Choosing Programs to Evaluate Performance

Measure the execution time of the workload.

(Real) Programs - compilers for C/Fortran, text processing like TeX, CAD tools like Spice. Have input, output and options the user can select.

Kernels - small key pieces from real programs. Livermore Loops, LINPACK are examples. Exist solely to evaluate performance. Best used to isolate features, explain differences on real programs.

(Toy) Benchmarks - Between 10 and 100 lines of code. Produces a result the user already knows before the run. Examples: Sieve of Erastosthenes, Puzzle, and Quicksort. Best used as beginning programming assignments.

Syntheic Benchmarks - Similar to kernels, trys to match the frequency of operations and operands of a large set of programs. These are artifically created to match average exec profile. Examples: Whetstone and Dhrystone.

A guiding principle in performance measurement should be reproducibility.
The gap between peak performance and observed performance is typically a factor of 10 or more in high-performance computers.

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

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 $||Ax - b||/(||A||||x||)$.

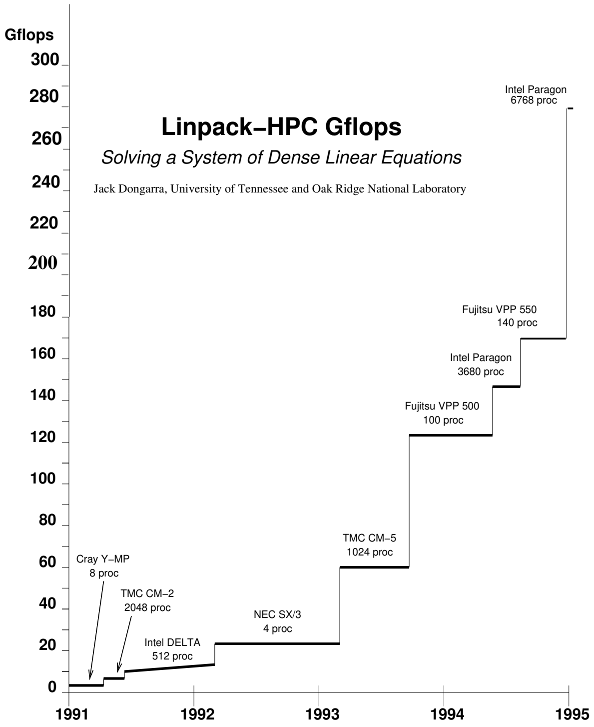
The executation rate is computed using $2/3n^3 + 2n^2$ operations.

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 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
NEC SX-3/11R (1 proc. 2.5 ns)	f77sx 040 R2.2 -pi*:*	202	1418	1600
NEC SX-3/1LR (1 proc. 2.5 ns)	f77sx 040 R2.2 -pi*:*	201	767	800
CRAY Y-MP/832 (2 proc. 6 ns)	CF77 5.0 -Zp -Wd-e68	181	604	667
CRAY X-MP/416 (4 proc. 8.5 ns)	CF77 4.0 -Zp -Wd-e68	178	822	940
CRAY Y-MP M98 (4 proc. 6 ns)	CF77 5.0 -Zp -Wd-e68	177	1114	1333
NEC SX-3/11 (1 proc. 2.9 ns)	f77sx 020 R1.13 -pi*:*	173	1223	1370
NEC SX-3/1L (1 proc. 2.9 ns)	f77sx 020 R1.13 -pi*:*	171	661	680
Fujitsu VP2400/10 (4 ns)	FORTRAN77 EX/VP V11L10	170	1688	2000

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
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
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 T3D 512 (150 MHz)	512	50.8	57856	7136	76
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
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
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
CRAY T3D 256 (150 MHz)	256	25.3	40960	4918	38
NEC SX-3/44R (2.5 ns)	4	23.2	6400	830	26
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
NEC SX-3/34R (2.5 ns)	3	17.4	6144	691	19
Paragon XP/S MP(128 Nodes, OS=SUNMOS S1.6)	384	16.0	30700	5700	19
Intel Paragon XPS-35 (50 MHz, OS=R1.1)	512	15.2	23000	9000	26
Thinking Machines CM-5	256	15.1	26112	12032	33
Hitachi S-3800/280 (2 ns)	2	14.6	15680	570	16
Hitachi S-3000 cluster/202 (1x2) (2 ns)	2	14.5	21600	1100	16
Intel Delta (40 MHz)	512	13.9	25000	7500	20
CRAY Y-MP C90 (240 MHz 4.2 ns)	16	13.7	10000	650	15
CRAY T3D 128 (150 MHz)	128	12.8	20736	3408	19



MIMD, multicomputer: heterogenous distributed environments

Work by J. Becker, UC Davis, and Leonardo Dagum, NAS Applied Research Branch

- particle sort accounts for 38% of the CM-2 computation time
- sorting is up to 10 times faster on one processor of the Cray Y-MP
- perform sort operation on the Y-MP using the HIPPI-UltraNet connection between the two machines.
- need to transpose data on the CM-2 from parallel to serial
- need to pack/unpack data on the Y-MP
- written in C with sockets

MIMD, multicomputer: heterogenous distributed environments

- distributed code takes longer than just running on CM-2 alone
- for large problem sizes (> 500,000 particles) less than a factor of two slower
- data conversion is significant overhead factor, taking 25% – 33% of the total time for large problems

NAS Numerical Aerodynamic Simulation (NAS) Program

Goal is to demonstrate by 2000 an operations computing system capable of simulating the entire aerospace vehicle system in one to several hours

Grand Challenge will require a sustained rate of approximately 1000 times faster than 1990 generation supercomputer.

Parallel Vector Processors (PVP)

- Cray C90, T90, J90
- Fujitsu VPP500

Highly Parallel Processors (MPP)

- Cray T3D
- IBM SP-2 WN / TN2

Symmetric Multiprocessors

- Convex Exemplar SPP1000
- DEC Alpha Server 8400 5/300
- SGI Power Challenge XL (90 MHz)

NAS Parallel Benchmarks

Developed in 1991 by NASA Ames Research Center

To study the performance of parallel supercomputers

Eight benchmark programs

Paper and pencil description

Benchmark implementors are free to select

- Language constructs
- Implementation techniques

Kernels and Pseudo-Applications

The NAS-Parallel Benchmarks consist of two parts: kernels and pseudo-applications.

- Kernels alone are insufficient to assess the performance potential of a parallel machine on "real" scientific applications
- Difficulty is that a certain data structure may be very efficient on a certain system for one of the isolated kernels, and yet this data structure would be inappropriate if incorporated into a larger application
- Performance of a real CFD application on a parallel system is critically dependent on data motion between kernels
- Pseudo-application attempts the complete reproduction of this data movement
- All results here are from "NAS Parallel Benchmarks Results 10/93" by D. Bailey, E. Barscze, L. Dagum, and H. Simon, available in Postscript from leo@nas.nasa.gov

The Kernel Benchmarks

EP: An "embarrassingly parallel" kernel. It provides an estimate of the upper achievable limits for floating point performance, i.e. the performance without significant interprocessor communication.

MG: A simplified multigrid kernel. It requires highly structured long distance communication and tests both short and long distance data communication.

CG: A conjugate gradient method is used to compute an approximation to the smallest eigenvalue of a large, sparse, symmetric positive definite matrix. This kernel is typical of unstructured grid computations in that it tests irregular long distance communication, employing unstructured matrix vector multiplication.

The Kernel Benchmarks (cont'd)

FT: A 3-D partial differential equation solution using FFTs. This kernel performs the essence of many "spectral" codes. It is a rigorous test of long-distance communication performance.

IS: A large integer sort. This kernel performs a sorting operation that is important in "particle method" codes. It tests both integer computation speed and communication performance.

The Pseudo Applications

LU: Regular-sparse, block (5x5) lower and upper triangular system solution. This pseudo-application represents the computations associated with the implicit operator of a newer class of implicit CFD algorithms, typified at NASA Ames by the code INS3D-LU. It may contain only a limited amount of parallelism, relative to the other kernels.

SP: Solution of multiple, independent systems of non diagonally dominant, scalar, pentadiagonal equations. SP and the following pseudo-application BT are representative of the computations associated with the implicit operators in version of the ARC3D code. They involve global data dependencies. SP and BT are similar in many respects, but there is a fundamental difference with respect to the communication to computation ratio.

BT: Solution of multiple, independent systems of non diagonally dominant, block tridiagonal equations with a (5x5) block size.

LU Simulated CFD Application (LU) Benchmark

Does lower-upper diagonal (LU)

Does not perform LU factorization, but does SSOR to solve a regular-sparse, block 5x5 lower and upper triangular system

Required about 250 iterations

SP Simulated CFD Application (SP) BENCHMARK

Multiple independent systems of nondiagonally dominant, scalar pentadiagonal equations are solved.

Requires about 400 iterations

Block Tridiagonal Simulated CFD Applications (BT) Benchmark

Multiple independent systems of non-diagonally dominant, block tridiagonal equations with a 5x5 blocksize are solved.

Conjugate Gradient (CG) Benchmark

Conjugate gradient method is used to compute the smallest eigenvalue of a large sparse symmetric positive definite matrix

The kernel is typical of unstructured grid computations

It tests irregular long distance communication and uses sparse matrix vector multiplication

3-D FFT PDE (FT) Benchmark

3-D PDE is solved using FFT's

Kernel performs the essence of many spectral methods

Good test of long distance communication

Integer Sort (IS) Benchmarks

Tests a sorting operation that is important in particle methods similar to particle in a cell
computations

Sorting operations are used to assign particles to a cell

Test both integer computation speed and communication performance

Floating point computations are not performed

Embarrassingly Parallel (EP) Benchmark

Typical of many Monte Carlo applications

2-D statistics are accumulated from a large number of

Gaussian pseudo-random numbers

Requires no communication

Provides an estimate of the upper achievable limit for floating point performance.

Multigrid (MG) Benchmark

Solves 3-D Poisson PDE

Has constant coefficients rather then variable coefficients

Good test of short and long distance highly structured communication

There are two standard sizes for the benchmarks called Class A and B.

		CLASS A			CLASS B		
Name	abb.	size	op ct ($x10^9$)	Mflop/s Y-MP/1	size ($x10^9$)	op ct C-90/1	Mflop/s
Embarrassingly Parallel	EP	2^{28}	26.68	211	2^{30}	100.9	543
Multigrid	MG	256^3	3.905	176	256^3	18.81	498
Conjugate	MG	$14x10^3$	1.508	127	$75x10^3$	54.89	447
3-D FFT PDE	CG	256^2x128	5.631	196	$512x256^2$	71.37	560
integer Sort	FT	2^32x2^{19}	.7812	68	$2^{25}x2^{21}$	3.150	244
LU Simulation CFD Appli.	LU	64^3	64.57	194	102^3	319.6	493
SP Simulation CFD Appli.	SP	64^3	102	216	102^3	447.1	627
BT Simulation CFD Appli.	BT	64^3	181.3	229	102^3	721.5	572

- Eight problems consisting of five kernels and three simulated computational fluid dynamics applications.
- Five kernels are a compact applications each emphasizing a particular type of numerical computation.
- They can be fairly readily implemented and provide insight as to the general levels of performance that can be expected for these types of computations.
- The CFD applications require more effort to implement and are more representative of the types of actual data movement and computation required in a s-o-a CFD application code.
- Sample Fortran implementations are provided on a single processor system.

Observations on the NAS Parallel Benchmarks:

- PVP Cray C90 is no longer the performance leader
- Absolute performance of 2 CFD application on 512 Cray T3D
- 160 nodes of an IBM SP-2 WN is significantly greater than on the C90-16.
- When the system performance is normalized by system price all the highly parallel systems outperform the C90
- Portability is a big issue
- Each vendor has its own programming paradigm for parallelization
  - CONVEX SPP-1000 specific directives
  - Cray C-() specific directives
  - Cray J-90 Specific directives
  - Cray T3D explicit shared memory model using shmem-get and shmem-put
  - Fujitsu VPP-500 specific directives
  - IBM SP-2 IBM specific calls to their MPL library
  - SGI PC-XL specific directives
- No vendor has used either MPI or HPF yet
- The best computer based on performance per dollar for Class B SP and BT benchmarks is the SMP from DEC Turbo-Laser

Summary on TOP 500 in the U.S.A.

- The U.S. is the clear world leader both as a producer and as consumer of high performance computers
- Both the U.S. and Japan seem to have mechanisms in place which restrict competition in favor of the local products
- MPPs account for more than a third of all installed supercomputers worldwide
- Market penetration by MPPs in the U.S. is significantly higher, where MPPs constitute already now a majority of the installed supercomputers. A sizeable portion of this market share can be attributed to government spending and the HPCC program
- Government installations dominate the market for large systems both in the U.S. and worldwide. Overall, the U.S. lags somewhat in industrial use of supercomputers
- MPPs have made major progress in replacing PVPs in several industries. In the U.S., MPPs usage in aerospace and in the oil industry is very high. MPPs appear to be considerably more widespread in industrial environments than commonly thought

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 - pbwg-comm@cs.utk.edu**
- **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.

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)
- Compiler Benchmarks (Tom Haupt)

PARKBENCH OUTPUTS

- Draft paper presented at SC'93 in Portland 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 - pbwg-comm@cs.utk.edu
- Prototype Performance Database Server set up on Xnetlib and the WWW via Mosaic
  - provides access to benchmarks results and literature.

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 - 1-D & 3-D Transforms
- PDE Kernels
  - Successive Over-Relaxation (SOR) (SU)
  - Multigrid (NAS)
- Miscellaneous
  - Embarrassingly Parallel (NAS)
  - Large Integer Sort (NAS)
  - CG (NAS)

COMPACT APPLICATIONS

David Walker, ORNL

Suitable applications are currently being solicited.

Criteria:

1. **Complete application capable of producing research results.**
2. **High Quality code, extensively tested and validated.**
3. **Well documented.**

- **POLMP Fluid dynamics climate model FD (HPF, f77, mess pass)**
- **Solver (QCD) CG based (HPF, f77, mess pass)**
- **GAUGE (QCD) Monte Carlo based ((HPF, f77, mess pass)**
- **Shallow Water (ORNL) (f77, mess pass)**
- **ARCO Suite (f77, PVM)**
- **LU from NAS, CFD based (f77, CM Fortran, NX)**
- **SP from NAS, CFD based (f77, CM Fortran, NX)**
- **BT from NAS, CFD based (f77, CM Fortran, NX)**

To come:
Mol Dyn (ORNL CCP5) f77, NX
PIC (lpm2 Genesis, LANL)

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**

COMPILER BENCHMARKS

Tom Haupt, Syracuse University

- **HPF Programs to test compiler capability**

CURRENT AND FUTURE WORK

- **Performance analysis of PARKBENCH kernels in terms of low-level parameters (ongoing work on FFT, Matrix Multiply and SOR).**
- **Design of Application-oriented communication benchmarks in order to improve accuracy of low-level communication parameters.**
  - **Butterfly communications;**
  - **Nearest neighbor exchange on a 2-d/3-d mesh;**
  - **Rowcast/columncast;**
  - **Roll left/right;**
  - **etc.**

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**

PERFORMANCE RESULTS

We report selected kernels' results obtained to date with NPB kernels on the following parallel computers: YMP, C90, and T3D by Cray Research Inc. (CRI); Paragon and iPSC/860 by Intel; SP-1 by IBM; KSR1 by Kendall Square Research; CM-2, CM-200, and CM-5 by TMC. The performance results in Gflop/s use the flop counts of corresponding PARKBENCH kernels as determined by the hardware performance monitor on a Cray Y-MP.

- **the graphs are very useful for performance comparison across different parallel computers**
- **there should be benchmark measurements for three problem sizes:**
  - **test problem size;**
  - **moderate problem size;**
  - **grand challenge problem size.**
- **the graphs give an idea for the "design template" of the Interactive Graphical Interface.**

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.

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 Status

Aim is to cover all advanced architectures, will focus on Basic guidelines for code selection have been adopted

- **must be representative of real-world applications**
- **must be in F77, ANSI C and MPI or PVM**

Currently studying code from areas:

- **seismic data processing - Arco Suite**
- **computational chemistry - Gamess and Amber**
- **computational fluid dynamics - Turb3d**

Still looking for candidate codes in other areas, particularly CFD and structural analysis

SPEC/HPSC Contacts

HPSC information is available through:

Siamak Hassanzadeh
Chair, HPSC
Email: siamak@hpsc.Corp.Sun.Com
Phone (415) 336-0118

Fiona Sim,
Vice-chair, HPSC
Email: fsim@vnet.IBM.COM
Phone: (914) 385-4615

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

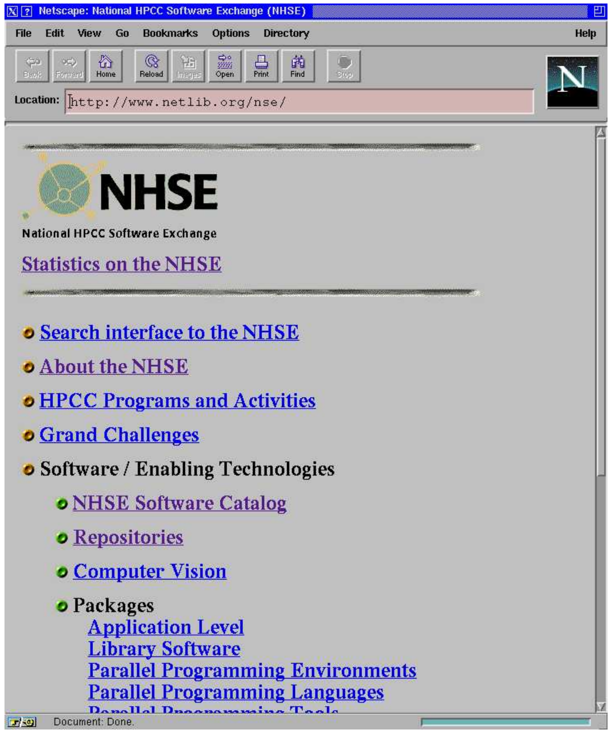
- commercial user interests
- vendor interests

To encourage collaboration and interaction

- Joint B.O.F. meeting at Supercomputing94
- Parkbench participation at SPEC/HPSC meetings
- Continuing SPEC/HPSC participation at Parkbench meetings

Interface based on existing features in common use

- Use existing protocols
- Client supports WWW, HTML, WAIS, Gopher, Archie, FTP, etc
- Define new ones (or extend old ones) when needed



PDS: The Performance Database Server

View complete, sorted benchmark reports
To display sorted data from various available published benchmark reports

Browse the performance database
To browse the performance data tree by selecting the benchmark to be listed and the machines desired

Search the performance database
To specify search strings for literal text matching

Papers
To access ASCII and Postscript benchmark papers as contributed by the authors

Bibliography
To access a bibliography of benchmark and performance–related literature in the BibTeX format. This data is contributed from various sources.

If you have questions or concerns, please send mail to utpds@cs.utk.edu

NSE Databases

- Conferences
- Glossary
- High–Performance Bibliographies
- Performance Information
- Numerical Analysis Net (NANET) Whitepages
- WAIS indices (reports, papers, etc.)
  - University of Tennessee Technical Reports
  - Numerical Analysis Digest (complete archives)
  - Numerical Analysis Digest (recent issues only)
  - LAPACK Working Notes

Add to WAIS indices section.

PDS: Browse

Select desired benchmark:

Select desired machines:

Alliant	ELXSI	Kendall Square	Pyramid
Amdahl	ETA	Maspar	Sequent
Apollo	FPS	Masscomp	SGI
Apple	Fujitsu	Meiko	Siemens
CDC	Gould	MIPS	Solbourne
Compaq	Harris	Multiflow	Sperry
Concurrent	Hitachi	NAS	Stardent
Convex	Honeywell	nCUBE	Sun
Cray	HP	NEC	Tektronix
Data General	IBM	Next	TMC
DEC	Intel	Prime	Vax

If you have questions or concerns, please send mail to utpds@cs.utk.edu

CONVEX C-120	fc 5.1	6.5	17	20
CONVEX C-1/XP	Fortran 2.0	3.0		20
CONVEX C-1/XL	Fortran 1.6	2.9		20
Alliant FX/4 (1 proc.)	-O -DAS -inline	2.1	6.3	12

PDS: Browse Query Results

Linpack Benchmark

Computer	OS/Compiler	N=100(Mflops)	N=1000(Mflops)	Peak(Mflops)
CONVEX C4/XA-4(4 proc)(7.41 ns)	fc9.0.0.5 -tm c4 -O3 -ds	160	2531	3240
CONVEX C4/XA-3(3 proc)(7.41 ns)	fc9.0.0.5 -tm c4 -O3 -ds	151	1933	2430
CONVEX C4/XA-2(2 proc)(7.41 ns)	fc9.0.0.5 -tm c4 -O3 -ds	129	1335	1620
Convex SPP-1000(8 procs)100 MHz			751	1600
CONVEX C-3880 (8 proc.) (16.7 ns)	fc7.0 -tm c38 -O3 -ep 8 -ds -is .	86	795	960
CONVEX C4/XA-1(1 proc.)(7.41 ns)	fc9.0.0.5 -tm c4 -O2 -is .	99	705	810
Convex SPP-1000(4 procs)100 MHz			442	800
Alliant FX/2800-200 (14 proc)	fortran 1.1.27 -O -inline	31	325	560
CONVEX C-3840 (4 proc.) (16.7 ns)	fc7.0 -tm c38 -O3 -ep 4 -ds -is .	75	425	480
Alliant FX/2800-200 (12 proc)	fortran 1.1.27 -O -inline	29	290	480
Convex SPP-1000(2 procs)100 MHz			255	400
Alliant FX/2800-200 (10 proc)	fortran 1.1.27 -O -inline	27	250	400
CONVEX C-3830 (3 proc.) (16.7 ns)	fc7.0 -tm c38 -O3 -ep 3 -ds -is .	71	327	360
Alliant FX/2800-200 (8 proc)	fortran 1.1.27 -O -inline	25	207	320
CONVEX C-3820 (2 proc.) (16.7 ns)	fc7.0 -tm c38 -O3 -ep 2 -ds -is .	62	222	240
Alliant FX/2800-200 (6 proc)	fortran 1.1.27 -O -inline	21	148	240
Convex SPP-1000(1 proc) 100 MHz	+OP3 -Wl,-aarchive -WP,-nv	41	123	200
CONVEX C-3440 (4 proc.)	fc7.0 fc -O3 -ep 4 -ds -is .	34	172	200
CONVEX C-3240 (4 proc.)	fc -O3 -ep 2 -uo -pp=fcpp1 -is .	26	171	200
CONVEX C-240 (4 proc.)	6.1 -O3 -ep2 -uo -pp=fcpp1 -is .	26	166	200
Alliant FX/80 (8 proc.)	-O -DAS -inline	10	69	188
Alliant FX/80 (7 proc.)	-O -DAS -inline	9.5	63	165
Alliant FX/2800-200 (4 proc)	fortran 1.1.27 -O -inline	17	94	160
CONVEX C-3430 (3 proc.)	fc7.0 fc -O3 -ep 3 -ds -is .	32	132	150
CONVEX C-3230 (3 proc.)	fc -O3 -ep 2 -uo -pp=fcpp1 -is .	26	132	150
CONVEX C-230 (3 proc.)	6.1 -O3 -ep2 -uo -pp=fcpp1 -is .	26	128	150
CONVEX C-3810 (1 proc.) (16.7 ns)	fc7.0 -tm c38 -O2 -is .	44	113	120
Alliant FX/80 (5 proc.)	-O -DAS -inline	8.1	49	118
CONVEX C-3420 (2 proc.)	fc7.0 fc -O3 -ep 2 -ds -is .	27	90	100
CONVEX C-3220 (2 proc.)	fc -O3 -ep 2 -uo -pp=fcpp1 -is .	22	89	100
CONVEX C-220 (2 proc.)	6.1 -O3 -ep2 -uo -pp=fcpp1 -is .	22	87	100
Alliant FX/80 (4 proc.)	-O -DAS -inline	7.2	33	94
Alliant FX/40 (4 proc.)	-O -DAS -inline	6.7	33	94
Alliant FX/2800-200 (2 proc)	fortran 1.1.27 -O -inline	10	53	80
Alliant FX/80 (3 proc.)	-O -DAS -inline	5.9	32	71
Alliant FX/40 (3 proc.)	-O -DAS -inline	5.6	27	71
CONVEX C-3410 (1 proc.)	fc7.0 fc -O2 -is .	19	47	50
CONVEX C-3210 (1 proc.)	fc -O2 -uo -pp=fcpp1 -is .	17	44	50
CONVEX C-210 (1 proc.)	6.1 -O2 -uo -pp=fcpp1 -is .	17	44	50
Alliant FX/2800 210 (1 proc)	fortran 1.3.02 -Ovg -inline	25	34	50
Alliant FX/80 (2 proc.)	-O -DAS -inline	4.4	22	47
Alliant FX/40 (2 proc.)	-O -DAS -inline	4.3	19	47
Alliant FX/4 (4 proc.)	-O -DAS -inline	6.4	21	47
Alliant FX/2800-200 (1 proc)	fortran 1.1.27 -O -inline	6.4	28	40
CONVEX C-130	Fortran 4.0	7.2	31	36
Alliant FX/4 (3 proc.)	-O -DAS -inline	5.1	17	35
Alliant FX/80 (1 proc.)	-O -DAS -inline	2.4	12	24
Alliant FX/40 (1 proc.)	-O -DAS -inline	2.4	10	24
Alliant FX/4 (2 proc.)	-O -DAS -inline	3.8	12	24

PDS: Papers & Notes

Papers

The following are papers related to benchmarks and benchmarking methodologies.

The Complete Linpack Report *(in Postscript)*
Date: July 22, 1994
Jack J. Dongarra / University of Tennese / dongarra@cs.utk.edu

The Development and Implementation of a Performance Database Server *(in Postscript)*
University of Tennessee Technical Report CS−93−195, August 1993
A description of the implementation of PDS
Brian LaRose / University of Tennessee / larose@cs.utk.edu

Notes

The following notes have been provided by the authors to aid in the understanding of the published performance metrics.

Dhrystone
Date: July 22, 1994
Alfred Aburto / Naval Ocean Systems Center, San Diego / aburto@nosc.mil

Fhourstone
Date: July 22, 1994
Alfred Aburto / Naval Ocean Systems Center, San Diego / aburto@nosc.mil

Flops
Date: July 22, 1994
Alfred Aburto / Naval Ocean Systems Center, San Diego / aburto@nosc.mil

Hanoi
Date: July 22, 1994
Alfred Aburto / Naval Ocean Systems Center, San Diego / aburto@nosc.mil

Heapsort
Date: July 22, 1994
Alfred Aburto / Naval Ocean Systems Center, San Diego / aburto@nosc.mil

NSieve
Date: July 22, 1994
Alfred Aburto / Naval Ocean Systems Center, San Diego / aburto@nosc.mil

Sim
Date: July 22, 1994
Alfred Aburto / Naval Ocean Systems Center, San Diego / aburto@nosc.mil

TFFTdp
Date: July 22, 1994
Alfred Aburto / Naval Ocean Systems Center, San Diego / aburto@nosc.mil

PDS: Search

Select desired benchmark:

Enter the keywords on which to search.

Type of search: OR AND

Search string:

If you have questions or concerns, please send mail to utpds@cs.utk.edu

PDS: Search Query Results

SPEC Benchmark CFP92 Summary

SPECfp92 (Volume 5, Issue 2, June 1993)

Company Name	Machine Name	geometric mean SPECfp92	013 spice2g6 SPEC ratio	015 doduc SPEC ratio	034 mdljdp2 SPEC ratio	039 wave5 SPEC ratio	047 tomcatv SPEC ratio	048 ora SPEC ratio	052 alvinn SPEC ratio	056 ear SPEC ratio	077 mdljsp2 SPEC ratio	078 swm256 SPEC ratio	089 su2cor SPEC ratio	090 [illegible] SPEC ratio	093 [illegible] SPEC ratio	094 [illegible] SPEC ratio
Sun Microsystems Inc.	SPARCstation 10 Model 51	83.0	56.2	81.2	86.4	57.9	77.3	160.3	175.6	95.0	42.2	43.7	113.3	79.1	93.5	101.7
Sun Microsystems Inc.	SPARCstation 10 Model 40	60.2	38.4	56.5	65.0	39.8	71.8	127.9	123.2	78.8	33.2	46.4	55.0	49.8	60.8	63.3

SPEC Benchmark CINT92 Summary

SPECint92 (Volume 5, Issue 2, June 1993)

Company Name	Machine Name	geometric mean SPECint92	008 espresso SPEC ratio	022 li SPEC ratio	023 eqntott SPEC ratio	026 compress SPEC ratio	072 sc SPEC ratio	085 gcc SPEC ratio
Sun Microsystems Inc.	SPARCstation 10 Model 51	65.2	58.7	63.3	108.9	38.2	93.0	53.4
Sun Microsystems Inc.	SPARCstation 10 Model 40	50.2	44.9	48.9	82.7	29.0	71.0	42.7

SPEC Benchmark CFP92 Homogeneous Capacity Summary

SPECrate_fp92 (Volume 5, Issue 4, December 1993)

Company Name	Machine Name	geometric mean rate_fp92	013 spice2g6 SPEC rate	015 doduc SPEC rate	034 mdljdp2 SPEC rate	039 wave5 SPEC rate	047 tomcatv SPEC rate	048 ora SPEC rate	052 alvinn SPEC rate	056 ear SPEC rate	077 mdljsp2 SPEC rate	078 swm256 SPEC rate	089 su2cor SPEC rate	090 hydro2d SPEC rate	093 nasa7 SPEC rate	094 fpppp SPEC rate
Sun Microsystems Inc.	SPARCstation/server 10 Model 514	5809	3230	5882	7156	4229	3308	14977	12799	8118	3739	2103	6837	6665	5713	8234

SPEC Benchmark CFP92 Homogeneous Capacity Summary

SPECrate_fp92 (Volume 5, Issue 2, June 1993)

geometric mean	013 spice2g6 SPEC	015 doduc SPEC	034 mdljdp2 SPEC	039 wave5 SPEC	047 tomcatv SPEC	048 ora SPEC	052 alvinn SPEC	056 ear SPEC	077 mdljsp2 SPEC	078 swm256 SPEC	089 su2cor SPEC	090 hydro2d SPEC	093 nasa7 SPEC	094 fpppp SPEC

Netlib Performance Database

This database contains benchmark results for linpack, hanoi, fhourstone, etc. Clicking here will allow you to query the database.

Related information may be found in the following places:

- benchmark
  This Netlib library contains benchmark programs and report, such as linpack, livermore loops, nas.
- machines
  This Netlib library contains information on high−performance computers
- PARKBENCH
  This library contains information on the parkbench workgroup to develop a benchmark for distributed memory comptuers.