

Is Your Computer Thirsty?

Rob Farber



Liquid-cooled hardware is silent cooling at far greater density than would be

possible with air-cooling

Ahhh! There is nothing like a tall, cool drink of water when thirsty. Not surprisingly, computers also prefer liquid refreshment as opposed to air cooling when hot. The choice for the technologist resides in when to make the move to liquid cooling and in what type of liquid cooling system is most appropriate.

Anyone who has been cautioned by their parents not to use an electrical appliance when taking a bath — especially one that plugs into the wall — has a visceral reaction against mixing electronics and water. Usually the combination results in disaster, otherwise referred to as letting the magic smoke out of the electrical device so it no longer works. (Better that than you, should your bath water come into contact with house current!)

The motivation in moving to liquid-cooled hardware is silent cooling at far greater density than would be possible with air-cooling. In some cases, the extra cost of the water-cooling system can be offset by a lower operating cost. Cooling and moving large amounts of air is expensive! For example, the 1.2 petaflop/s Peregrine supercomputer at the National Renewable Energy Laboratory (NREL) built in conjunction with Intel and HP has a near-perfect annualized average power usage effectiveness (PUE) of 1.06. An ideal PUE is 1.0, where the total energy utilized by the computer exactly equals the power delivered to the computer center. The Peregrine warm-water cooling architecture is part of the reason for the extreme annualized average efficiency, because the waste heat from the supercomputer also is used to heat the building. In contrast, Jeff Vetter notes in his book that air cooling the TSUBAME 1.0 supercomputer had a PUE of 1.44, as the air cooling system alone required 44 percent of the energy consumed by the supercomputer. When a system consumes megawatts of power, a lower PUE translates to big money savings!

While water is used in many liquid cooling systems, other systems utilize electrically insulating liquids such as fluorinert, which eliminates concerns about the liquid coming into contact with the electronics. It really is counterintuitive to submerge your cellphone or a piece of expensive — and likely critical — computer equipment into a vat of liquid and then apply power. Both water and fluorinert are examples of a single-phase liquid cooling system, where the temperature is kept below the boiling point of the liquid.

Many elementary school students perform a physics experiment where they put a thermometer into a pot of water and then apply heat. They observe that the temperature rises until the liquid starts boiling, at which time the temperature of the liquid stabilizes. This is a crude example of phase-change cooling where a liquid can carry away much more heat through evaporation than by simply absorbing heat. Try pouring some rubbing alcohol on your hand and blowing on it to observe the efficiency of this phase-change effect.

Phase-change cooling systems are utilized to provide very compact and efficient cooling systems for building very dense and compact computational systems. Essentially, phase-change systems work by pumping a liquid to a heat sink mounted on the CPU and other heat-generating components such as GPUs, coprocessors, bus and network components. The heat from these devices causes the liquid to evaporate. Tubes and a pump are used to move the resulting vapor to a remote condenser, where the waste heat is removed or used to heat water for the building, while also reverting the vapor to a liquid. The idea is very similar to how the air conditioner works in your car.

The cost of both liquid and phase-change cooling systems is rapidly dropping. This is a good thing, as one of my favorite Seymour Cray expressions highlights the cost of the engineering put into these cooling systems, “When I first started out, I was an electrical engineer and could barely keep my family fed. I then became a mechanical engineer and made a comfortable living. Once I became a plumber — as is typical of those in that trade — I made a very good living.”

Installation and maintenance costs must be considered when purchasing a liquid cooling system. Self-contained liquid cooling components are now entering the market at prices competitive to enthusiast air-cooling products. The advantage for the PC enthusiast is that they can run with the latest generation high-power CPU and GPU devices without having to listen to significant noise. Further, these devices are sealed and do not require maintenance. Enterprise and HPC customers will require more engineered liquid cooling systems that tie into their organization’s water system.

Two additional advantages of liquid cooling are

- cool chips last longer and exhibit fewer errors
- more efficient cooling means that power-intensive workloads can potentially run at the full hardware performance by avoiding thermal throttling.

Is Your Computer Thirsty?

Published on Scientific Computing (<http://www.scientificcomputing.com>)

As the HPC community strives to build an exascale machine, the power and cooling discussion continues to be a gating factor.

Bill Dally, NVIDIA Chief Scientist and VP of Research, recently spoke at the 2014 HPCAC Stanford HPC & Exascale conference about the path to exascale computation. In that talk, he reported that a Westmere 32 nm CPU requires about 1690 pJ/flop (picojoule or trillionth of a joule of energy per floating-point operation) when performing floating-point operations on the very efficient AVX vector unit. This is a best-case scenario, as non-vector floating-point operations consume even more power. (Of course, Bill highlighted the efficiency of an NVIDIA Kepler 28 nm GPU that requires 140 pJ/flop or roughly 11.4-times less power than the Westmere processor.)

Overall, data movement is a very expensive and power-inefficient operation on both CPUs and GPU devices. This implies that languages that frequently move data on behalf of the programmer (such as Java, Perl, Python and poorly written C++ classes) might be promoting low PUE in the enterprise space. Sadly, commonly utilized Web-based protocols, such as Google Protobufs, can also be considered as power inefficient because they require multiple data movements to create usable structures. In the HPC space, embarrassingly-parallel single-threaded applications might be the least power-efficient applications to run across an exascale machine, unless they can exploit vector operations. Look to data locality, vectorization and minimizing data movement to help your software achieve a better PUE. Happily, each of these options also will help to increase application performance.

For HPC, the choice of exascale architecture will be dictated as much or more by power and cooling than by legacy code and algorithm compatibility. This will, hopefully, open the doors to new thinking and algorithms as the HPC community abandons a quest that has already spent billions seeking modern hardware solutions to run decades-old scientific codes. In a world of tightening resources and profit motive, enterprise users may find that they are charged according to power consumed along with conventional charges such as CPU time and disk storage.

For both HPC and enterprise data centers, space is expensive. Thus, time spent with a spreadsheet and bids from liquid cooling vendors can demonstrate the cost-effectiveness of liquid cooling. There is also the “thinking green” appeal. Meanwhile, computer enthusiasts might find the cost differential — sometimes less than \$50 USD — worth paying to get a quiet, very high-performance and definitely overclockable system.

A fun side effect of the push to greater flop/s-per-square-foot and higher PUEs is that many supercomputer centers are now adorned by architecturally appealing cooling pools that occasionally provide evaporative cooling motivated fountain displays. It's a mark of programming success to cause the outside fountains to be turned on when your job runs!

Rob Farber is an independent HPC expert to startups, Fortune 100 companies, and government and academic organizations. He may be reached at editor@ScientificComputing.

Source URL (retrieved on 05/28/2016 - 7:47am): <http://www.scientificcomputing.com/articles/2014/04/your-computer-thirsty>