dup

Massively Parallel Computers: Why Not Parallel Computers for the Masses?

Gordon Bell

Abstract

During the 1980s the computers engineering and science research community generally ignored parallel processing. With a focus on high performance computing embodied in the massive 1990s High Performance Computing and Communications (HPCC) program that has the short-term, teraflop peak performance goal using a network of thousands of computers, everyone with even passing interest in parallelism is involved with the massive parallelism "gold rush". Funding-wise the situation is bright; applications-wise massive parallelism is microscopic. While there are several programming models, the mainline is data parallel Fortran. However, new algorithms are required, negating the decades of progress in algorithms. Thus, utility will no doubt be the Achilles Heal of massive parallelism.

The Teraflop: 1992

The quest for the Teraflops Supercomputer to operate at a peak speed of 10^{12} floating-point operations per second is almost a decade old, and only one, three-year computer generation from being fulfilled. To accelerate its development would require an ultracomputer. First generation, ultracomputers are networked computers using switches that interconnect 1000s of computers to form a multicomputer, and cost $50 to $300 million in 1992. These scalable computers are also classified as massively parallel since they can be configured to have more than 1000 processing elements in 1992. Unfortunately, such computers are specialized since only highly parallel, coarse grain applications, requiring algorithm and program development, can exploit them. Government purchase of such computers would be foolish since waiting three years will allow computers with a peak speed of a teraflops to be purchased at supercomputer prices ($30 million) caused by advancements in semiconductors and the intense competition resulting in "commodity supercomputing". More importantly, substantially better computers will be available in 1995 in the supercomputer price range if the funding that would be wasted in buying such computers is spent on training and software to exploit their power.

In 1989 I described the situation in high performance computers including several parallel architectures that could deliver teraflop power by 1995, but with no price constraint. I felt SIMDs and multicomputers could achieve this goal. A shared memory multiprocessor looked infeasible then. Traditional, multiple vector processor supercomputers such as Crays would simply not evolve to a teraflop until 2000. Here's what happened.

1. During the first half of 1992, NEC's four processor SX3 is the fastest computer, delivering 90% of its peak 22 Glops for the Linpeak benchmark, and Cray's 16 processor YMP C90 has the greatest throughput.

2. The SIMD hardware approach of Thinking Machines was abandoned because it was only suitable for a few, very large scale problems, barely multiprogrammed, and uneconomical for workloads. It's unclear whether large SIMDs are "generation" scalable, and they are clearly not "size" scalable. The CM2 achieved a 10 Gigaflop of performance for some large problems.

3. Ultracomputer sized, scalable multicomputers (smC) were introduced by Intel and Thinking Machines, using "Killer" CMOS, 32-bit microprocessors that offer over 100 Gflops in the supercomputer price range. These product introductions join multicomputers from Alliant (now defunct), AT&T, IBM, Intel, Meiko, Mercury, NCUBE, Parsytec, Transtech, etc. At least Convex, Cray, Fujitsu, IBM, and NEC are working on new generation smCs that use 64-bit processors. The Intel Delta multicomputer with 500 computers achieved a peak speed of over 10 Glops and routinely achieves several Glops on real applications using O(100) computers.

   By 1995, this score of efforts, together with the evolution of fast, LAN-connected workstations will create "commodity supercomputing". Workstation clusters formed by interconnecting high speed workstations via new high speed, low overhead switches, in lieu of special purpose multicomputers are advocated.

Massively Parallel Computers: Why Not Parallel Computers for the Masses?

Gordon Bell

Abstract

During the 1980s the computers engineering and science research community generally ignored parallel processing. With a focus on high performance computing embodied in the massive 1990s High Performance Computing and Communications (HPCC) program that has the short-term, teraflop peak performance goal using a network of thousands of computers, everyone with even passing interest in parallelism is involved with the massive parallelism "gold rush". Funding-wise the situation is bright; applications-wise massive parallelism is microscopic. While there are several programming models, the mainline is data parallel Fortran. However, new algorithms are required, negating the decades of progress in algorithms. Thus, utility will no doubt be the Achilles Heal of massive parallelism.

The Teraflop: 1992

The quest for the Teraflops Supercomputer to operate at a peak speed of 10^{12} floating-point operations per second is almost a decade old, and only one, three-year computer generation from being fulfilled. To accelerate its development would require an ultracomputer. First generation, ultracomputers are networked computers using switches that interconnect 1000s of computers to form a multicomputer, and cost $50 to $300 million in 1992. These scalable computers are also classified as massively parallel since they can be configured to have more than 1000 processing elements in 1992. Unfortunately, such computers are specialized since only highly parallel, coarse grain applications, requiring algorithm and program development, can exploit them. Government purchase of such computers would be foolish since waiting three years will allow computers with a peak speed of a teraflops to be purchased at supercomputer prices ($30 million) caused by advancements in semiconductors and the intense competition resulting in "commodity supercomputing". More importantly, substantially better computers will be available in 1995 in the supercomputer price range if the funding that would be wasted in buying such computers is spent on training and software to exploit their power.

In 1989 I described the situation in high performance computers including several parallel architectures that could deliver teraflop power by 1995, but with no price constraint. I felt SIMDs and multicomputers could achieve this goal. A shared memory multiprocessor looked infeasible then. Traditional, multiple vector processor supercomputers such as Crays would simply not evolve to a teraflop until 2000. Here's what happened.

1. During the first half of 1992, NEC's four processor SX3 is the fastest computer, delivering 90% of its peak 22 Glops for the Linpeak benchmark, and Cray's 16 processor YMP C90 has the greatest throughput.

2. The SIMD hardware approach of Thinking Machines was abandoned because it was only suitable for a few, very large scale problems, barely multiprogrammed, and uneconomical for workloads. It's unclear whether large SIMDs are "generation" scalable, and they are clearly not "size" scalable. The CM2 achieved a 10 Gigaflop of performance for some large problems.

3. Ultracomputer sized, scalable multicomputers (smC) were introduced by Intel and Thinking Machines, using "Killer" CMOS, 32-bit microprocessors that offer over 100 Gflops in the supercomputer price range. These product introductions join multicomputers from Alliant (now defunct), AT&T, IBM, Intel, Meiko, Mercury, NCUBE, Parsytec, Transtech, etc. At least Convex, Cray, Fujitsu, IBM, and NEC are working on new generation smCs that use 64-bit processors. The Intel Delta multicomputer with 500 computers achieved a peak speed of over 10 Glops and routinely achieves several Glops on real applications using O(100) computers.

By 1995, this score of efforts, together with the evolution of fast, LAN-connected workstations will create "commodity supercomputing". Workstation clusters formed by interconnecting high speed workstations via new high speed, low overhead switches, in lieu of special purpose multicomputers are advocated.

4. Kendall Square Research introduced their KSR 1 scalable, shared memory multiprocessors (smP) with 1088 64-bit microprocessors. It provides a sequentially consistent memory and programming model, proving that smPs are feasible. A multiprocessor provides the greatest and most flexible ability for workload since any processor can be deployed on either scalar or parallel (e.g. vector) applications, and is general purpose, being equally useful for scientific and commercial processing, including transaction processing, databases, real time, and command and control. The KSR machine is most likely the blueprint for future scalable, massively parallel computers.

   University research is directed primarily at smPs and software to enable smC to operate as smPs.

The net result of the quest for parallelism as chronicled by the Gordon Bell Prize is that applications evolved from 1(1987) to 10 (1990) to my estimate of 100 Glops (1993) in roughly 3 year, or 1 generation, increments or 115% per year and will most likely achieve 1 teraflop in 1995, for a factor of 1000 increase. Moore's law applied to speed accounts for 60% annual increase in performance, a factor of 4 every 3 years; the remaining factor of 2.5 or 36% per year is due to parallelism. This is in line with a prognostication I made in 1985 about getting a factor of 100 speedup for parallelism in a decade would be a major accomplishment. Supers evolve with a 4-5 year gestation period and micro-based scalable computers have a 3 year gestation. In order to reach a teraflop in 1993, it is necessary to spend $250 million for an ultracomputer. Meiko has announced a multicomputer capable of delivering 1 Teraflop (32 bits) for $50 million in 1993. In 1992, a flop (10^{15} flops) ultracomputers, costing a half billion dollars do not look feasible by 2001.

The irony of the teraflop quest is that programming may not change very much even though virtually all programs must be rewritten to exploit the high degree of parallelism that is required to achieve peak speed and for efficient operation of the course grain, scalable computers. Scientists and engineers will use just another dialect of Fortran, i.e. High Performance Fortran (HPF) that supports data parallelism. Even though a dialect, algorithms and programs must be reconsidered to deal with the idiosyncrasies inherent in running on large, scalable multicomputers. The flaw in the 1992-1995 generation of multicomputers is that large codes are likely to run very poorly and will have to be completely rewritten even if the main loops are parallelized. Montry conjectures: "the degree of parallelism in a program varies with the size of the problem (program scalability) and inversely with its size". For example, if with modest effort, 99% of a large program representing 5-10% of the program size can be parallelized to run infinitely fast, it is likely that the remaining 90% of the code will slow down a factor of 10 due to the computer's idiosyncrasies, resulting in only a factor of 10 speedup.

The 1992-1995 Generation of Computers

By mid-1992 a completely new generation of computers have been introduced. Understanding a new generation and redesigning it to be less flawed takes at least three years. Understanding this generation should make it possible to build the next generation supercomputer class machine, that would reach a teraflop of peak power for a few, large scale applications by the end of 1995. Unlike previous computer generations that have been able to draw on experience from other computer classes and user experience (the market) to create their next generation, virtually no experience exists for the design of next generation massively parallel computers. This experience can only come with use. So far the lessons have only been what most of us have believed: a shared memory is needed, and that multiprocessors provide significant advantages, and are thus the main line of computer evolution (Bell, 1991).

Table 1 shows six alternatives for high performance computing, ranging from two traditional supers, one smP, and three "commodity supers" or smCs, including 1000 workstations. Three metrics characterize a computer's performance and workload abilities. Linpeak is the operation rate for solving a system of linear equations and is the best case for a highly parallel application. Large, well programmed applications typically run at 1/4-1/2 this rate. Linpack 1K x 1K is typical of problems solved on supercomputers in 1992. The Livermore Fortran Kernels (LFK) harmonic mean for 24 loops and 3 sizes, is used to characterize a numerical computer's ability, and is the worst-case rating for a computer as it represents an untuned workload.

New generation, traditional or "true" multiple vector processor supercomputers provide 1/4 to 1/8th the peak power of the smCs to be delivered in 1992. "True" supercomputers use the Cray design formula: ECL circuits and dense packaging technology to reduce size, allow the fastest clock; one or more pipelined vector units with each processor provide peak processing for a Fortran program; and multiple vector processors communicate via a switch to a common, shared memory to handle large workloads and parallel processing. Because of the dense physical packaging of high power chips and relatively low density of the 100,000 gate ECL chips, the inherent cost per operation for a supercomputer is roughly 500 -1000 peak flops/$ or 4 - 10 times greater than simply packaged, 2 million transistor "killer" CMOS microprocessors that go into leading edge workstations (5000 peak flops/$). True supercomputers are not in the teraflops race, even though they are certain to provide most of the supercomputing capacity until 1995.

Whether traditional supercomputers or massively parallel computers provide more computing, measured in flops/month by 1995 is the object of a bet between the author and Danny Hillis of Thinking Machines (Hennessy

Table 1a. Physical Characteristics of 1992 Supercomputing Alternatives

Machine	Proc's	Clk	Peak	Price	Mp.size	I/O.Bw
	#	Mhz	Gflops	$M	Gbytes	Gbytes/s
Traditional supercomputers						
Cray C90	16	250	16	30	2(16)**	13.6
NEC SX3 (R series)	4	400	25.6	25??	8(64)	5.4
Scalable Multiprocessors						
KSR 1	1088	20	43.5	30	34.8	15.3
Scalable Multicomputers						
Intel Paragon§	4096	50	300	55	128	
TMC CM5†	Cc+ Cio+1024	33	128	30	32+	
Workstations						
DEC alpha	1024	150	150	20	32	100

*Author's estimate
**Fast access blocked, secondary memory
§Available: Q1 1993. Four processors form a multiprocessor node; a fifth processor handles communication.
†Cc := control computer; Cio := i/o computers;

Table 1b. Workload Characteristics of 1992 Supercomputing Alternatives

Machine	Streams	Linpeak	Lin1K	LFKWorkload
	#jobs	Gflops(size)	Gflops	Mflops
Traditional supercomputers				
Cray C90	16	13.7(4K)	9.7	16 x 44
NEC SX3 (345 Mhz clock)	4	20 (6K)	13.4	4 x 39
Scalable Multiprocessors				
KSR 1	1088		513(32Proc.)	1088 x 6.6
Scalable Multicomputers				
Intel Paragon	1024	13.9(25K)/.5K	??	1K x 6*
TMC CM5	≤32 Cc's	70*	32 x ??	≤32 x 6*
LAN-connected Workstations				
DEC Alpha	1024		64	1K x 15

*Author's estimate
??Unavailable

Table 2. Contemporary Microprocessor Performance

Micro	Year	Clock	i-Spec	f-Spec	Spec	Linpack	LapeakLFK(hm)		Pk
		(Mhz)	(Specmarks*)---------			(Mflops)	--------		
DEC VAX780	78.2	5*	1	1	1	0.15		0.16	1
DEC Alpha	92.2	200			150†		85†	20†	200†
Fujistu-VP	92.4	50				>50	95	12.5	108
HP PA	92.2	100	-	-	138	56	67	-	200
HP PA	91.1	66	52	78	102	24		13.3	66
IBM RS6000	91.2	42	33	120	72	27	70	14.5	83
Intel 486 PC	91.3	50	28	15	19	2.0	-	1.8	
MIPS R3000	88.3	25			18	4.2	7	3.6	8
MIPS R4000	92.2	100**	60	77	70	17.5	36	11.5	50
SUN Sparc 2	91.3	40	22	27	25	4.1	-	-	-
1995 Micro	95	200-400			300				400-800

*CISC architecture. A comparable RISC architecture would operate at approx. 2 Mhz.
**External clock rate is 50 Mhz.
† Estimate

and Patterson, 1990). Scalable multicomputers (smCs) are applicable to coarse grain, highly parallel codes and someone must invent new algorithms and write newprograms. Universities are rewarded with grants, papers, and the production of knowledge. Hence, they are a key to utilizing coarse grain, parallel computers. With pressure to aid industry, the Department of Energy laboratories can embrace massive parallelism in order to maintain budgets and staffs. On the other hand, any organization concerned with cost-effectiveness, simply can't afford the rewrite effort for one-of-a-kind computers unless they obtain uniquely competitive capabilities and are prepared to support unique software on transient computers.

"Killer" CMOS Micros for building scalable computers

Progress toward the affordable teraflop using "killer" CMOS micros is determined by advances in microprocessor speeds. The projection (Bell, 1989) that microprocessors would improve in speed at a 60% per year rate following Moore's Law appears to be possible for the next few years (Table 2). Moore's Law that stated that semiconductor density would quadruple every 3 years. This explains memory chip size evolution. Memory size can grow proportionally with processor performance even though the memory bandwidth is not keeping up. Since clock speed only improves at 25% per year (a doubling in 3 years), the additional speed must come from architectural features (e.g. superscalar or wider words, larger cache memories, and vector processing).

The leading edge microprocessors described at the 1992 International Solid State Circuits Conference included: a microprocessor based on Digital's Alpha architecture with a 150 or 200 Mhz clock rate; and the Fujitsu 108 (64) | 216 (32-bit) Mflop Vector Processor chip. The Fujitsu chip would provide the best performance for traditional supercomputer oriented problems. Perhaps the most important improvement to enhance massive parallelism is the 64-bit address in order that a computer can have a large global address space. With 64-bit addresses and substantially faster networks, some of the limitations of message-passing multicomputers can be overcome. Daly's J-machine at MIT provides critical primitives for the processor architecture that deal with communication among computers such that operating system and library software for multicomputers can be written to enable them to look like multiprocessors that they are evolving to simulate. These primitives should provide for the elimination of message passing primitives that found their way into the programming paradigm and a return to a Fortran dialect.

In 1995, $20,000 distributed computing node microprocessors with peak speeds of 400-800 Mflops can provide 20,000-40,000 flops/$. For example, such chips are a factor of 12-25 times faster than the vector processor chips used in the CM5 and would be 4.5 - 9 times more cost-effective.

Scalability

The perception that a computer can grow forever has always been a design goal e.g. IBM System/360 (c1964) provided a 100:1 range. VAX existed at a range of 1000:1 over its lifetime. Size scalability simply means that a very large computer such as the ultracomputer can be built. Typical definitions fail to recognize cost, efficiency, and whether such a large scale computer is practical (affordable) in a reasonable time scale. Ideally, one would start with a single computer and buy more components as needed. System size scalability has been the main outcome of the search for the teraflop.

Similarly, when new processor technology increased performance, one would add new generation computers in a generations scalable fashion. All characteristics of a computer must scale proportionally: processing speed, memory speed and sizes, interconnect bandwidth and latency, I/O, and software overhead in order to be useful for a given application. Ordinary workstations provide some size and generation scalability, but are LAN-limited. By providing suitable high speed switching, workstation clusters can supply parallel computing power and are an alternative to scalable multicomputers.

Problem scalability is the ability of a problem, algorithm, or program to exist at a range of sizes so that it can be used efficiently and correctly on a given, scalable computer. In practical terms, problem scalability means that a program can be made large enough to operate efficiently on a computer with a given granularity.

Worlton (1992b) points out the need for a very large fraction, F, of a given program to be parallel, when using a large number of processors, N to obtain high efficiency, E(F,N).

$$E(F,N) = 1 / (F + N \times (1 - F))$$

Thus, scaling up slow processors is a losing proposition for a given fraction of parallelism. For 1000 processors F must be 0.999 parallel for 50% efficiency.

The critical question about size scalability is whether sufficient applications or problem scalability exists to merit procurement of a large scale systems.

Conjectures about the Future

In 1987, as Assistant Director of NSF's CISE, I argued that reasonable goals would be to achieve factors of 10 and 100 by times speedups due to parallelism, excluding vectorization, by 90 and 95 using conventional mPs and scalable computers. I also argued that the limits were primarily training because the microprocessor would provide the low cost, high performance components, including the option of idle, high speed workstations.

Thus, let me prognosticate:

1. The mainline general purpose computers will continue to be multiprocessors in three forms: supercomputers, mainframes, and scalable mPs. The current scalable, multicomputers will all evolve and become multiprocessors, but with limited coherent memories in their next generation.

2. Both mainframes and supers will have attached scalable multicomputer clusters, including workstation clusters with O(1000) computers to achieve the 100 Gflop level and to reduce the cost for bulk or massively parallel, computation.

3. Workstations will (should) supply a substantial amount of the parallel power, including those attached to supers and mainframes for massive parallelism. LLNL made the observation that it spends about three times as much on workstations that are only 15% utilized, as it does on supercomputers. By 1995, workstations could reach a peak of 500 Mflops, providing 25,000 flops per dollar or 10 times the projected cost-effectiveness of a super. This would mean that inherent in its spending, LLNL would have about 25 times more unused peak power in its workstations than it has in its central supercomputer or any massively parallel computers it might have.

4. The amount of parallelism will be 30-100, or achieving peak speeds of 10 Glops for the limited applications that run on the scalable multiprocessors. Thus, progress in massive parallelism[1] will be very slow.

5. Programming will be done in HPF Fortran. Programming environments such as Linda, PVM, and Parasoft's Express will be routinely used for large applications.

6. Training and applications will still be dominant limiters to massive parallelism.

7. The amount of parallelism for scientific applications that can run on massively parallel computers is comparatively small as shown in the following table giving my estimates about application parallelsim. Note that fast workstations and workstation clusters can handle a large fraction of applications.

Computer	degree of parallelism	% of programs
Scalar	no parallelism	60
Vector & smPs	fine grain	15
Vector/MP & smPs	medium grain	5
Scalable mCs (>>//)	coarse grain	5
W/S Clusters	very coarse grain	15

8. Algorithms have improved faster than clock over the last 15 years. Coarse grain computers are unlikely to be able to take advantage of these advances because they require new programs and new algorithms.

9. The cost and time to rewrite major applications for one-of-a-kind machines is sufficiently large to make them uneconomical. Each massively parallel computer exists as a hierarchy of bottlenecks, with no two machines having close enough characteristics to enable a program to be run on different computers without significant tuning and rewriting.

Thus, $1 invested in rewriting and maintaining software for a given machine buys exactly $1 of software. In comparison, $1 invested in workstation software buys $100-$10,000 software and up to $1 million for a PC. For example, software selling for $20K may cost $2M - $200M to write. Despite the existence of massively parallel computers, no major CFD, finite element, or computational chemistry software packages run as production codes.

Concerns About Real Progress

As an author of the network report and the HPCC, the teraflop search is of concern to me: a focus on unbalanced and "pap[2]" teraflops, using multicomputers with minimal focus on applications, training, and programmability; lack of need or demand to drive development; wasting resources to accelerate and select computer structures instead of letting natural evolution of the species occur; destruction of the high performance, i.e. supercomputing industry by government sponsored multicomputer design, followed by mandated purchases; and finally, negligible progress on a high performance network and its applications - the initial reason for the HPCC.

Worlton (1992a) describes the potential risk of massive parallelism in terms of the "bandwagon effect" where a community makes its biggest mistakes. He defines "bandwagon" as "a propaganda device by which the purported acceptance of an idea, product or the like by a large number of people is claimed in order to win further public acceptance." He describes a massively parallel bandwagon drawn by: vendors, computer science researchers, and bureaucrats who gain power by increased funding. Innovators and early adopters are the rider-drivers. He also believes the bandwagon's four flat tires are caused by the lack of: systems software, skilled programmers,

1Massive is defined in 1992 as either between 100-1000 or ≥1000 processing elements, processors, or computers depending on whether the computer is a SIMD, smP, or smC. Alternatively, massive can be defined as: either at or ahead of the state-of-the-art; or a computer at the limit of size that is considered to be economically viable.

2defined by Worlton as peak announced performance

guideposts (heuristics about design and use), and parallelizable applications.

We've seen tremendous improvement in the ability to exploit a massive number of computers working together on a problem, beginning with Sandia's applications of a 1K node NCUBE computer (Dongarra, et al 1991) that obtained a factor of 600 speedup, and is estimated to have a speedup of 1000 provided the nodes had more memory. The Sandia results showed that problems were scalable. During 1992, a multicomputer such as the CM5 or a collection of traditional supercomputers should be able to operate at a rate of over 100 Gflops for a problem.

These two laws of massive parallelism, governs progress:

> Some problem can be scaled to sufficient size such that an arbitrary network of computers can run at their collected peak speed, given enough programming time and effort; but this problem may be unrelated to any other problem or workload.

> The first law of massive parallelism is the foundation for massive marketing that supports massive budgets that supports the search for massive parallelism.

References

Bell, G., The Future of High Performance Computers in Science and Engineering, Communications of the ACM, Vol 32, No. 9, September 1989, 1091-1101.

Bell, G., Three Decades of Multiprocessors, Richard Rashid, editor CMU Computer Science: 25th Anniversary Commemorative, ACM Press, Addison-Wesley Publishing, Reading, Mass. 1991, 3-27.

Dongarra, J. J., Karp, Miura, K., and Simon, H.D., Gordon Bell Prize Lectures, In Proceedings Supercomputing 91 (1991), 328-337.

Hennessy, J. L., and Patterson, D. A. Computer Architecture: A Quantitative Approach, Morgan Kaufman, San Mateo, Ca., 1990.

Worlton, J., The MPP Bandwagon, Supercomputing Review, to be published.

Worlton, J., A Critique of "Massively" Parallel Computing, Worlton & Associates Technical Report, No. 41, Salt Lake City UT, May 1992.