



Courtesy of O'Reilly & Associates

Ten Tips for Building Your First High-Performance Cluster

by *Joseph D. Sloan* 12/29/2004

While high-performance clusters often provide the most cost-effective way to speed up your calculations, building your first cluster can be a frustrating experience. But it doesn't have to be. Here are ten tips to help you get started.

One: Set Realistic Goals

It should be obvious: the first step in building a high-performance cluster is to decide what you are going to do with it. You'll need to select the appropriate hardware and determine what software your users will need. Clearly, these decisions will affect your design. For example, for data-intensive calculations, you'll need a large, fast I/O subsystem.

Cluster planning can be difficult if you haven't used clusters before. Understanding the capabilities and limitations of a cluster comes with experience, so you may be faced with a chicken-and-egg problem. The best way to avoid costly mistakes is to start small. Build a test system with a small number of machines just to see what you are likely to encounter. This will allow you to determine what you want and need before you make major commitments. In the long run, this will probably save time and money, since correcting mistakes in a large cluster can be very time-consuming and very costly.

Two: Use Identical Hardware

There are exceptions to this rule. You'll almost certainly want a faster

head node for your system. And you'll want larger disks on I/O servers. But in general, life will be much simpler if you can standardize your hardware.

Using the same hardware for each machine in the cluster will simplify installing and configuring your clusters, since you'll be able to use identical system images on each machine. It will simplify maintaining your cluster since, all of the systems have the same basic configuration. You'll need to stock fewer spare parts and will be able to swap systems in and out of your cluster as needed. But the really big savings will come when you program your cluster; you won't have to code for differences in performance among machines. With dissimilar hardware, this can be a never-ending frustration. I realize this recommendation runs counter to the traditional image of a Beowulf cluster composed of a pile of old, recycled computers. But you won't regret it. Hint: if you are looking to use similar hardware but can't afford new equipment, look around for IT departments doing bulk upgrades.

Three: Avoid the Siren Call of Diskless Systems

If you are putting together tens of thousands of machines in a cluster, trying to use disk-based systems is a major problem. You'll be replacing drives on a daily basis. But for small clusters, disk reliability isn't really an issue. And prices are continuing to drop as capacity and reliability increase. Disk-based systems are much easier for the beginner to set up and will provide better performance for many applications. (For this reason, my book doesn't describe setting up diskless systems.)

Four: Don't Skimp on the Network

All too often, the network is an afterthought when building a cluster. Computing is about computers, isn't it? In practice, the network is usually the performance bottleneck for non-proprietary clusters. Adapting your code to minimize the impact of a slow network is another source of unending frustration and is a tremendous waste of a programmer's time. With rapidly

falling Ethernet prices, there really isn't any need to skimp. Think in terms of switched Fast Ethernet as a minimum. And if funding will allow it, consider Gigabit Ethernet.

Five: Minimize But Don't Over-Minimize Your Hardware

When configuring the compute nodes in a cluster, there is a lot of hardware you can do without. You certainly won't need sound cards and speakers. And you can probably get by without mice, keyboards, and displays. It is easy to set up a crash-cart with display, keyboard, and mouse that can be wheeled to individual machines when problems arise. If you are purchasing equipment, minimizing the hardware can lower your total cost or allow you to buy more machines.

Less equipment also means fewer maintenance headaches--up to a point. But it is certainly possible to go too far. If you have to stop and install a display adapter before you can attach a monitor, you've probably crossed the line. And in many cases, installing software will be easier if computers contain CDROM drives. Neither a basic display adapter nor a CD-ROM will add very much to the price of a new computer, and there certainly isn't anything to be gained in removing them when using recycled computers. While you won't be using either very much, when they are needed, it is nice to have them in place. Since they aren't used that often, they shouldn't create a maintenance problem.

Six: Isolate Your Cluster

While it is certainly possible to put every node of a cluster on the Internet, I've never seen a good reason for doing so. But it doesn't take much thought to come up with some pretty good reasons not to--security being foremost on the list. If you don't have to harden the compute nodes on your system, then the installation will be simpler, performance will be better, and you won't have to be constantly installing security patches to a large number of machines.

If you need to provide network access to your cluster, then you can provide

it through a single hardened machine, typically the cluster's head node. You'll need to maintain and monitor that node closely. But don't provide more access than is needed. Putting the head node on an internal network that's behind a firewall is often considerably less risky than putting it directly on the Internet.

Seven: Use Cluster Kits, But Learn What They Do

Packages like OSCAR and Rocks greatly simplify setting up and maintaining clusters. Either of these packages will probably install and configure all of the software you'll need to get started, and both provide a mechanism for going beyond the basic software.

But while they provide working systems with a minimal of effort, at least initially, you will need to learn the ins and outs of the software packages they install. Don't kid yourself into believing that you can install one of these systems and just walk away from it. Once you have installed OSCAR or Rocks, you'll still have a lot of work ahead of you. The good news is that these kits will buy you the time you need to accomplish it.

Eight: Get Over the "Latest and Greatest" Mindset

What you want is a working cluster that meets your needs. It is very unlikely that you need the latest release of your favorite Linux distribution. Cluster kits are based on the more common Linux distributions, but it takes time to adapt to new releases. With most clusters, the users will only log directly onto the head node. So for the compute nodes (i.e., the bulk of the cluster), it doesn't matter which Linux distribution you are running, as long as it does the job. And you should restrict use on the head node to cluster-related tasks, so you won't need or want the latest and greatest email or news reader, etc. If you insist on using the latest release of a less-common Linux distribution, you'll be creating unnecessary work for yourself.

Nine: Plan for Expansion and Replacement from the Beginning

If I may be permitted to make a prediction, you are going to be pleased with the performance improvements your cluster provides, but you are going to want more. The useful lifecycle of a computer is incredibly short, and this applies to clusters as well. This is particularly true if you are using recycled equipment, since part of the computers' useful lives has already passed before you get them. It won't be long before you need to add or replace equipment.

There are several things that you can do that will make life easier. Make sure you have adequate space and power to expand. When buying network equipment, look for equipment that can be expanded or daisy-chained. Of course, you can recycle some equipment, such as monitors, etc. But the best thing you can do is collect the information you will need to design your next cluster.

Ten: Document Your Cluster as You Go

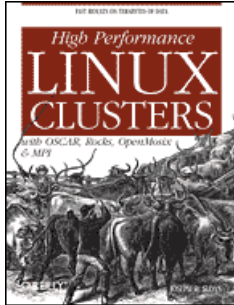
Good documentation is the key for both optimizing the use of your current cluster and for designing your next cluster. If you are installing using kits, record the configuration information. While you may not initially understand everything, figuring things out later will be easier if you have good records. When building your next cluster, being able to look back on how the last one was built and configured may help you avoid some unpleasant inconsistencies that your users may notice, even if you don't.

It is often said that rules are meant to be broken. It is true that you can ignore all of the above tips and still build a working cluster, and there are certainly exceptions to all of the above. But if you are new to clusters, these tips should make your life easier. Once you have built your first cluster, you'll know enough to judge which tips are useful and which tips don't apply in your special circumstances when building your next cluster. And, you'll be able to write your own book.

Joseph D. Sloan has been working with computers since the mid-1970s.

Copyright © 2004 O'Reilly Media, Inc.

Related Reading



***High Performance
Linux Clusters***
with OSCAR, Rocks,
OpenMosix, and MPI

By *Joseph D. Sloan*