The Personal Linux Cluster: Putting GFLOPS on Desktops

By Douglas Eadline
deadline@linux-mag.com

September 19, 2007

Back in the day, before the IBM PC or the Apple II or the TRS 80 for that matter, there were those who said, "There is no need for a personal computer". "What could someone possibly do with a desktop that could not be done better on a mainframe?" seemed to be the thinking. Indeed, even after spreadsheets and world processors put computers in virtually every business, there were those that saw the home computer as a glorified video game, recipe index, and typewriter. We all know how well those kind of predictions turn out. To find out more just Google “bad predictions” on *your home computer*.

Enough early history, lets move on the the more recent variety. In the fall of 2004, as editor of *ClusterWorld* magazine, I thought it would be interesting to try and build a cluster for $2500. When I suggested the idea to some people, they laughed and basically thought I was crazy. After all, conventional wisdom told you that clusters were built from servers, not cheap PC hardware. (Actually many of the first clusters were built from PC/workstation hardware as the PC/Server distinction was just developing.) Not one to walk away from a challenge, I enlisted the help of Jeff Layton and we proceeded to build Kronos the Value Cluster (Eight AMD Sempron nodes with GigE). Kronos delivered 14.90 GFLOPS (Giga-Floating Point Operations per Second) and silenced many critics. We even set a record of sorts in terms of dollars/GFLOPS. (You can read all about it on Linux-mag.com http://www.linux-mag.com/id/2314/).

The Kronos cluster was valuable for several reasons. First it performed quite well on real applications like those used in protein folding and computational fluid dynamics. Second, it provided a great platform for software development and testing. The system was personal and under my total control. I could test new packages, change kernels, and basically do anything I wanted because I was not disturbing anyone as I broke things. In addition, because it was located in my basement, I was also paying for the electricity. This situation made me think about using power efficiently and I developed a set a scripts to power-off and on unused nodes using Wake-on-Lan.

Recently, I had the pleasure of learning about Microwulf (http://www.calvin.edu/~adams/research/microwulf/), another cluster built for $2500. The design was a bit more custom than Kronos, but Calvin College professor Joel Adams and student Tim Brom managed to get 26.25 GFLOPS out of four dual-core AMD Athlon X2 processors. Their goal was to have a low cost cluster for educational purposes. The advantage of dual-core was clear, more performance in less space. Earlier this year, I decided that building dual core successor to Kronos was in order. This time, I used Intel Core-Duo processors instead of those from AMD. After building the hardware and installing the software, I was able to quickly achieve 45.55 GFLOPS for $2500. For some reason, I decided to name this cluster Norbert, Son of Kronos. *Table One* summarizes the High Performance Linpack (HPL) results for all three clusters. HPL is used to rank the worlds fastest

Name/Processor	Clock Speed (MHz)	Release Date	HPL Performance (GFLOPS)
Kronos/Sempron 2500+	1750	7/2004	14.90 (Atlas)
Microwulf/Athlon64 X2 3800+	2000	8/2005	26.25 (Goto)
Norbert/Core Duo E6550	2333	7/2007	45.55 (Goto)

Table One: Yearly performance increase of $2500 clusters. Note: Atlas and Goto designate the BLAS library used in the benchmark

Figure One attempts to do some prediction based on this trend. As the figure shows, a line drawn through these three points and extended to 2010 predicts 80 GFLOPS for $2500. Some perspective may be in order. Ten years ago in 1997, 45.55 GLOPS would have landed at number 76 in the top 500 list, just below a Fujitsu 32 processor vector supercomputer.

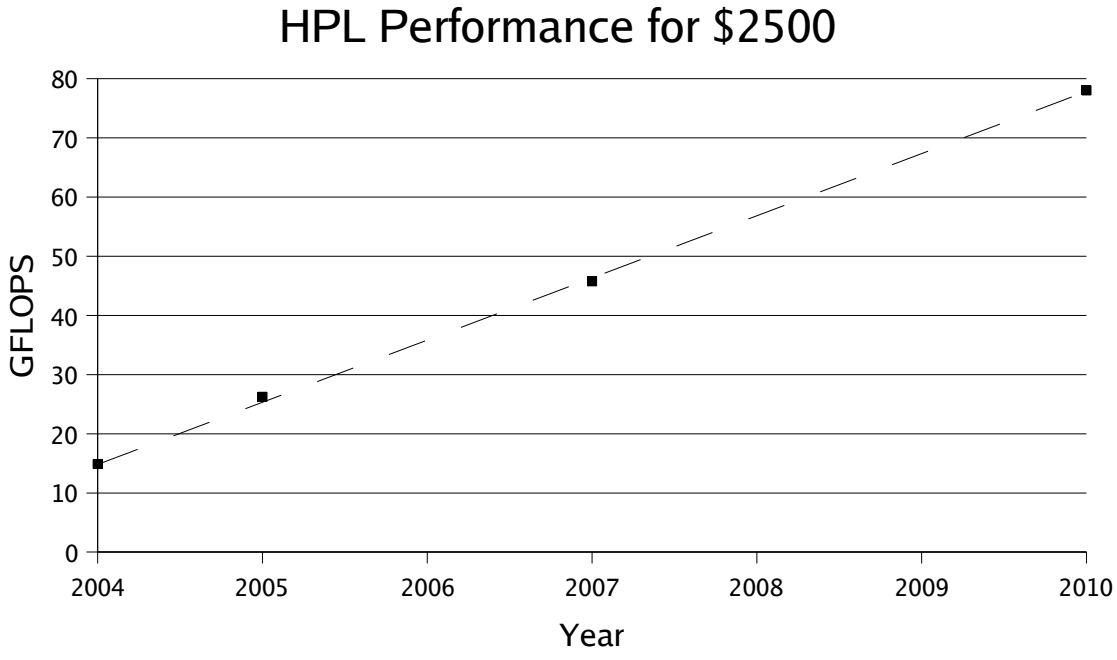


Figure One: HPL Performance Trend for $2500 Cluster

The Personal Cluster

The concept of the personal cluster is not new. There have been attempts to create such systems, but in the end, the cost did not seem to fall into what most users considered a personal price range. The expectation seems to be that the cost should fall in line with other desktop hardware. In this way, the cost of a four node personal cluster should be about the same as four desktop systems. Previous attempts seemed to miss this mark and thus were priced out of the personal range.

In addition, there was always the issue of power, cooling, space, and noise. A personal system should not require a dedicated power outlet and should be able to use ambient air for cooling. In terms of space and noise it should fit on or beside a desktop and not have minimal fan noise.

Finally, there is the issue of software. A personal cluster should be a “plug and play” device. The user should not have to deal with configuration and integration issues.

Looking at the issues, it seems that we are not that far from a true personal cluster and a possible business opportunity for a forward thinking company. While thinking about this situation, I decided that waiting for someone to recognize the opportunity might take too long and what was really needed was a community driven personal cluster project. A project of this type could resolve some of the final issues and possibly convince vendors that packaging hardware in the form of a small cluster may be an advantageous business opportunity.

We Need A Project

The great thing about an open project is that it draws on the wisdom from a community. Based on the recent press interest in the Microwulf cluster, the time seems right for personal cluster project and a community appears ready. The goals I have set for this project are both reachable and pragmatic. There are three major milestones that for such a project are three fold. First, to fill in the gaps as noted above and create a true “plug and play” device. Second, to provide an open Linux based software platform that can be used to develop applications and learn about clusters. And finally, help build an economic case for so vendors can package the hardware in a convenient desktop fashion.

We'll take a closer look at each these below, but first two important issues must be resolved. What to call the project and where to host the it. As the self appointed project leader, I have chosen the name LIMULUS(tm) which stands for LInux MULti-core Unified Supercomputer. See the *Sidebar* for more background on the name. The hosting part is simple, as I already have secured space and set up a project page at http://limulus.basement-supercomputing.com.

Defining Limulus

Defining a cluster is hard because the design can vary widely. Because we have decided to limit our choices and costs, we can narrow down the definition of a Limulus personal cluster and even create a specification of sorts. Therefore, a Limulus personal cluster

1. Uses standard hardware off the shelf hardware and allows for both file system and compute node expansion;
2. Uses an Open Source cluster software stack that contains an integrated set of software packages to manage the cluster and create and run software. The software stack must run on top of existing Linux distributions;
3. Consumes less than 100 watts when idle and less then 1000 watts when fully loaded;
4. Has a power supply efficiency above 80% to reduce heat
5. Has a noise signature that is acceptable in a modern office environment (less than 45dB).
6. Achieves a price to performance of less then $100(US) per GFLOPS

As you can see some of the design parameters are rather loose allowing design flexibility, while others are very tight providing desktop/office functionality. As we take a look at these in more detail, the design goals will become more clear.

Hardware Design

The hardware specification has plenty of wiggle room to allow for future hardware innovations. The other requirements will almost force the hardware toward low cost desktop devices so using a server as Limulus probably

will not work. For instance Norbert uses single CPU socket micro-ATX motherboards to help reduce cost. The key requirement is that it is expandable.

As a an example a reference design is given in *Figure Two*. This design was used for the latest Norbert cluster. Notice that there are two separate Ethernet switches. Ostensively, one is for compute traffic and one for administration and file system traffic, however, there is no restriction on how applications may use these two channels. Also note that one port on the switch is used for expansion within the private cluster network. A complete hardware manifest for the design in *Figure Two* is provide on the project site (total cost is $2500).

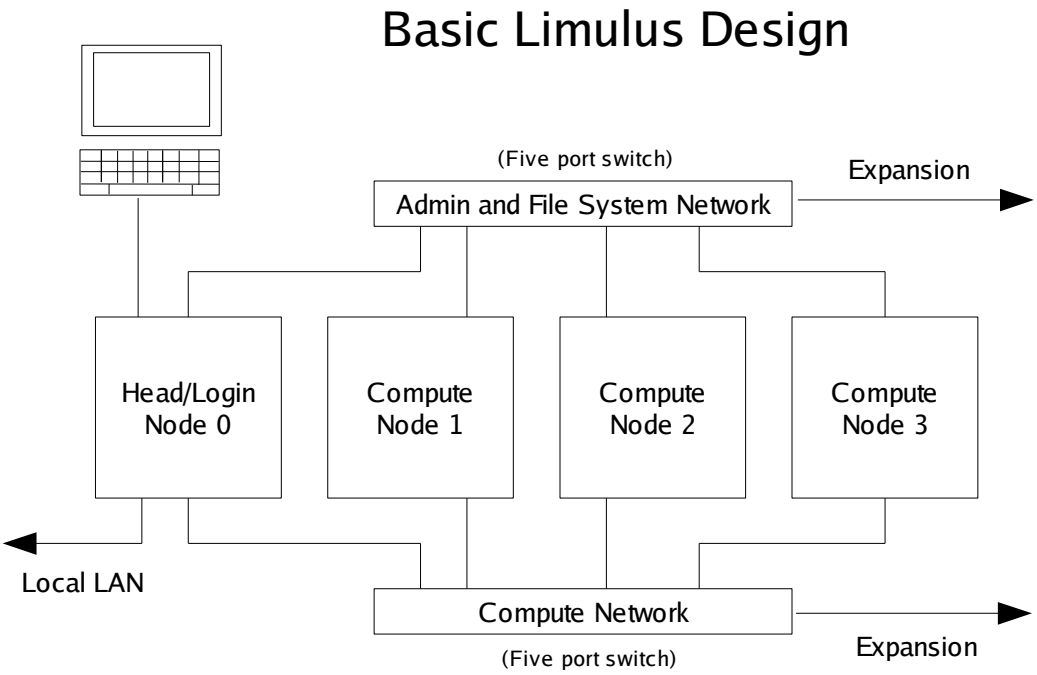


Figure Two: The basic Limulus design used for the Norbert Cluster

Software Requirements

Due to space, the software requirement is a bit vague in terms of actual packages. There are two important issues, however. All software must be integrated (e.g. if you provide an MPI version it must also work with the scheduler). The second requirement is that all software work "on top of" an existing Linux distribution. Modifying and supporting a Linux distribution is not the goal of the project. There is also list of the minimum required software on the project site as well. The goal will be to allow users to start using the cluster right away with out excessive configuration issues.

Power Consumption

This requirement, while restrictive, is quite achievable. Let's take a look at some numbers. The four motherboard Norbert cluster consumes 250 Watts of power when idle as measured by placing a Kill-a-Watt meter on the power cord. This number, while low is not going to be acceptable to the personal user, where the system may be powered on most or all of the day and the extra processing power may only be used for a small portion of the this time. Part of the project will work to reduce this to less than 100 Watts (possibly less than 50 Watts) when idle. These numbers are achievable by employing the new Core Duo C-states and the new tickless idle feature in the kernel. (See the

GearHeads column in the September Issue.)

When it is fully loaded running the HPL benchmark Norbert draws between 371-483 Watts. The number varied as the benchmark progressed which I assumed is an indication of how modern processors are managing power consumption based on load. Even with the maximum power draw, the power performance ratio is a comfortable 11 Watts per GFLOPS.

Heat Issues

When one thinks of HPC they often envision racks of servers sucking in cold air and pushing out lots of hot air. In an office or desktop environment this is not going to be acceptable. While the CPU is certainly a source of heat, another large source is the power supply. A typical power supply is about 65% efficient (this number depends on the load as well). In terms of heat, that means that 35% of the electricity going into the power supply comes out as heat. New power supplies are aiming for an 80% or better efficiently and thus a reduction in heat. Therefore, in order to minimize the heat generated by the cluster, highly effective power supplies will be required. One aspect of the project will be to find ways to increase the efficiency of the power supply and possibly use only one power supply for the whole cluster.

Noise Issues

In addition to heat, anyone who spends time around servers knows that noise can be an issue as well. To move the cool air through a small server chassis, banks of small fans are normally used. A modern 1U server may employ 12-15 small fans or a lesser amount of blowers to move air through the case. Small fast fans equals big noise. Fortunately in a desktop office environment, larger fans can be employed and the noise level made tolerable. Such is the goal for Limulus. Currently the Norbert cluster used cases with a 3.5 inch fan and a single CPU cooler. The noise level is quite low and certainly tolerable in an office environment. Part of the project will also address more dense packaging schemes that are just as quiet.

Price-to-Performance

One defining feature of a Limulus will be the sub $100 per GFLOPS performance ratio. This number clearly puts Limulus in the low cost realm and ensures that as hardware performance increases that the system cost remains reasonable. It should be noted that Norbert has achieved $55 per GFLOPS.

Filling in the Gaps

As mentioned the project is designed to provide the engineering and integration that is needed to meet our self imposed specification. In terms of the recent Norbert cluster, managing power and power supply efficiency are the only two issues that need completion. These should be complete by the end of the year and the goal will be to have the project page with a hardware manifest for variations on the basic design as well. (i.e. with quad-core processors, a Limulus can have 16 cores!) There will also be a software ISO file and How-To guides. Once the hardware is assembled, the system should operational within an hour.

As part of the software stack, special emphasis will be given to software development tools. In particular, we plan to use Limulus as basis for developing tools that push the user above message passing and closer to application programing. We also envision turn-key applications that can be easily downloaded, installed, and run on a Limulus.

Packaging

The one remaining issue, not part of the specification however, is packaging. This aspect was not placed in the specification because it allows users to get creative in terms of how they package the hardware. The simple and easiest method for now is to buy inexpensive cases, place them on wire shelves and start computing.

While using commodity cases is a workable solution, a case where I can place four micro-ATX motherboards,

power supply, wires, etc and place it on my desk (or beside it) is high on my list. I have started to create some prototypes and I believe it is not only feasible, but it may be possible to license the design to a vendor (See below).

Funding Limulus

While the fruits of open projects are great for the community, there is the often quoted saying "Well, how are we going to fund this, as we do have to eat". As I conceived the project I also had the goal of eating.

Because the project has the ability to sell hardware, there certainly is an incentive for vendors to support such an effort. Of course outright funding is always welcome, but I also wanted to include provision for the project to get paid on market success. Having watched various open business models grow in the past, the one thing I noticed worked well was branding. Did you ever notice that in the early days every time there was an article about Linux Red Hat was mentioned. I don't think this was an accident. Using this as a model, I have outlined a funding plan for the project as follows.

The Limulus name and logo will be trademarked. There may be some other packaging Intellectual property (IP) that the project develops as well. There will also be a software stack which will be include the Limulus trademark.

Vendors who wish to sell genuine Limulus systems will be required to license the name and any IP they use from the project. In this way, if a vendor wants to use the recognizable Limulus name to help sell hardware, they will be required to license it. For example, a Limulus Software CD will be available to licensed vendors. Unlicensed vendors can certainly copy designs and software, however, they cannot use the Limulus trademark (or possible IP).

Individual and academic users are free to use all Limulus technology and software as they wish. If an end user is not building Limulus clusters for re-sale, then there is no need for a license.

The goal would be to funnel all license money back into the project to help develop software and hardware solutions for the Limulus Cluster Project. It should be noted that after the basic project goals are complete the emphasis on programming methods and tools will be increased. The ultimate goal will be to abstract away the parallel nature of the Limulus (both multi-core and cluster) from the programmer and provide a truly unified system view.

You Are Invited

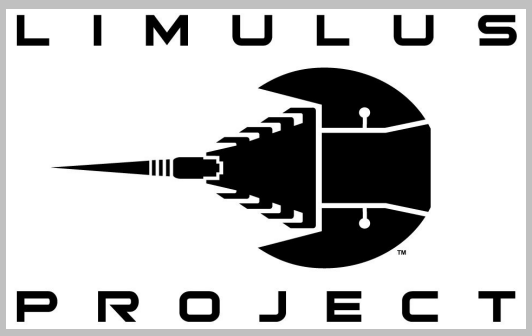
There is much more to do and talk about, but space and time are real limiting factors. The project page will have more information as it develops. Of course, I realize, I may have just announced a party and I might be my only guest. Well Jeff Layton will show up because he wants a cluster that fits under an airplane seat, so there is two people already! If you have interest, drop me a line or check out the project page.

Finally, I also expect to get many puzzled looks as I continue to describe the project and all the goodness it brings. I certainly expect to hear many people say "What can someone possibly do with personal cluster?" That certainly is a good question, and I don't expect to have an answer. I'll just have them chat with my intelligent agent-assistant. That is as soon as it is finished prioritizing my email, optimizing my schedule, and telling me how it found an Internet story about someone who just archived 50 GLOPS using cell phones and the Bluetooth.

What is a LIMULUS?

The name Limulus polyphemus or what is commonly called the horseshoe crab is a fascinating animal. The species has been around for over 350 million years and is often referred to as a living fossil. When you think about how long the Limulus has manage to preserved its biological information, it really does boggle the mind. We humans, still have a way to go as we only have about 250,000 years to our credit. Its blood is based on copper, which by the way has a primitive antibody system that is used to test contamination in modern vaccines. It can even regrow lost limbs. You can read much more at http://www.horseshoecrab.org/ as well.

The name LIMULUS was chosen for the project because in addition to coming up with a reasonable cool acronym, longevity and usability of information are the goals. There are many ways to build a cluster, and the Limulus Project is designed to provide a long lasting series of best practices, recipes, methods, and software for personal supercomputing. Well, long lasting in terms of technology.



Limulus Project Logo