HINT: A New Way to Measure Computer Performance

John L. Gustafson and Quinn O Snell

Iowa State
Sun Microsystems
ClearSpeed
AMD
Ceranovo

Benchmark Strategies

1. Fixed-Computation Benchmarks

   - Measure the time taken

2. Fixed-Time Benchmarks

   - Measure the amount of computation performed

3. Variable-Computation and variable-time Benchmarks

   - Measure some aspect of performance that is a function of the computation and the execution time eg: quality of the answer

Fixed-Computation Benchmarks

1. Measure the computer's speed
2. In physical world, speed is distance/time
3. In computer world, distance analogous to operations or instructions
4. Measure time for the computations
5. Execution time
6. MIPS
7. MFLOPS
8. SPEC Benchmarks

Fixed-Time Benchmarks

1. Basic idea similar to - Count for 1 minute (how much did you reach)?
2. Walk for an hour –how long did you reach?
3. At the end of the fixed-time, measure the total amount of computation
4. Find prime numbers – how many numbers did you find?
5. SLALOM (Gustafson)

SLALOM

First Benchmark to do fixed-time variable-computation strategy

Based on a scientific application to compute radiosity

Radiosity is a global illumination algorithm in computer graphics

Accuracy of the answer computed in 1 min

Benchmark did not specify a particular algorithm

Defined the accuracy of the answer as the number of “patches” or areas into which a geometric shape was subdivided in the 1-min interval.

SLALOM - Weaknesses

Loosely defined problem statement

Clever programming became important

Original complexity – O(n^3)

Later O(n^2)

Eventually O (n logn)

Non-linear complexity of the algorithm makes the performance metric non-linear

You can’t say that a system that computes 2N pacthes is twice as fast as one that computes N patches

SLALOM - Weaknesses

SLALOM – unrealistically forgiving of machines with inadequate memory bandwidth

SLALOM has storage demands that scaled, but it failed to run for 1 min on computers with insufficient memory relative to arithmetic speed.

Low ease of use – converting to parallel versions took huge amounts of time. SLALOM started with 1000 lines of FORTRAN/C, expanded with better alg to 8000 lines;

SLALOM – led to - HINT

Variable-computation, Variable-time strategy

HINT stands for Hierarchical INTegration

Produces a speed measure called QUIPS

QUIPS = Quality Improvement Per Second

HINT fixes nether time nor problem size

Objective: Use interval subdivision to find rational bounds on the area under curve in the x-y plane

QUIPS curve

NetQUIPS

HINT

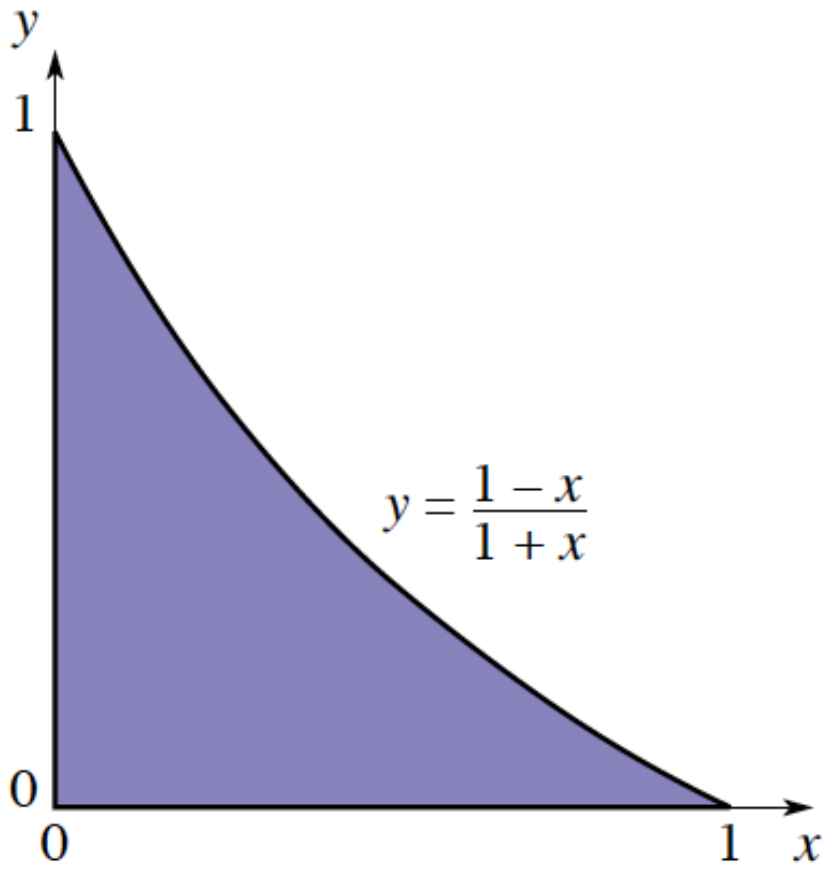
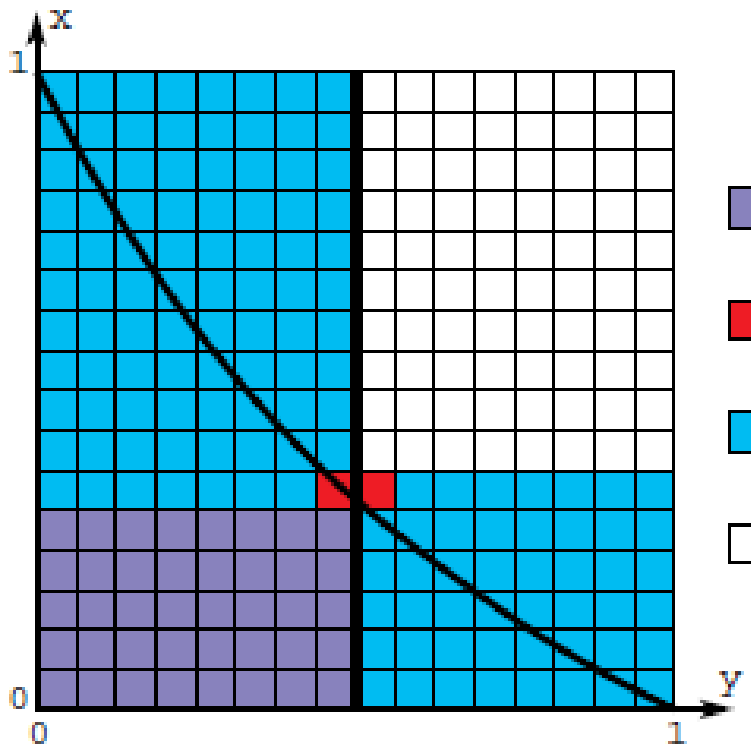


Fig. 1. Area to be bounded by HINT



$5/16 < f(1/2) < 6/16$

Known to contribute to lower bound

Limited by arithmetic precision

Available for further refinement

Known not to contribute to upper bound

Quality = 1/(u-l), where u = estimate of upper bound
l= estimate of lower bound

Initially, u=256; l= 0

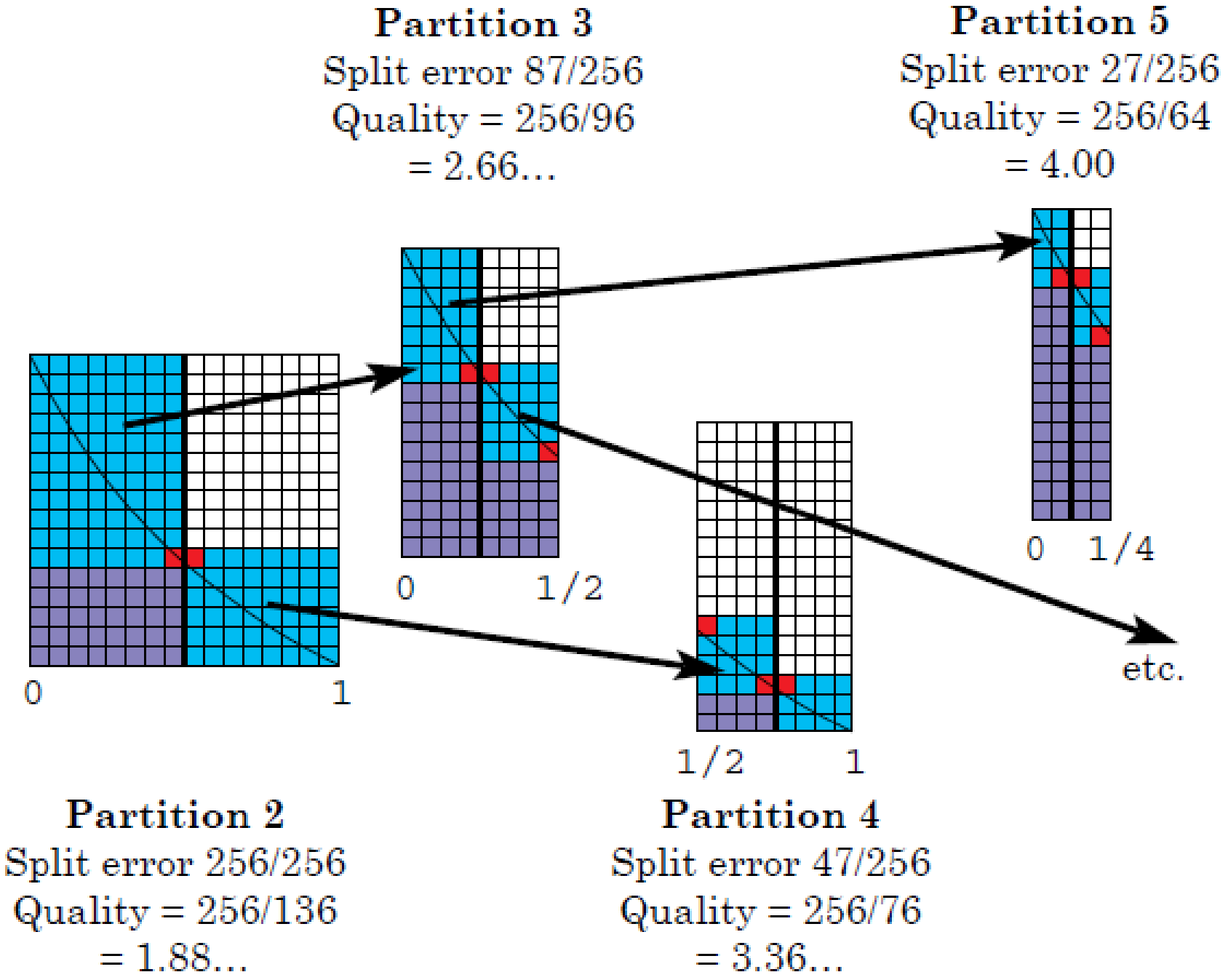


Fig. 3. Sequence of hierarchical refinement of integral bounds

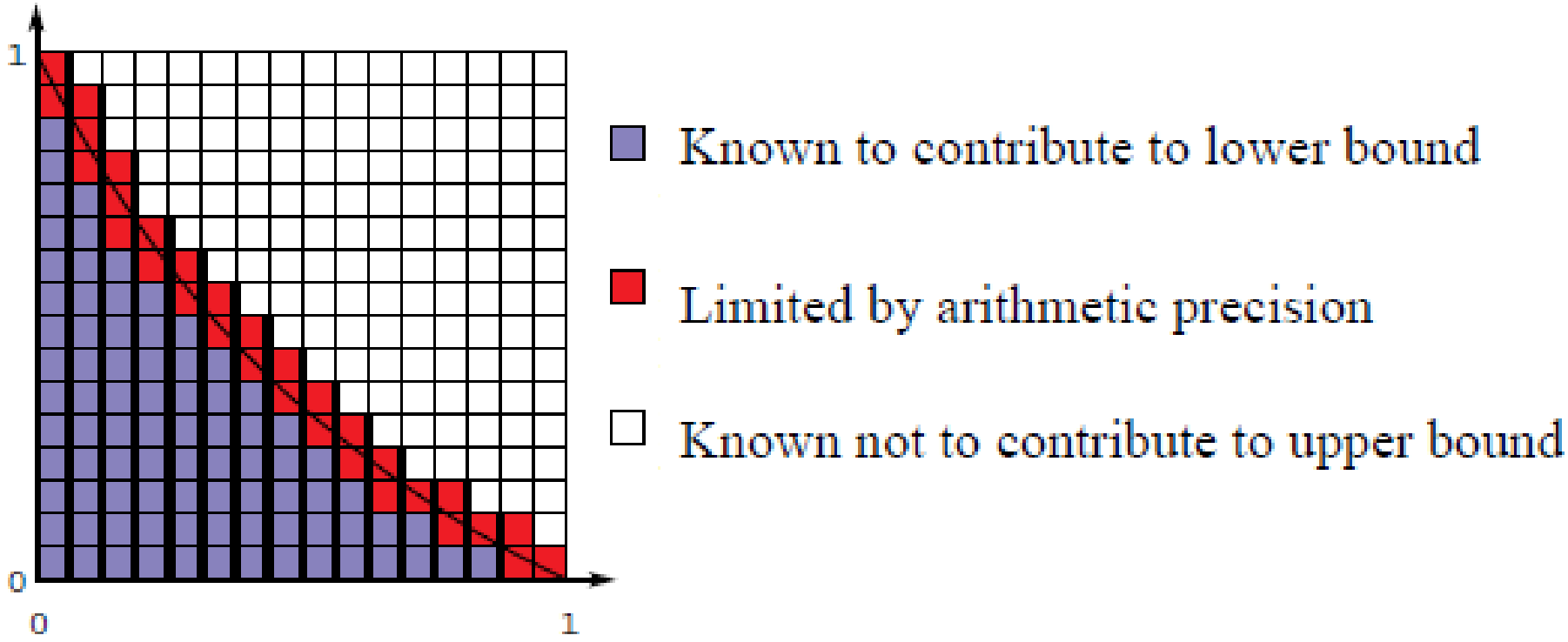


Fig. 4. Precision-limited last iteration, 8-bit data

A compilation of the HINT kernel for a conventional processor revealed the following operation distribution for indices and data:

Index operations:	Data operations:
39 adds or subtracts	69 fetches or stores
16 fetches or stores	24 adds or subtracts
6 shifts	10 multiplies
3 conditional branches	2 conditional branches
2 multiplies	2 divides

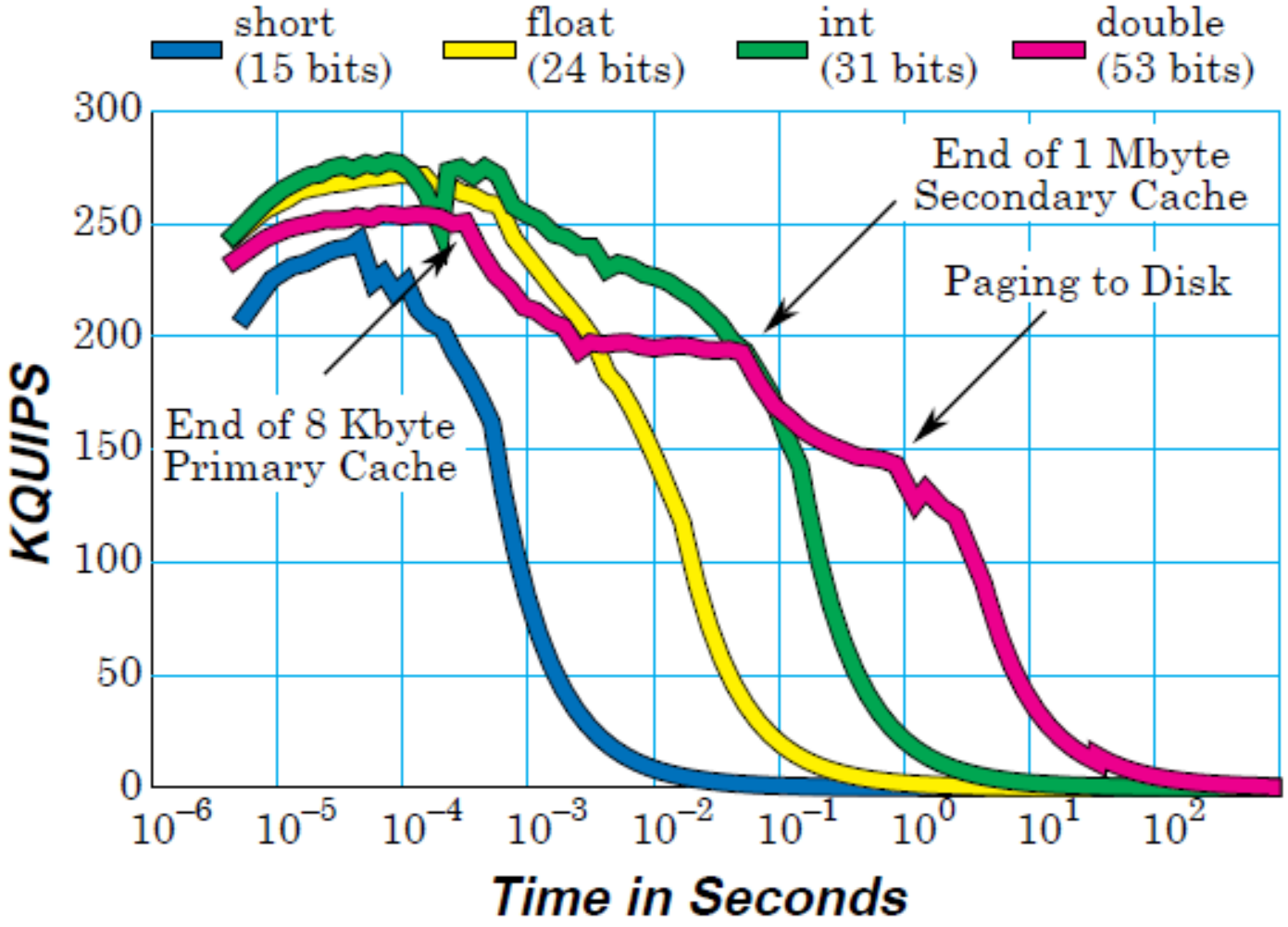


Fig. 5. Comparison of Different Precisions

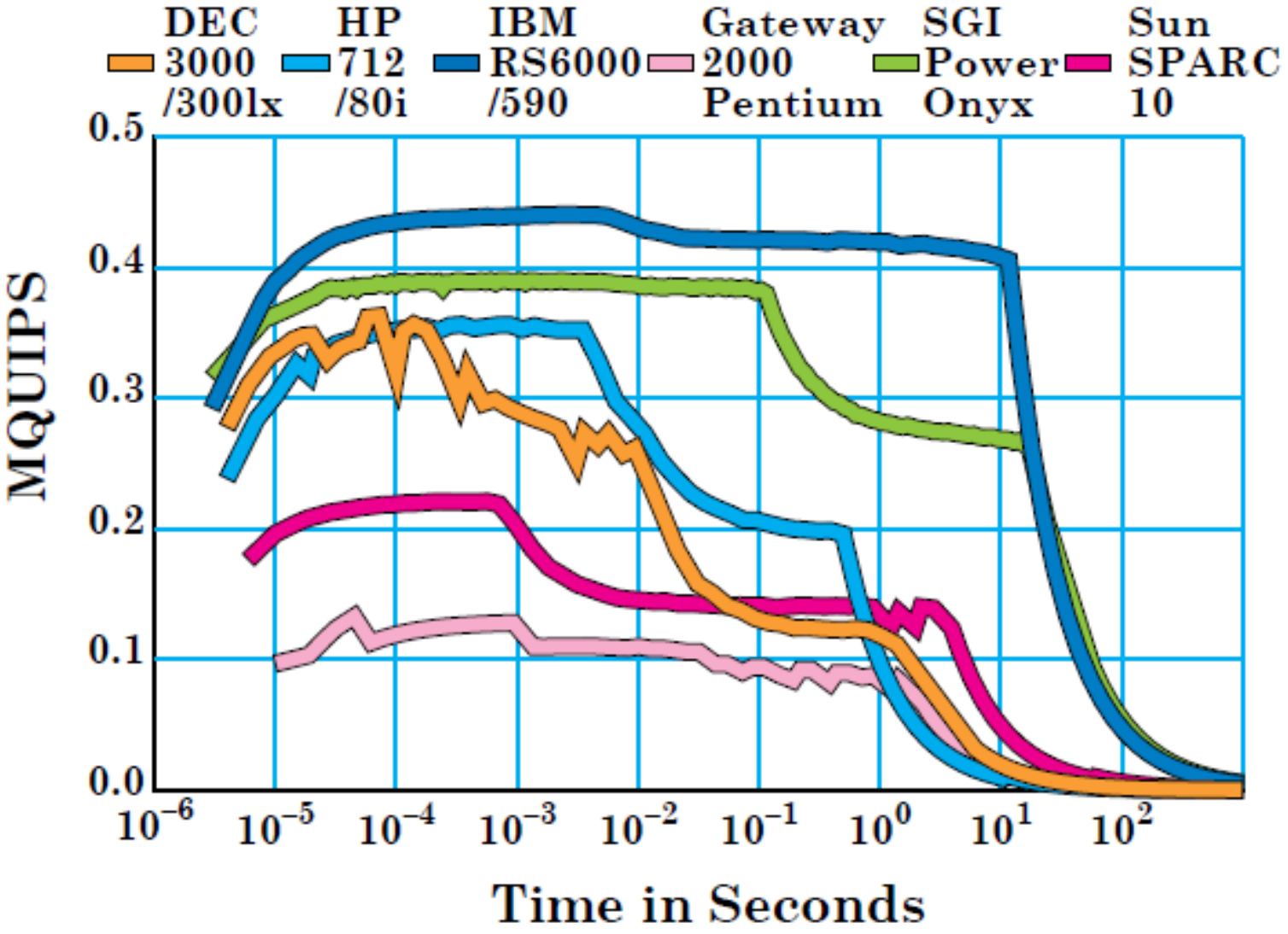


Fig. 6. Comparison of Various Workstations

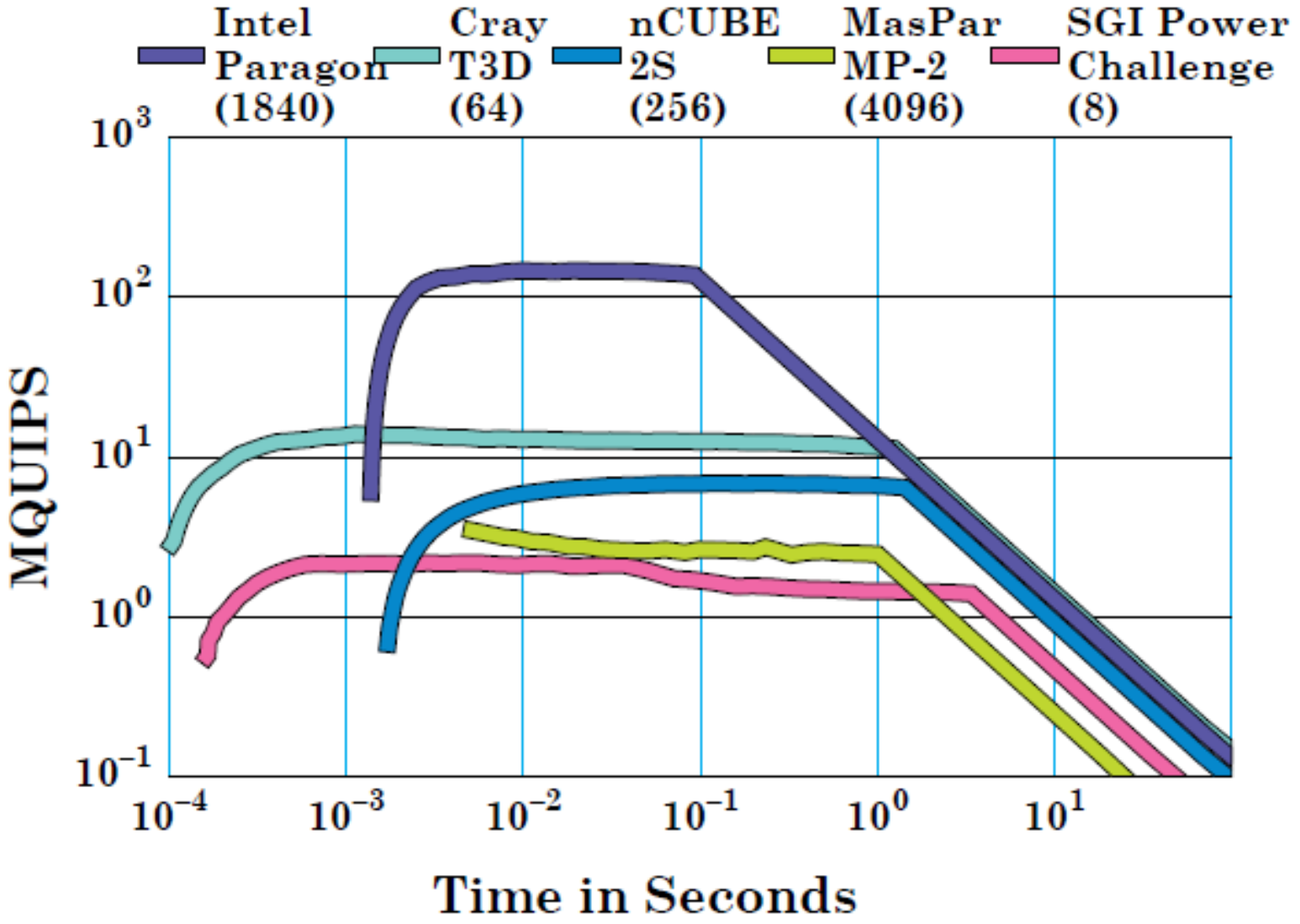


Fig. 7. Comparison of Several Parallel Systems

Table 1. Net QUIPS ratings

Vendor, Hardware	No. of PE's	Net MQUIPS, data type	Operating System	Compiler and Command Options
Intel Paragon	1840 512 64 32 16 8 4 2	633. fp 249. 46.2 25.7 13.5 7.07 3.76 2.03	SUNMOS	icc -04 -knoieee -Mvect
Intel Paragon	32	12.6 fp	OSF/1 1.0.4	cc -03 -knoieee
nCUBE 2S	256 128 64 32 16 8 4 2 1	35.8 fp 18.4 9.42 4.84 2.49 1.29 0.67 0.36 0.26	IRIX 4.0.5 + Vertex 3.2	ncc -02 - ncube2s

nCUBE2	128 64 32 16 8 4 2 1	12.6 fp 7.81 4.00 2.06 1.07 0.57 0.33 0.20	IRIX 4.0.5 + Vertex 3.2	ncc -0
SGI Challenge L R4400/150	8 4 1	17.5 fp 10.2 4.62	IRIX 5.2	cc v3.18 -03 -sopt
MasPar MP-1	16384	16.5 fp	ULTRIX 4.3	mpl
MasPar MP-2	4096	15.7 fp	ULTRIX 4.3	mpl
HP 712/80i	1	3.48 fp	HP-UX 9.05	gcc v2.5.8 -03
DEC 3000/300L	1	3.39 fp	OSF/1 1.3	cc -03
SGI Indy SC R4000/100	1	2.70 fp	IRIX 5.2	cc v3.18 -03 -sopt
Sun SPARC 10	1	2.34 fp	SunOS 5.3	gcc v2.5.8 -03

IBM PC Pentium	1	2.09 int	MS-DOS 5.0	gcc 2.5.7 -03
SGI Indy PC R4000/100	1	1.86 int	IRIX 5.2	cc v3.18 -03
DEC 5000/240	1	1.31	ULTRIX 4.3	cc -03
SGI Indigo R3000/33	1	0.97 fp	IRIX 5.2	cc v3.18 -03
IBM PC 486/50	1	0.82 int	MS DOS 5.0	gcc 2.5.7 -03
COMPAQ Contura Aero 486SX/25	1	0.38 int	MS-DOS 5.0	gcc 2.5.7 -03
Macintosh Quadra 840AV full opt.	1	0.27 int	MacOS 7.1	MPW C
Mac intosh Powerbook 520c full opt.	1	0.13 int	MacOS 7.1	MPW C

Net QUIPS

To satisfy thirst for single number

You can have QUIPS curves with time on x-axis or memory capacity on x-axis, but marketing folks want single number

Area under the QUIPS curve (plotted on log time scale)

- $$\text{Net QUIPS} = \int_{\log(t_0)} \text{QUIPS}(t)\, d(\log t)$$

$$= \int_{\log(t_0)} Q(t)/t\, d(\log t) = \int_{t_0} Q(t)/t^2\, dt$$

Cost of moving bits vs compute – From Prof. Bill Dally's paper

Table 1. Technology and circuit projections for processor chip components.

	2010	2017	
Process technology	**40 nm**	**10 nm, high frequency**	**10 nm, low voltage**
V_{DD} (nominal)	0.9 V	0.75 V	0.65 V
Frequency target	1.6 GHz	2.5 GHz	2 GHz
Double-precision fused-multiply add (DFMA) energy	50 picojoules (pJ)	8.7 pJ	6.5 pJ
64-bit read from an 8-Kbyte static RAM (SRAM)	14 pJ	2.4 pJ	1.8 pJ
Wire energy (per transition)	240 femtojoules (fJ) per bit per mm	150 fJ/bit/mm	115 fJ/bit/mm
Wire energy (256 bits, 10 mm)	310 pJ	200 pJ	150 pJ