72

Visions

Why there won't be apps: The problem with MPPs

Gordon Bell

Gordon Bell
gbell@mojave.stanford.edu

In spite of many years of research, massively funded, massively parallel (AKA "scalable") computers aren't yet successful. Nor are they likely to be unless they undergo a massive transformation to leverage developments in the mainstream computer and communications industries. The latest threat comes from standard workstations and fast, low-latency networks based on ATM. Like MPPs, these networks offer *size scalability* (from fewer to more processors), but they also offer *generation scalability* (from previous to future generations) and *space scalability* (from multiple nodes in a box, to computers in multiple rooms, buildings, or geographic regions). Furthermore, these networks offer a critical capability that MPPs sorely lack: application compatibility with workstations and multiprocessor servers.

The meager existence to date of special-purpose MPPs stems from four factors:

1. Their distributed structure is best suited to new, highly tuned, coarse-grained and/or large data-parallel problems.
2. Their market is too small to support vigorous application development.
3. They are incompatible with traditional, cost-effective computers that do have applications (if only limited scalability), such as networked workstations, workstation farms connected through high-speed switches, and shared-memory multiprocessors (including supercomputers).
4. Every generation differs, requiring recompiling, retuning, rewriting, and so on.

In other words, MPPs lack the top three success factors in any computer market: apps, apps, and apps. This fact should cause a reexamination and redirection of federal high-performance-computing programs, including those aimed at building "10-terafloppers." (Moore's Law specifies that researchers will attain a supercomputer with 10 teraflops of "peak advertised performance" in 2001 at the earliest. Funding an "ultracomputer" costing $60–240 million can reduce this time by 1.5–4.5 years.)

However, subsidizing application development will not necessarily make federally funded MPPs much more useful, competitive, or self-sustaining. An ideal scalable computer must be more than size, space, and generation scalable. To survive, it must also be *competitive* and *compatible* with other computer structures. But MPPs are competitive with other structures only in the rarefied domain of large, data-parallel applications (costing more than $2 million). They aren't competitive with computers that scale from 20-64 processors, nor with workstations that scale to thousands of processors.

Why not? Because MPPs lack a key property of great computer designs: *elegance*, whereby one component carries out multiple functions to handle the overhead inherent in all scalable structures (packaging, power, switching, operating system copies, and so on). For example, single-bus multiprocessors *are* elegant because the bus/backplane carries processor-memory-I/O communication, packaging, cooling, and power; the shared memory provides memory and infinite communication among processes. Networked workstations *are* elegant because the network carries out many communication functions, including support for parallel processing. MPPs *aren't* elegant because they leverage nothing. (They don't even leverage the companies that sell them: It is rare that a single organization can support products whose prices differ by more than an order of magnitude.)

Scalable computers can exist in a small niche as long as they are fully compatible with other computer structures, that is, as long as they build on the operating systems, tools, libraries, applications, and other components of existing computers. MPPs don't do this. Furthermore, with the emergence of high-speed ubiquitous networking, scalable computers must build on standard network structures that enable space scalability. MPPs don't do this either. But by leveraging space scalability we could build a massively parallel computer across any environment at "zero" cost by using existing workstations, servers, and standard networking. Hence, space scalability is a requirement for viability because it is the key to attracting applications.

So, to be competitive and compatible, scalable computers must adopt one of two standard programming environments: networked workstations, or shared-memory multiprocessors. Software (such as PVM and IBM's Loadleveller) is emerging that allows such a collection of nodes to operate as a single facility, including for the execution of parallel programs. Over time, the hardware to support such an environment will also evolve, as it must if parallelism is to enjoy widespread success.

A piece of advice

A European group that owns a size scalable computer asked me what they should do in massive parallelism, given that they had a few million dollars to spend and wanted to have a state-of-the-art computing environment. It turned out that they had virtually no applications or any source for them, that it was impossible to port applications from their supercomputer, and that they got little scientific output when the computer ran. (Many US users of MPPs say the same thing — as long as they aren't quoted. On the other hand, the National Center for Supercomputing Applications operates a CM-5 that is as accessible as a supercomputer and delivers significant flops for one-of-a-kind, highly data-parallel applications using Fortran.)

Here's what I recommended to them — *and what I recommend to managers of US high-performance-computing programs* — so they could write their own applications, run applications from others, get scientific and technical output, establish lasting industries, and get back on the main road toward parallelism:

1. Forget today's MPP, with its collection of special nodes interconnected by a special network and controlled by special software to support unique program language dialects. Instead, spend a large fraction of your budget on high-performance workstations (and applications), and interconnect them using an ATM switch that has the bandwidth and latency of a typical 100-node MPP. In such an architecture of "standard" nodes and networks, parallelism comes directly from the structure, is "free," and is a property of every collection of computers. This leverages industrial developments in both nodes and networks.
2. For finer grained parallelism, use compatible, shared-memory multiprocessor servers with 20-64 processors. These outperform comparably priced MPPs, they are general-purpose and more cost-effective, and they have applications.
3. Upgrade existing supercomputers so that vectorized applications will run effectively — whether supplied by a user or an independent software vendor.
4. Give the MPP to a sister institution that has an identical machine, thereby concentrating the support and pain in one larger, network-accessible site. This conflicts with a recommendation I made in 1993 to fund small (<128 nodes) MPPs at universities for training. However, any rational university will buy either ATM-linked workstations or shared-memory multiprocessors rather than a special-purpose MPP. Since the teraflop has become our symbol of pseudoindustrial strength, agencies such as the US Department of Energy might also concentrate their MPPs in a single site such as Los Alamos, Livermore, or Oak Ridge for "peak advertised performance" and support.

There are five requirements for the long-term survival of scalable computers. MPPs fail in three. They simply haven't been successful measured against any reasonable set of goals.

These recommendations follow directly from the price ranges, application granularity, and work applicability of the various computer structures. The applicability of MPPs is limited to data-parallel applications that are too big for large multiprocessors and too fine-grained for current networked workstations. They cannot process a general workload, scalars, or vectorized applications, and they lack standard applications. A ubiquitous, fast, ATM network supporting workstations or multiprocessor servers completely overtakes traditional MPPs in performance, price/performance, and applications at every price level.

The bottom line

There are five requirements for the long-term survival of scalable computers: size scalability, generation scalability, space scalability, compatibility, and competitiveness. MPPs fail in the last three. They simply haven't been successful measured against any reasonable set of goals.

If vendors such as Intel, Meiko, nCube, and Thinking Machines used standard nodes, networks, and programming environments, they could reduce development and product costs (by building from a single learning curve), improve their time to market, and concentrate their considerable skills (in networks, compilers, and applications) on paying back our federal and commercial investments in high-performance computing. Likewise, companies selling traditional workstations with compatible multiprocessor servers — such as DEC, HP, Convex, IBM, SGI, and Sun — could offer standard, high-speed networked environments at zero incremental cost.

Such developments would finally stimulate parallelism. Perhaps it's time to reevaluate our R&D agenda and funding policies.