

The New Limits on High-Performance Computing

John L. Gustafson, PhD

ClearSpeed Technology Inc.

Seymour Cray once quipped that he never made any money until he became a plumber.

We laugh because it's a case of very low-tech trumping very high-tech. This kind of irony is plentiful lately, as PhDs steeped in computational science find themselves wrestling, not with whether to interchange the level of loop nesting, or whether conjugate gradient solvers are superior to direct solvers, but with whether their power company is charging 23 cents per kilowatt-hour or merely 12 cents. Something profound has happened to the art of architecting supercomputing systems, and it has a lot more to do with issues from the era of GE's Thomas Edison than from the era of JPL's Thomas Sterling.

The Limit of Power Dissipation

Suppose you dream of building the world's first system capable of achieving a petaflop on LINPACK (one quadrillion floating-point operations per second solving dense systems of linear equations). The excitement builds as you sketch the possibility on the restaurant placemat... a few thousand nodes, each at a sustained speed of a few teraflops... and then you note what that kind of cluster will require in the way of power dissipation and real estate (square footage), and the "gulp" sound can be heard all over the restaurant.

Walk into any Wal-Mart and you will have no problem finding a Windows-based PC (that you can reconfigure to run Linux in a matter of hours) for a few hundred dollars. If you put it in a cluster, how much will you spend on electricity over its useful life of about three years?

Over *a thousand dollars*. Hmm... can that be right? If a server consumes just 300 watts and you keep it on for a year, how much does the electricity cost? At 12 cents per kilowatt-hour, it will use \$375 per year. Google's biggest single line item expense is the power bill for their enormous server farms.

Electricity ranges from about 5 cents per kilowatt-hour in places like DOE's Pacific Northwest National Laboratory (which is right next to a surplus of nuclear power as well as plenty of hydroelectric power) to about 23 cents per kilowatt-hour at the Maui High-Performance Computing Center (MPHCC). As the price of hardware falls with Moore's law, the price of the energy to flip all those ever-denser bits keeps rising with inflation and the price of a barrel of oil.

The chip and system suppliers have definitely gotten the message. Intel and AMD have moved away from advertising their gigahertz rates and made quite a point about

performance-per-watt. Having spent years convincing the typical consumer that speed and clock frequency are the same thing, their marketing departments now face the task of preaching an entirely new figure of merit. It might not be a tough sell to anyone who's run out of laptop battery power on a long plane trip, or has noticed how laptop computers have almost gotten too hot to keep on your lap for very long.

The POWER architecture folks at IBM may be amused by all this, since for the last several years the POWER-based entries at or near the top of the TOP500 list have very high performance-per-watt... and that didn't happen by luck. The POWER-based systems have long had people scratching their head when they win benchmarks despite having much lower clock rates than their competition.

But speaking of POWER-based designs, the Xbox 360 has set some records for the wattage consumed by a game console... about 170 watts. That is more than double the wattage of the original Xbox, and far more than the early video game consoles. As with supercomputing, the push for performance has resulted in a collision with the practical limits of power supply.

Unfortunately, in multi-gigahertz chips, the clock is itself a big part of the problem. Pumping the clock signal up and down takes from 25% to 50% of the power sent to the chip. And most of those frenetic cycles are simply waiting for some much slower part of the system, like the dynamic RAM or the disk or the network. It is a lot like roaring a car engine at the stop light. Chip designers used to think the power consumption simply went up linearly with the clock speed, but in practice, it rises much faster than linearly.

For the physicists among us, there is an interesting thing about making performance-per-watt a figure of merit. Performance is *computational* work per second. A watt is a unit of *physical* work (one joule) per second. (If you do not have a gut feel for what a "joule" is, 3.6 million of them is one kilowatt-hour.) Thus, performance in flops/sec divided by watts is simply flops per joule! As we approach physical limits to computation, there may be a convergence in the metrics used by computing people and those used by physicists.

For the largest supercomputing clusters, the procurement planners now must entangle a power budget with the hardware budget. It is a bit simplistic to equate the power budget with money. It usually does not work that way. The facility has some maximum ability to supply power to the computer room, and increasing it by several megawatts is a major engineering project. The money for doing that comes from a different pot, if the money is even obtainable for that purpose.

At an early hypercube conference, Intel discovered there was not enough electric power in the conference area of the Knoxville Hilton to power their iPSC system, so they rigged a rather audible diesel generator in the courtyard to get their demonstrations running. At a presentation later that day, the Intel speaker got a catcall from the audience: "Hey, how many megaflops per gallon are you getting?" The laughter lasted several minutes, but in 2006, it is no joke for those seeking the leading edge of HPC.

The Limit of Floor Space

When I first proposed to LANL that they fill a building with racks of microprocessor-based units to reach new supercomputing performance, they just about laughed me out of the room. That was in 1984, and supercomputing floor space was whatever it took for a love seat-shaped mainframe from Cray Research and some boxes full of proprietary disk drives, not an entire building.

Now it is generally accepted that if you want to do supercomputing, you will be filling a building with racks. The issue is how many you can accommodate.

Even if your facility has enough floor space, the communication fabric might limit the maximum distance between racks. And even if the electrical specification does not limit the distance, the speed of light will start to add to the message-passing latency between cabinets if they get too far apart. With MPI latency at about one microsecond lately, that has so far managed to mask plenty of speed-of-light delay; light in a vacuum travels about a thousand feet in a microsecond. But for a system that occupies over 10,000 square feet, connecting the opposite corners of a 100 by 100 foot square might (at best) be done with an optical cable in which signals transmit at only 70% the speed of light. Moreover, the cable is not line-of-sight straight between the corners, so allow perhaps 180 feet of cable between points. That delay adds about 250 nanoseconds to the MPI latency, reducing performance significantly for latency-sensitive applications.

The first thing to do, of course, is to pack each cabinet with as much computing capability as possible to reduce the number of racks. It may be easy to use a 25-inch rack instead of a 19-inch rack, but increasing the height introduces an objection from an unexpected source: those concerned with safety. Taller racks have a greater chance of injuring people if they tip over, and any system that asks administrative staff to stand on stools or ladders incurs a nasty reaction from the insurance company.

Packing more general-purpose computers into a rack, however, intensifies the heat generated. That forces the ugly choice between using more floor space and introducing liquid cooling. Air cooling hits its limit at about 70 watts per liter... near sea level. For a place like Los Alamos National Laboratory, at 7500 feet above sea level, forced-air cooling is only about half as effective. Hence, air-cooled racks hit a limit of about 20 to 30 kilowatts. Besides blowing the heat away from the processors, the requirement to blow the heat out of the computing room is getting arduous as well. The Thunder system at Lawrence Livermore National Laboratory requires an air current under the raised floor at 60 miles per hour. A typical guideline is that the power to remove the heat adds 20% to the total power required for the system.

Finally, the limit of floor space is like the limit of power dissipation, in that it does not simply translate into cost. The floor space may not be available, at any price. At a national laboratory, creating new building space is literally a line item for Congress to approve. In the financial centers like Manhattan and London that use HPC for financial modeling, the space is not only expensive but unlikely to be for sale right where it is needed.

So, What Should We Do?

Things are not as dismal as they sound. Perhaps the most visible ray of hope is in the emergence of multicore processor chips running at lower clock speeds. Every time a microprocessor vendor doubles the number of processing elements but lowers the clock speed, it drops the power consumption yet raises the effective performance. The ClearSpeed chip is currently the most radical example of this, with 96 processing elements running at only 250 megahertz. The result is a chip that is simultaneously the fastest at 64-bit floating point speed (about 25 Gflops/s sustained) yet one of the lowest for power consumption (under 10 watts). The ClearSpeed chip is a true coprocessor, and depends on a general-purpose processor for a host. If you need double-precision floating-point performance, you can achieve it without adding to the facilities burdens of power demand or floor space.

The new frontier in chip design is finding clever ways to compute with less energy. A startup company named MultiGig Inc. has a good example of the coming innovations to reduce power demand... a terahertz clock generator that slashes the power required to run the clock on digital chips, through a differential transmission line twisted into a Mobius loop. They and other chip designers are looking at adiabatic switching technologies that promise dramatic reductions in power per gate. DARPA's HPCS program has helped Sun Microsystems explore a method for inter-chip communication that promises to reduce energy per bit moved by over two orders of magnitude. This is exciting stuff, and it will directly benefit everyone from the consumer of video games to the scientists and engineers pursuing the fastest computing possible.

Using chips optimized for an application regime in combination with standard, generic processors is something we can do right now to mitigate facilities costs. This "hybrid" computing approach is very reminiscent of hybrid cars... two kinds of motors for two kinds of driving, resulting in much higher efficiency than if one tries to use one kind of engine for everything. The resurgence of interest in the coprocessor approach may remind people of the old days of attached processors from Floating Point Systems, but the FPS approach never had anything to do with saving electricity or space in the computing room. Coprocessors are no longer separate cabinets, but instead are available via plug-in boards that consume an otherwise empty expansion slot on a server or workstation.

The Tokyo Institute of Technology has used this approach to create the fastest supercomputer in Asia with a power budget of less than a megawatt. Los Alamos intends to create a hybrid petaflop computer with its Roadrunner project.

In 2006 high performance computing facilities are much more pervasive than they were 30 years ago. Performance is only one of the many attributes to be considered, with overall environmental impact being one of the most significant. Consequently making energy consumption a central design criterion to avoid generating excessive heat rather than solving for how to dissipate it after the fact is one of today's hot topics.

So don't do the plumbing, do the math! ■