio optimization
- Plasma Physics
- Quantum Chemistry
- Quantum Chromodynamics
- Oil-Reservoir Modelling

PARKBENCH HPF COMPILER BENCHMARKS

- Synthetic applications to test aspects of HPF compilation
- Addresses HPF subset
  - FORALL statement - kernel FL
  - Explicit template - kernel TL
  - Communication detection in array assignments
    - kernels AA, SH, ST (structured communication)
    - kernel IR (unstructured communication)
  - Non-elemental intrinsic functions - kernel RD
  - Passing distributed arrays as subprogram arguments
    - kernels AS, IT, IM

PARKBENCH FURTHER WORK REQUIRED

- Investigation of suitable codes for the Input/Output Kernel
- Uniformity of benchmark codes
  - Uniform timer routine (currently, gettimeofday())
  - Language conformance - C and F77 (with defined extensions)
    - PVM and MPI message passing
    - no machine dependent code
  - Standard Makefile
  - Minimize Interactive features
  - Single source code file if possible
  - Standard data sets for benchmarks or a configuration file with details of how to alter the data and the effect on memory requirement
- Reorganize Server Repository (in progress)
  - Headings to clearly identify each benchmark
  - Identify available versions e.g. PVM, MPI
  - Check each code is the current version
  - Test each code on a number of machines
  - Check that documentation is included
- Methods for verifying benchmark results

PARKBENCH FURTHER WORK REQUIRED

- Define Rules for Running the Benchmarks
  - Possibly: One result based on unmodified source code with compiler flags listed
  - A second result which allows any modification to optimize performance, so long as the final result is correct.
- Further results are required -
  - For GBIS Graphical Interface
  - These results will be forwarded for inclusion in PDS
- Define a method for reporting bugs
- Future - MPI versions of all codes
  - Benchmarks to test gather/scatter/global sum
  - Shared memory versions of codes (using defined primitives, e.g. put/get)
  - GUI for setting up and running codes
- Further Compact Applications (e.g. Database applications)
- Analysis of Kernel Benchmarks in terms of low level parameters

STATUS UPDATE

Kernel Benchmarks are publicly available from Netlib:
Contents of the kernels suite:

- **Matrix benchmarks**
  - **Dense matrix multiply**
  - **Matrix transpose**
  - **Dense LU factorization with partial pivoting.**
  - **QR Decomposition.**
  - **Matrix tridiagonalization.**
- **Fourier transformations**
  - **1-D FFT**
  - **3-D FFT**
- **PDE kernels**
  - **SOR**
  - **MG**
- **Other**
  - **EP**
  - **CG**
  - **IS**

PARKBENCH POLMP RESULTS

- **Cray Y-MP/8**
- **Cray C90**
- **Intel iPSC/860**
- **Maspar MP-1104**
- **Meiko CS-1 i860**
- **TMC CM-200 16k**

Results have been published in the following three papers:

Ashworth, M., Parallel Processing Environmental Modelling, Proceedings of the Fifth Workshop on the Use of Parallel Processors in Meteorology, ECMWF 23-27 November 1992, (World Scientific), pp 1-25.

Ashworth, M. and Davies, A.M., Performance of a Three Dimensional Hydrodynamic Model on a Range of Parallel Computers, in Proceedings of the Euromicro Workshop on Parallel and Distributed Computing, Gran Canaria 27-29 January 1993, pp 383-390, (IEEE Computer Society Press)

Ashworth, M., A Review of the Exploitation of Par. Proc. in Environmental Modelling, in Proc. of the Conference Parallel Computational Fluid Dynamics '93, Paris, 10th-12th May 1993 (Elsevier), in press, 10pp.

The last of these is available in PostScript by anonymous ftp to bisag.nbi.ac.uk in /pub/ncssuper/reports/pcfd93.ps.

Rules

- **Run the codes "as is", must get the correct solution.**
- **Run driver with a version that must be a supported library that is available with a brief description of what is done (for disclosure).**
- **We reserve the right to verify the results of the benchmarks.**
- **Problem size: test, medium, and large**
- **Data maintained in www.netlib.org.**

PARKBENCH

ON-GOING WORK

— Uniformity of benchmark codes

Uniform timer routine

Language conformance

Remove Interactive features

Single source file, machine dependent code in separate file

— Further Results

PARKBENCH AVAILABILITY

- PARKBENCH Description, Source codes, Report, Documentation
  - WWW URL:-http://www.netlib.org/parkbench/html/
  - to go directly to the download page http://www.netlib.org/parkbench/
- Genesis Benchmark Information Service (GBIS) Graphical
  - Interface
  - Interactive graphical results for genesis and PARKBENCH
  - http://hpcc.soton.ac.uk/RandD/gbis/papiani-new-gbis-top.html
- PDS
  - http://netlib2.cs.utk.edu/performance/html/PDStop.html
- HPCC National Software Exchange (NSE)
  - http://www.netlib.org/nse/home.html

Genesis benchmark Information Service (GBIS) Graphical Interface

- Graphical benchmark results for Genesis and ParkBench Benchmark suites available on the World Wide Web
- Interactive selection of benchmark and computers from which a performance graph is generated and displayed
- Default graph options can be changed:
  - Available output formats :- gif, xbm, postscript, tabular
  - Selectable ranges
  - Choice of log/linear axes
  - Key to traces can be positioned as required
- Links to Genesis and ParkBench information including Benchmark documentation, papers, source codes, etc.
- **World Wide Web URL's:-**
  - http://hpcc.soton.ac.uk/RandD/gbis/papiani-new-gbis-top.html
  - http://hpcc.soton.ac.uk/RandD/gbis/papiani-new-gbis.html
  - http://hpcc.soton.ac.uk/RanD/RandD.html
  - **Anonymous ftp Access to Tabular Results:-**
  - par.soton.ac.uk
  - in directory /pub/benchmark_results

PARKBENCH

AVAILABILITY

— PARKBENCH Description, Source codes, Report, Documentation

WWW URL:-http://www.epm.ornl.gov/ walker/parkbench/

to go directly to the download page

http://www.netlib.org/parkbench/

or via xnetlib

— Genesis Benchmark Information Service (GBIS) Graphical Interface

Interactive graphical results for genesis and PARKBENCH

http://hpcc.soton.ac.uk/RandD/gbis/papiani-new-gbis-top.html

— PDS

http://netlib2.cs.utk.edu/performance/html/PDStop.html

— HPCC National Software Exchange (NSE)

http://www.netlib.org/nse/home.html

SPEC High Performance Steering Committee

Founded in January 1994

Members have met twice to elect officers, define charter, discuss policies, benchmark selection criteria, run and reporting rules

Current members are:

- **Convex Computers**
- **Cray Research**
- **Dartmouth College**
- **Digital Equipment Corporation**
- **Fujitsu America**
- **Hewlett-Packard**
- **IBM**
- **Intel SSD**
- **Kuck and Associates**
- **NEC/HNSX Supercomputers**
- **Silicon Graphics Inc.**
- **Sun Microsystems Inc.**
- **University of Illinois**
- **University of Michigan**
- **University of Minnesota**
- **University of Tennessee**

SPEC/HPSC - Aims and Scope

HPSC's mission is to establish, maintain and endorse a suite of benchmarks representative of real world high-performance computing applications, for standardized cross platform comparison on a level playing field.

Aim is to cover all advanced architectures, will focus on

- symmetric multi-processor systems
- workstation clusters
- distributed memory parallel systems
- vector and vector parallel supercomputers

End users are seen as the ultimate beneficiaries of results

- wish to establish a close relationship with the high- performance computing community.

SPEC/HPSC - Parkbench Relationship

SPEC/HPSC will draw on Parkbench experience and expertise

- choice of API, choice of codes, methodology

Parkbench more research oriented than SPEC/HPSC

Parkbench will draw on SPEC/HPSC experience

- commercial user interests
- vendor interests

To encourage collaboration and interaction

- Joint B.O.F. meeting at Supercomputing 95
- Parkbench participation at SPEC/HPSC meetings
- Continuing SPEC/HPSC participation at Parkbench meetings

NHSE

- National HPCC Software Exchange
- NASA (plus other agencies) funded CRPC project
- Center for Research on Parallel Computation (CRPC)
  - Argonne National Laboratory
  - California Institute of Technology
  - Rice University
  - Syracuse University
  - University of Tennessee
- Uniform interface to distributed HPCC software repositories
- Facilitation of cross-agency and interdisciplinary software reuse
- Material from ASTA, HPCS, and IITA components of the HPCC program
- http://www.netlib.org/nhse/

Goals:

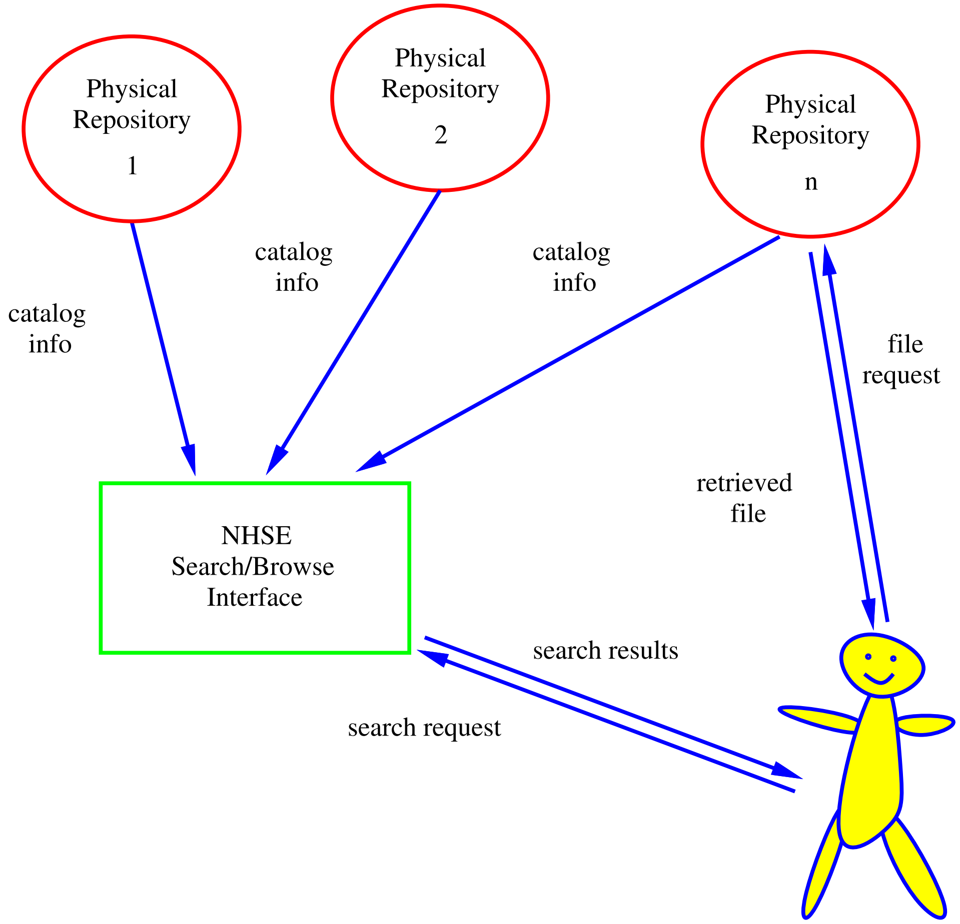
- To facilitate an active exchange program for HPCC software and enabling technologies via the National Information Infrastructure.

- To promote contributions and use by Grand Challenge teams, as well as other members of the high performance computing community.

Software here includes algorithms, specifications, designs, documentations, reports...

NHSE Components

- Outreach and technology transition
  - To the HPCC user community and industry
- Distribution via the WWW
- Discipline oriented repositories
- Uniform user interface
- Software review
  - Simple submission
  - On going review
- Measurement
- Hypertext road map
- Publishing tools
  - Repository-in-a-box
- Naming & authentication architecture
- Selective capitalization of emerging technologies



Virtual Repository Architecture

Benefits:

1. Faster development of high-quality software so that scientists can spend less time writing and debugging programs and more time on research problems.
2. Less duplication of software development effort by sharing of software modules.
3. Less time and effort spent in locating relevant software and information through the use of appropriate indexing and search mechanisms and domain-specific expert help systems.
4. Reducing information overload through the use of filters and automatic search mechanisms.

Intended Audience:

- HPCC application and computer science community
  - Source of material for NHSE
- Users of NASA, NSF, DOE and other supercomputer centers
  - Good targets for NHSE
  - Natural support organization: supercomputer center staff
- Other users of high performance computers
  - Current and potential industrial users
  - No natural support organization
- Applicable to other domains

NHSE

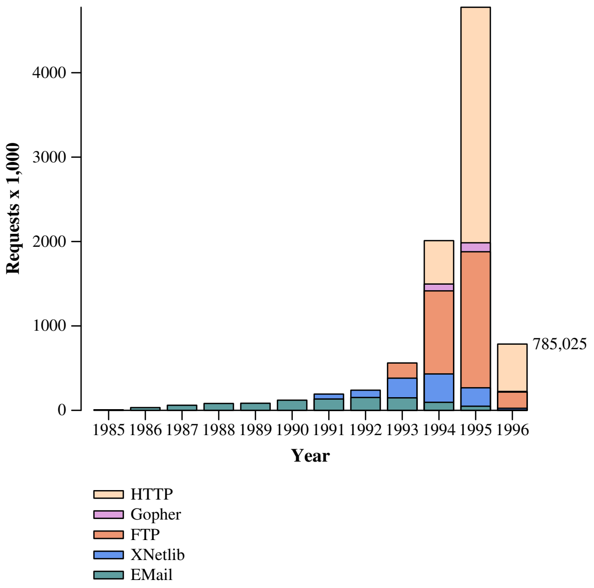
- Based on Existing Technologies
  - WWW Browser (Mosaic / Netscape / etc)
    - Distributed / Scalable
    - URL: http://www.netlib.org/nhse/
  - Netlib
    - Repository for math software since 1985
- Repositories Currently Available
  - Netlib, Softlib, CITlib
  - ASSET - (Asset Source for SW Engineering Tech.)
  - CARDS - (Comprehensive Approach to Reusable Defense SW)
  - ELSA - (Electronic Library Services and Appl.)
  - GAMS (Virtual Software Repository)
  - STARS - (SW Technology for Adaptable, Reliable Systems)
  - Many examples related to GC problems
- Currently Available Information
  - NHSE currently points to 350+ modules
    - software catalog, tech reports and papers
    - parallel processing tools
    - numerical libraries
    - Grand Challenge prototype codes
    - data analysis and visulization
    - benchmarks

Netlib - Network Access to Mathematical Software and Data

- Began in 1985
  - JD and Eric Grosse, AT&T Bell Labs
- Motivated by the need for cost-effective, timely distribution of high-quality mathematical software to the community.
- Designed to send, by return electronic mail, requested items.
- Automatic mechanism for the disseminate of public domain software.
  - Still in use and growing
  - Mirrored at a number of sites
    - netlib2.cs.utk.edu
    - netlib1.epm.ornl.gov
    - research.att.com
    - netlib.no
    - unix.hensa.ac.uk
    - ftp.zip-berlin.de
    - nchc.edu.tw
- Moderated collection of high-quality math software
- Distributed maintenance
- Model for domain-specific repositories

Requests Made to the Netlib Repositories at the Univ. of Tennessee & ORNL

8,946,816 total requests to these repositories as of Feb 15, 1996



Data as of - Feb 15, 1996 at 02:07:27

Breakdown of requests to each Netlib library

(an alphabetical listing is also available.)
Data as of 02/15/96 at 02:09:20

Library Name	Number of accesses
lapack	475,979
pvm3	379,849
linpack	256,403
slatec	248,292
blas	178,728
clapack	129,256
linalg	127,022
eispack	126,116
slatec/src	118,366
toms	117,152
f2c	98,025
c++	96,774
benchmark	85,552
master	69,997
f2c/src	67,415
minpack	60,632
fn	59,781
fftpack	58,805
na-digest	50,970
port	49,496
slatec/lin	46,800
hence	45,229
confdb	43,118
slatec/chk	37,640
c++/answerbook	37,524
napack	36,719

The following types of software are being made available:

- Systems software and software tools.
  - compilers
  - message-passing communication subsystems
  - parallel monitors and debuggers.
- Basic building blocks for accomplishing common computational and communication tasks.
  - Building blocks are meant to be used by Grand Challenge teams
- Research codes that have been developed to solve difficult computational problems.
  - Many have been developed to solve specific problems
  - Serve as proofs of concept
  - Models for developing general-purpose reusable software

Review Procedure

- Submissions are “subject” to (ongoing) review.
- Review status abstract for each submission.
  - Based on author comments
  - Package documentation
  - Our independent reviewer testing
  - Comments from users.

NHSE Software Catalog

- Benchmark and example programs (4)
- Data analysis and visualization (22)
- Numerical libraries and routines (57)
  - Computational geometry (7)
  - Linear algebra (18)
  - Optimization (4)
  - Partial differential equations (3)
  - Other (25)
- Parallel processing tools
  - Communication libraries (25)
  - Execution and performance analyzers (31)
  - Parallel I/O systems (5)
  - Parallel programming environments (12)
  - Parallel programming languages and compilers (26)
  - Parallel runtime systems (10)
  - Source code analyzers and restructurers (7)
  - Miscellaneous (16)
- Scientific and engineering applications (66)

NHSE Software Submission

Goals

- Exercise quality control
  (review classification)
- Ensure *fixity of publication*
  (file fingerprints, unique name)
- Prevent impersonation and unauthorized changes
  (digital signatures)
- Promote interoperability
  (RIG Basic Interoperability Data Model)

Technical and Political Issues

- How will the naive user find the right software?
  - Answer: Via the NHSE search/browser interface and the Road Map
- How will authentication, integrity, and version control be implemented?
  - Answer: By a publishing system that includes unique naming & digital signatures.
- Will the NHSE support distribution of software that is not free or cannot be freely distributed?
  - Answer: Yes, but...
    - only provide classification, review, and access
    - use of encryption and separate key distribution
    - NHSE will not have an accounting department
- Will the NHSE be responsible for support of software?
  - Answer: NO!
    - Any support will be by author (or appropriate agent)

Summary

- Initial implementation built on existing technologies
  - WWW
    - Distributed
    - Scalable
  - Netlib, etc
  - Rapid deployment
- Multilevel review and classification scheme
- Road Map
- Measurement and evaluation
  - Statistics
  - Evaluations for reviewed

Summary (continued)

- Outreach and technology transition
  - Educational activities aimed at the user community
  - Fostering technology development by industry
- Working on standardization within WWW community
  - member of RIG, IETF, IESG, WWW consort

Some Url's Related to Scientific Computing

- **National HPCC Software Exchange**
  http://www.netlib.org/nhse/
- **NHSE Software Catalog**
  http://www.netlib.org/nhse/sw_catalog/index.html
- **Netlib Repository**
  http://www.netlib.org/
- **Computational Science Education**
  http://www.netlib.org/nhse/cse_edu.html
- **Books, Course Materials, and Tutorials**
  http://www.netlib.org/nhse/cse_edu.html#books
- **CS267 - Applications of Parallel Computers**
  U.C. Berkeley CS267: Spring 1995 Jim Demmel
  http://www.icsi.berkeley.edu/cs267/
- **18.337 Parallel Scientific Computing**
  MIT Spring, 1995 Alan Edelman
  http://web.mit.edu/18.337/WWW/home.html
- **North Carolina State University**
  Visualizations in Materials Science
  http://vims.ncsu.edu/cgi/index.acgi
- **Computational Science Education Project**
  http://csep1.phy.ornl.gov/csep.html

High Performance Fortran (HPF)

- **Low-level parallel programming is hard**
- **Parallel programs are not portable**
- **Parallel programs could not be run on sequential computers**
- **Data parallel programming model seems to be good for many applications**
- **HPF is an emerging shared memory programming standard for massively parallel computers**
- *HPF Language Specification, Version 1.0, May 3, 1993*
- **Limited HPF subset compilers**

Conclusions on Interconnects

- **High bandwidth and low latency remain a challenge for MPP systems**
- **However, clustered workstations will provide an efficient throughput system for small and medium range jobs**
- **Latency is particularly poor on workstation clusters and limits their use on most applications**

Lessons Learned During 1985-1995

1. Massive parallel designs are as effective as highly parallel.
2. SIMD hardware was not very successful.
3. Designs employing RISC processors are quite successful.
4. Interconnect latency and bandwidth are very important.
5. Reliable system software is a challenge.
6. Efficient implementations of good serial algorithms do the job and are best parallel algorithms.
7. Different applications have different performance characteristics.

The Maturation of Highly Parallel Technology

- Affordable parallel systems now out-perform the best conventional supercomputers.
- Performance per dollar is particularly favorable.
- The field is thinning to a few very capable systems.
- Reliability is greatly improved.
- Third-party scientific and engineering applications are appearing.
- Business applications are appearing.
- Commercial customers, not just research labs, are acquiring systems.

Vector Parallel Architecture

- **Peak performance of vector processor is at least 3-4 times better than microprocessor. Peak performance of a single NEC SX-4 processor is 2.0 Gflop/s and DEC 21164 300 MHz is 600 Mflop/s.**
- **PVP needs only 1/4 parallelism of HPP**
- **PVP can use crossbar providing high network efficiency**
- **Ease of programming**

RISC Workstations: The Enemy Below

- **Sales revenues now exceed that of mainframe supers and minisupers combined.**
- **Most include high-resolution color graphics.**
- **Highly reliable hardware and software.**
- **Single-CPU performance approaches that of vector supers.**
- **Vigorous competition between leading vendors assures continued rapid increases in sustained performance.**

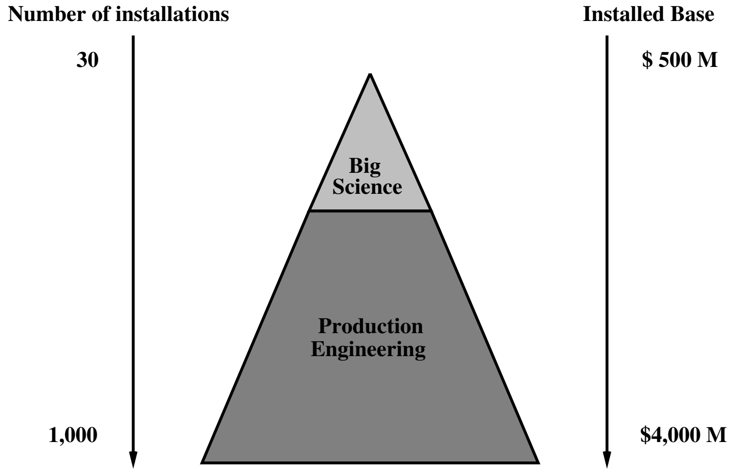
Lesson to supercomputers centers:

Those centers that do not give priority to large, grand challenge jobs risk losing their funding.

Lesson to supercomputers manufacturers:

Those vendors that do not press the outer envelope of single job performance risk being driven out of business.

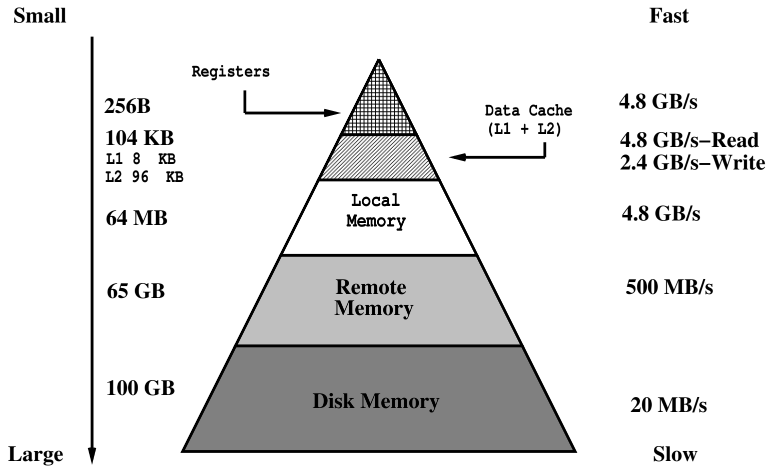
Current State of HPC Market



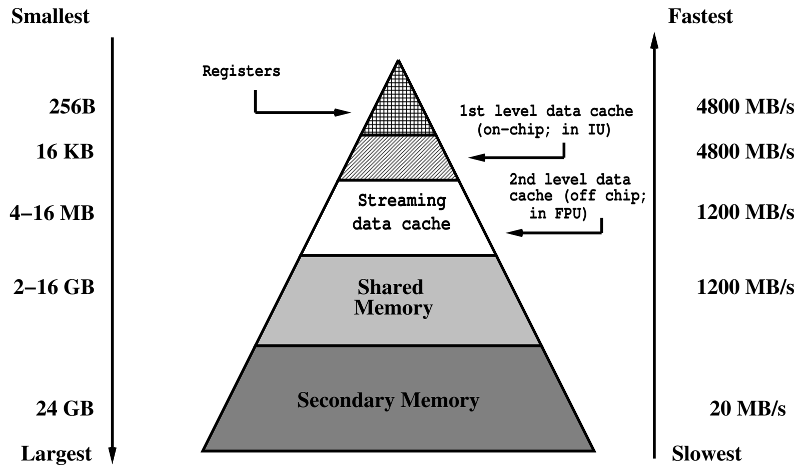
Memory Latencies

Function	EV4 used in Cray T3D		EV5 used in Cray T3E	
	Time in CP	Time in ns	Time in CP	Time in ns
L1 cache load	3	20.01	2	6.67
L2 cache load	na	na	8	26.64
RA/stream hit	13	86.71	21	69.93
DRAM page hit	22	146.74	63	209.73
DRAM page miss	37	246.79	87	289.71

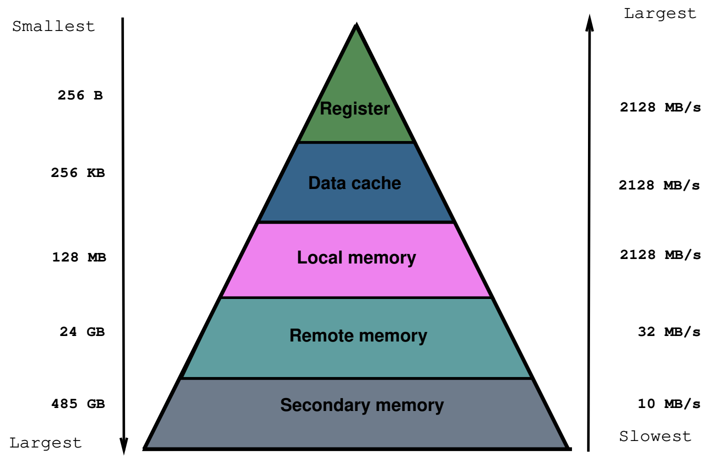
CRAY T3E Memory Hierarchy



Power Challenge Memory Hierarchy



IBM SP2 Memory Hierarchy



Intel Paragon Memory Hierarchy