The Labs and Linux

Revision 1.3

Larry McVoy

BitMover, Inc.
San Francisco, California
415-821-5758
lm@bitmover.com

ABSTRACT

The computing world is evolving at a rapid pace. While most segments of the computer market are growing, the supercomputer area has not kept up with the expansion of other areas in the market. In fact, the supercomputer segment is in danger of being relegated to a small niche player.

The Labs have a vested interest in the health of the supercomputer market, being one of the primary users of supercomputer technology. The Labs have very specific needs which are not currently being well addressed by either the commercial vendors or the free software world.

In this document, we will contrast the Labs needs versus the needs of the more general computing market. The role of Linux with respect to supercomputing will be examined, with special attention being paid to Linux' clustering plans. We will consider several options open to the Labs. Finally, we will suggest a moderate approach to solving the computing problems facing the Labs. We believe that it is possible for the Labs to get what they want with a fairly small investment and some innovative approaches to the problem.

The approach we suggest involves Linux clusters, so the second half of this document describes what we mean by a cluster, why this view is important, and then finishes by explaining why this sort of cluster is important to the Labs. It may seem obvious that the Labs want clusters, but it is not immediately obvious why the Labs would want the sort of clusters described here.

1. Outline

- Lab needs
- Labs place in the computing market
- Linux clustering roadmap
- Options
- SMP clusters
- Why the Labs need SMP clusters

2. Lab needs

The Labs have some of the most advanced computing needs in the world: bigger, faster, and more scalable than just about any other institution. The ASCI SOW is a good place to go for the details, we just touch on some of the high points here.

- **Unix operating system semantics** across a cluster of 10,000 CPUs. This has never been done to date. Clusters almost this large have been done but none have anything remotely approaching a single system image.
- **Low latency communication**, i.e., 10 microsecond round trips for locking. The rest of the computing market, while viewing this as interesting, is unwilling to pay the associated high cost.
- **Large data sets**, which can be terabytes each, and need to span across nodes in a cluster.
- **Gang scheduling**, for efficient use of resources on clustered systems.
- **Check point/restart**. While useful to the Labs, not universally viewed as important.
- **Network attached storage**.

These are just a few of the computing needs of the Labs. The interesting question is: are these needs shared by a significant portion of the computing market?

3. Lab's place in the market

The supercomputing market is dramatically smaller, as a percentage of the computing market place, than it was 5 years ago. We estimate the supercomputing market to be around 1% of the total computing market. It used to be closer to 10%. We believe that the supercomputer market will continue to decrease in size as a percentage of the market. We expect the dollars spent annually to remain fairly constant.

If it is true that the supercomputer portion of the market is that small, the Labs can no longer demand Lab specific features from the computer vendors. It simply doesn't have a financial justification. It is important to realize that it is not that the features that the Labs want are bad, or uninteresting; it is that there are many other features which provide a far higher return on investment. As computing meets the mass market, the relevance of the high end technical features decreases - neither Jane Public nor Joe Business need gang scheduling but Jane and Joe represent 99% of the market. From the point of view of the shareholders and the board of directors of any computer company, there is no economic reason to cater to the Labs. In fact, it is unhealthy to do so - companies which do cater to the Labs will be at a distinct disadvantage when compared to their competitors who have been working on features which are useful to the 99% of the market rather than the 1%.

The Labs need to either adapt to the new market or fund the development (**and** maintainence) of the features they need. We'll talk more about this in the Options section below.

4. Linux clustering roadmap

Without any input from the Labs or other members of the HPC world, we believe that the Linux clustering roadmap will focus on the following issues:

- Highly available services such as redundant DNS/HTTP/SMTP servers.
- Beowulf style clusters, i.e., racks of machines connected with conventional networking and programmed with PVM and/or MPI.
- Database clusters, i.e., clusters similar to Beowulf but with a distributed database such as Oracle as the only application.
- A cluster file system will at least be attempted.
- Other clustering features will be attempted but will suffer because of no effort to coordinate work.

We do not think that the roadmap will include things which are important to the Labs, such as low latency interconnects, scaling to large sized clusters, or any of the other features which are are important but too esoteric to be considered part of the mainstream.

5. Options

Given this somewhat bleak picture, there are a number of options open to the Labs. The options range from do nothing and hope things get better to decide to take on 100% of the problem as a Lab project.

5.1. Do nothing option

One option is to hope that the various computer vendors will continue to consider the supercomputer a worthwhile expense for pure marketing reasons. Most of the work done in the supercomputer market can be somewhat justified based on the idea that if a company can solve problems of size X then problems of size X/100 should be trivial. This line of reasoning is getting somewhat outdated; SGI is a company that has taken this approach and it isn't exactly making their stock soar.

It is true that the government considers this market segment important and will continue to bring pressure on computer vendors to support the supercomputer development efforts. Whether that can continue in the face of the changing market demographics is an open question.

5.2. Do everything option

The other end of the spectrum is to assume that the vendors are going to completely abandon the high end. The Labs could decide to fund 100% of the development of supercomputers, perhaps based on clustering foundations. This approach would have both a software cost, for the OS and utilities development, and a hardware cost, for the interconnect and network attached storage development. We estimate the annual cost of each to be $4-$20M and $10-$100M, respectively.

5.3. Mainstream option

Another option is to carefully examine the mainstream development efforts and see if there is some way to satisfy the Lab's needs using commodity software and hardware. We expect that the Labs have done this already and have not yet found a suitable answer.

We are going to make a case that there is a mainstream option not yet explored because it doesn't exist yet. The thrust of our suggestion is that the Labs help encourage a particular model of computing because that model, while not an exact match, is very close to what the Labs needs. And the model is likely to be mainstream in due time.

6. SMP clusters

What's an SMP cluster? It's not what you are used to thinking of as a cluster. An SMP cluster is a cluster of operating system images cooperating on a single machine. Rather than running one OS image on a N processor system, we'll run N/S OS images, where S is some small number such as 4 or 8. S is the highest number reachable before going over the locking cliff.[1]

The cluster behaves, to a first order approximation, as if it were one large SMP. In other words, it has the following attributes:

- Global namespaces, such a process, file, and device names. This means that any node in the cluster can say

  $ kill -HUP 12345

  and process 12345 gets that signal, whether it is on the local node or not.
- Cluster wide process groups for existing semantics (controlling ttys). Supplementary process groups which can be used for things like gang scheduling.
- Virtual processes, or "process objects." Think of this as doing to the process concept what vnodes did to the file concept - it allows multiple instances of things which behave as processes. The obvious first two instances are local and remote processes. There are other instances as well: debugged processes (ptrace et al), different sorts of remote processes (we advocate a process model that does not support process migration; the MOSIX folks like process migration - the virtual process object allows us to have both models and see which one we like better).
- The system automatically does some load balancing at process creation time. The load balancing has the following characteristic: if all jobs took exactly the same amount of time, then all jobs could be started on one node in the cluster and the load would be perfectly balanced. This means simple parallel tasks, such as `make -j` (parallel builds), would scale up close to perfectly.
- There is a cluster file system, SMPFS, which allows operating systems to share a global page cache. The cache is coherent by definition, since the data is actually truly shared rather than copied. Think of it as an mmap() which works across OS boundaries, i.e., one OS can map another OS's pages.

There are differences between an SMP cluster OS and a traditional SMP OS. For example, on an SMP OS, /tmp is a globally shared scratch area, /proc is a global view of the processes, etc. On an SMP cluster OS, these areas are private, i.e., per node. There are global versions under /g/tmp, /g/proc, etc. So if you want to see all the processes in the system, the `ps` command would chroot to /g and then run as normal.

This is a very brief overview of what we have in mind for a clustered OS model for large SMP hardware. There are already plans underway to refine this model, the next update will be some time after the O'Reilly Open Source conference in August - a small group is gathering there to try and come up with a more detailed architecture.

[1] See "SMP scaling considered harmful" for a definition of the locking cliff.

6.1. *Why SMP clusters?*

A good question is: why should this sort of technology ever exist? We already have several operating systems which scale up to 64 processors and at least one example, IRIX, which scales up to 256 processors. Why bother with a different way to solve the same problem?

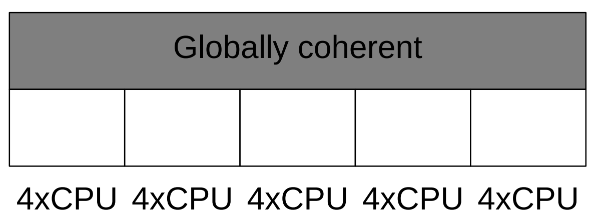
Consider the different demands placed on a 2 processor SMP OS and a 256 processor SMP OS. They both offer the same abstractions but the larger machine needs an operating system which can have 256 processors all doing something useful - in the kernel - at the same time. That means the 256 processor OS is fair more fine grained in its locking approach than the 2 processor case. The problems associated with locking are discussed in another paper, suffice it to say that asking one operating system to handle uniprocessor and 2-256 SMP operating system tasks is an unreasonable request.

There is another consideration, which is Linux specific. Linux today is not a fine grained threaded OS, it scales only to a few processors. That means that Linux is still a relatively simple OS by today's standards. That's a good thing. Scaling Linux to 100 or 200 processors would turn Linux into something that was no longer the lightweight, high performance OS we know on small machines. It would trade low end performance for high end performance, which is not a good trade off.

None the less, not scaling is not an option. There are problems which are larger than 4 processor systems. Linux has to be able to solve those problems if it is to be taken seriously in the enterprise market. The pressure to scale up is already here and is being felt on a daily basis by the Linux developers. There has to be a scaling answer.

SMP OS clusters are a less invasive approach to scaling and could be applied to Linux. While the virtual process changes are quite invasive, the rest of the changes are not - they tend to fit nicely under existing abstractions such as the file system layer. The problem is nicely partitionable as well - parallel projects could be started on the process model, the load balancing, the file system coherency, etc. This approach has the built in advantage that it is inherently more scalable; consider scaling up a run queue on a 1024 processor SMP system and then consider scaling it up on an SMP OS cluster - the partitioning needed for scaling is already done.

The final plug for this model is this: we have knowledge that at least two >$10B/year hardware vendors have designed and are building hardware with features specifically designed to support this model. In particular, the hardware has support for multiple coherency domains, so each OS can see two kinds of memory: locally coherent and globally coherent. You can than have a view of memory like this:



where each lower box is memory which is coherent for a single 4 processor node running one OS image, and the upper box is a large portion of memory which is globally coherent.

The reason for building such an architecture is that SMP doesn't really scale - it gets harder and harder to build systems with globally coherent memory. Once you realize that you are going to essentially partition all of the important data structures in the OS in order to scale up, having those partitioned data structures in localized memory is just the next logical step. Have each localized memory running its own OS is only a small leap forward.

6.2. *Why should the Labs care?*

At first glance, this seems like a don't care for the Labs. The Labs are facing problems which are larger than any one box can handle, so investing in an architecture which is single box doesn't help. Or does it?

To see if this approach is at all helpful, consider the following: Beowulf clusters, the SMP OS cluster, and a Lab cluster. Consider the differences between what a Beowulf offers and what the Labs want. The Beowulf approach is all commodity parts, with commodity software. The latencies between nodes are at least 10x to slow and are actually closer to 100x too slow. The Lab cluster looks like a many nodes connected with a high bandwidth, low latency interconnect. Finally, consider an SMP OS cluster on something like an SGI Origin. The Origin looks like a many nodes with a high bandwidth, low latency interconnect.

What's the point? If the Labs rely on the Beowulf approach, the numbers which matter are orders of magnitude off. On the other hand, because the SMP hardware has "interconnects" which are as fast as memory (because they are memory), an OS which is tuned to run on such an interconnect will be far more likely to run well on a real cluster. The code paths which need to be short already will be short. In the Beowulf case that won't (and can't) be true.

The summary is this: Linux is facing a scaling problem. Linux could follow in the footsteps of every other OS and trade off low end performance for high end performance. Or Linux could try something new. If this new idea could be made to work, it is something that has many of the necessary characteristics that a Labs cluster would need to perform well. It wouldn't have everything, the Labs would still need to fund some work, but it would be dramatically less work than any other choice facing the Labs today and for the foreseeable future.

7. Suggestions

I think the Labs should first digest this idea, then discuss it for a while trying to find flaws. If after a time, the idea seems to stand up, the Labs might want to host a conference for interested parties. I could be helpful in putting the right people together to attend such a conference. I believe that the Labs did such a conference a number of years ago with the result being a short document, listing the various features that the Labs considered important in a computing environment.

I make no claims that this paper is anything approaching a serious architecture specification. We need such a specification and a working cluster conference would be a good place to start. The Labs could essentially play the same role that IEEE plays with respect to POSIX.

If that approach worked, the result would be that we would soon have people from SGI, HP, SUN, IBM, Red Hat, and various Linux folks, all working on a joint clustering architecture for both SMP and true clusters. There is no need to limit this to any one company or any one OS, by the way. If the FreeBSD folks got wind of this and wanted to do a FreeBSD implementation, that would be great.

There are numerous people working on these issues already, the situation clearly calls for a more organized approach. The Labs could provide that structure.

The Labs also have resources, already paid for, which could be used to test out the ideas. Think about those big SGI machines at Los Alamos, for example. Development could be started on a small SMP machines and then continued by researchers at the Labs on large SMP machines.

8. Conclusion

I believe that it is in the best interest of both Linux and the Labs to seriously consider this combined approach to the problems of scaling on SMP and clusters of SMPs. The Labs could take a leadership role this area, with the results having long term benefits for both the Labs and the general computing market.