SYBASE®

WHITE PAPER

ASE Cluster Edition on Linux: Installation And Best Practices Guide

Principal Author: Chris Brown, Sybase, Inc.

TABLE OF CONTENTS

INTRODUCTION

Since its release in December of 2007, ASE Cluster Edition has been one of the most readily accepted Sybase products released by in recent memory. By combining greater high availability, with ease-of-use, the product creates a win-win situation for both the DBA and the application end user. It allows dynamic workload management along with intelligent routing of connections among the individual instances of the cluster, all without significant application changes i.e., all of the decisions are made at the database server level and not at the application layer. This enables quicker and easier deployments of applications to a shared-disk architecture, without any major application changes or re-coding.

One of the major design goals of ASE Cluster Edition was ease of use, and by logical extension, ease of installation and management. Therefore, installing ASE Cluster Edition on the Linux platform or any supported platform does not have to take days of trial and error. In fact, it's not uncommon for customers to have their first cluster built, installed and possibly loaded on a single day, if not a single afternoon.

This document is designed to aid both the novice and experienced ASE Database Administrator install and configure ASE Cluster Edition on the Linux platform. Drawing on the results of many customer installations and Proof of Concepts, the recommendations provided here are tried and true in the "real world". This paper should also help answer some of the most commonly asked questions about hardware and ASE-specific configurations and guidelines.

One thing to remember. This guide does not discuss the issues that arise when architecting applications for shared-disk based databases. It only discusses the installation and configuration of the hardware and ASE software itself required to successfully plan, install and build an ASE Cluster on the Linux platform.

This document is most relevant to ASE Cluster Edition version 15.0.3 and higher.

HARDWARE RECOMMENDATIONS

It is not the intention of this paper to recommend one vendor over another. We are only providing guidelines as to what should be included in your hardware purchase and what should be considered in your hardware buying decision.

Physical Hardware Recommendations

I recommend the following minimal hardware configuration for each node. Note that each node can contain a different number of processors, cores and RAM. This difference in configuration can be accounted for within the cluster configuration file (and subsequently within the Workload Manager) once ASE Cluster Edition is installed and your cluster is built. However, keep in mind that you are not allowed to mix different operating systems and/or hardware architectures, with the one exception listed below.

Note also that these are minimum hardware recommendations and, in those cases, "more is better".

- 64-bit CPU(s). This means that the CPUs themselves can either be AMD64, Intel EMT64 or a mixture of the two because they run the same Linux binaries. In other words, one of the hosts may have AMD64 processors in it while another one may have Intel-based EMT64 processors in it, but they still run the same Linux distribution—in this case, x86_64.
  - At the current time, there is no 32-bit platform support, IBM POWER (running Linux) support, Itanium support (running Linux), ARM processor support or SPARC support (running Linux) for ASE Cluster Edition.
    - NOTE: check http://certification.sybase.com for the latest platform support as this list may change in the future.
- One of the following Linux distributions (again, 64-bit only):
  - Red Hat Enterprise Linux 4 update 4 or higher
  - Red Hat Enterprise Linux 5 update 1 or higher
  - SuSE Linux Enterprise Server 9 update 3 or higher
  - SuSE Linux Enterprise Server 10 update 1 or higher
    - NOTE: check http://certification.sybase.com for the latest supported operating system certification levels, as this list is subject to change as new Linux versions or updates are released.

- At least four physical processors with one core each or multi-core processors totaling four cores in all.
  - This is one of the cases where "more may be better". Due to the way that ASE cluster edition works, it's better to have one core (or processor, in uniprocessor-based machines) perform all cluster-interconnect based transfers per node and leave the others to service client requests, etc.
  - As you might expect, the "faster the better" when it comes to CPU's. This allows for faster in-memory hashing, computation and a longer in-service life for the application running on top of ASE Cluster Edition.
- At least 4 GB RAM per processor / core, or 16 GB RAM minimum
  - Although the total amount of RAM that each node uses is up to you, I suggest that at minimum you use 1GB per core as a rule of thumb. This will allow sufficient memory for caching data, distributed lock management, etc. without ASE having to swap anything to disk.
- Redundant power supplies, etc.
  - History would tell you to build as much redundancy as you can into the physical machine—but at the same time understand that this might not be necessary. With redundant power supplies, etc. the physical machine won't crash in the event of hardware failure, but the general architecture of ASE Cluster Edition would "fail over" any connections and/or applications that were using the failed node to a surviving one. Therefore it might be better to spend more on faster CPU's, more RAM or faster network cards than additional redundant components.

As you can see, the physical hardware requirements for ASE Cluster Edition are quite basic. Each machine could have a different number of CPU's, RAM and cores, as in the following (sample) architectures:

1. In a 3-node cluster, two machines might be identical and service the same workload, with a 3rd idle machine, larger or smaller in size, designed to handle the load of the failed machine(s)—understanding that when failed over, applications might run in a degraded state for a time until the main machines are brought back online.
2. In a 3-node cluster, one machine might be large, with lots of CPUs and RAM for OLTP-based use while another may have fewer CPU's and less RAM (but faster I/O cards) for DSS-based activity. The 3rd machine in the configuration may only be used in the event of OLTP failover, but not for DSS.

These are only examples. The flexibility of ASE Cluster edition allows for many different types of hardware configurations as determined by budgets and the service-level agreements of the underlying applications.

Storage Recommendations

For underlying storage, the choices are more limited on Linux than on other UNIX-based operating systems. Officially, the product only supports raw partitions, with no filesystem-based device support yet. Therefore you cannot use a GFS or CFS-based filesystem, shared amongst each of the nodes, for database devices. Customers have used NAS-based devices that present themselves to the operating system as raw partitions, and this has worked well. But on each one of the nodes the raw partitions must be located (mounted) at the same location because each one of the ASE instances shares the same master device (and as such, the same sysdevices table), so each node is going to look for database devices in the same location.

Normally there is no need for a cluster manager to manage devices and filesystems within ASE Cluster Edition because you don't need to "fail" devices, filesystems, and virtual IP addresses back and forth between nodes of the cluster should one of them fail. However, there can be exceptions to this, depending on the type of installation, as outlined below.

Remember that ASE Cluster Edition is a shared-disk architecture, which means that all nodes see (and share) the same underlying storage and databases, with the exception of the local tempdb. This is why you only need single devices shared amongst each of the nodes.

For storage, the following guidelines apply:

- You must use RAW devices (as of this writing, filesystem-based devices are NOT supported). You will need the following devices created by your SAN administrator:
  - One for the master device. It should be at least 30 MB in size. This device will hold the master database as well as model and the global tempdb (dbid=2). Please read on for more information about tempdb requirements because tempdb works differently in ASE Cluster Edition than in the standard SMP edition.
  - One for the sybsystemprocsdevice. This will hold the sybsystemprocs database, and it should be at least 152 MB or greater, depending on how many user-created global stored procedures you will install afterwards.
  - One for the sybsystemdb device (this is required with ASE Cluster Edition) of at least 6 MB in size.
  - One for the quorum device (note: no actual database visible from within ASE is created here). It doesn't need to be large—10 to 20MB will be more than enough—but it will contain basic configuration and state information about each of the instances/nodes participating in the cluster, as well as the master copy of the configuration file if you decide to use a "private" $SYBASE (see below). The quorum device should also be on a separate LUN and included as part of your backup scheme because the backupserver doesn't do a dump of the quorum (it's a device, not a database visible from within ASE Cluster Edition).
  - One device EACH for the local tempdb on each node. Therefore, if you are going to build a 3-node cluster, then you will need three individual raw partitions for tempdb—one used by each nodes. For performance reasons this is preferable to building one large tempdb device that is "chopped up" and shared amongst each of the nodes.
  - An appropriate number of devices (data and log, etc.) for your user databases.
    - Remember that each instance of the cluster sees the same thing as another, so you only need to create the devices once.

      For example, if you have a four node cluster and are connected to instance ASE1, and you run a disk-init command for "datadev1" on ASE1, then ASE2, ASE3 and ASE4 are going to "see" datadev1 once the disk init command completes, and be able to write to it the same as ASE1.
- Make sure that your SAN supports SCSI-3 PGR. This is a SAN level setting that you will have to verify with your storage administrator. ASE Cluster Edition uses SCSI-3 PGR for I/O fencing. This is important in case one of the instances of ASE becomes isolated and doesn't "know" it is isolated. Having this property enabled at the SAN level will prevent the rogue node from causing corruption by continuing to write data to the underlying devices. When creating your database devices, remember that this is set at the device level.
- You have the option of creating the raw partitions either though the use of the /etc/rawdevices file or by using a udev-based rule (for RHEL5 and SLES10 only). When given the choice, it is better to write a simple udev-based rule for the raw partitions. But make sure that, in the rule, permissions are set correctly on the newly created device file(s). If they aren't, the raw devices will always be owned by root, and you'll have to change them manually on each node when the node is booted (or rebooted).
- Optionally, you could create shared filesystems on each of the nodes, mounted in the same location, for each of the following situations:
  - If you dump your transaction logs and/or databases to disk, then you should have a shared filesystem for this use mounted at the same location on each node. This can be an NFS-mounted filesystem, or preferably a cluster-based filesystem.

    Why should this be shared? Although the backupserver can run in multiple availability modes, if a transaction log (or database backup) is dumped to ASE1 located on node1, node2 won't be able to access it unless it's located in a shared location.
  - If you have administrative scripts that you normally run via CRON on each node, then it would be wise to have a shared filesystem, mounted at the same location on each node, to run each of these administrative scripts from. This will prevent having multiple copies of the same script on each node and allow you to manage everything from a single location on the cluster.

To install the default $SYBASE, you have two options. It can be shared (via an NFS-based or preferably CFS-based filesystem) by each of the nodes in the cluster, or it can be installed locally. ASE Cluster Edition 15.0.1 only had the option of installing $SYBASE on a shared filesystem. For redundancy purposes, Sybase recommended that customers use a cluster-based filesystem because NFS-based servers raise a host of availability and performance concerns at client sites.

- For ASE Cluster Edition 15.0.3 and higher, you have the option of installing $SYBASE in "legacy" mode (meaning, the same way as in 15.0.1) or in "private" mode (where each node has a copy of the binaries installed locally, not shared in any way with any other node).
- For "legacy" mode installations, make sure that you use a cluster-based filesystem, not an NFS-based filesystem, for $SYBASE. Also, $SYBASE must be mounted in the same location on each of the nodes.

For "private" installation (preferred), it's best to install $SYBASE in the same location on each node to prevent any confusion and make management of the cluster easier. Installing $SYBASE locally also lays the foundation for possible "rolling upgrades" in a future release of ASE cluster edition and allows you to easily shut down a node, perform O/S maintenance and then reboot that node without affecting any of the other nodes in the cluster.

Following these guidelines will make storage management within the cluster quite simple.

Network Recommendations

For new Cluster Edition customers, networking seems to be the most problematic area during cluster installation. Nothing is more frustrating then building a cluster and not being able to connect to it, or worse, not having each of the instances "see" each other on the individual nodes.

The following guidelines will help you avoid network problems.

- Do *NOT* use DHCP for any network interface within cluster edition. All IP addresses for the public and private interconnect addresses should be static.
- Use Gig-E or faster for the interconnect(s) between the hosts. Remember that as the number of nodes within the cluster or the number of users increases, the amount of data that is transferred between the nodes will increase as well. Using a fast interconnect will prevent performance problems and/or cluster instability.
- Make sure that the "public" IP addresses for the hosts you are going to use are in your corporate DNS. You should be able to access/ping the hosts by hostname (both full and short) from hosts outside the cluster in this way, not only by IP address.
- Make sure that the hostname isn't aliased to the "localhost" entry in /etc/hosts. This is common in initial installs with RHEL and SLES. In RHEL it will be aliased to 127.0.0.1 and in SLES it will be aliased to 127.0.0.2 if the machine is built before a static IP address is assigned to it. Also, issues have been encountered with multiple engines not being able to come online if the hostname is aliased to "localhost".
- Do *NOT* use virtual IP addresses, or "floating" IP addresses, within the cluster. Failover and connection management is handled at the OpenClient (ct-lib only) /JDBC/ODBC level, and there isn't a need for this going forward.
- Create at least three network interfaces, consisting of the following:
  - At least one "public" network interface for clients to connect to. In the Linux world, this is most often known as "eth0" (the first Ethernet interface) and is the first physical network card on the machine. If you want to use additional public network interfaces, you can also add them to the machine at this time and have an ASE listener on each one of these interfaces, as in the SMP version.
  - Two "private" network interfaces for the nodes to pass data pages back and forth. While you can technically get away with just using one private network interface, for performance and redundancy we recommend using two. These private network interfaces should be accessible and ping-able from every node in the cluster.
    - At this time, ASE cluster edition only supports two private network interfaces.
    - Make sure that the private network is truly private! Data pages are passed around using UDP (which isn't encrypted) for performance reasons, so you will want to make sure that no other hosts are using this network and that it cannot be compromised in any way.

  - You'll also need to add these "private" network addresses to each host's /etc/hosts file, and the IP's should NOT be ping-able from any other hosts in the network, nor should they be DNS-resolvable.
  - Make *sure* that the private network is being handled by a switch and not a router. All of the IP addresses used by the private network must be on the same subnet.

Note that, due to the way ASE transfers pages between each node in the cluster, the faster the interconnect between the hosts, the better the performance of the cluster (especially as additional nodes are added or the workload is increased on each node of the cluster). Not only are locks passed back and forth between the instances of the cluster, but ASE Cluster Edition will pass data pages back and forth between the nodes (vs. reading them off of disk) if the page requested is located in the cache of one of the instances. This means that performance of the internconnect network is critical to the overall performance of the cluster itself.

Also, remember that the cluster interconnect is used for instance-to-instance pings to make sure that nodes are "alive". If one of these pings isn't received in time (or a certain number of them aren't successively "answered") then the other members of the cluster will vote the non-responding node out of the cluster, and it will shut down and failover connections that are attached to it—even though it may not truly have gone down. If pages are delayed while being transferred between nodes, or node-to-node pings are not be received, cluster instability will result.

CONFIGURING ASE CLUSTER EDITION

Configuring the Operating System

After installing the operating system as indicated above and making sure that everything works correctly (i.e., that the networks are configured correctly, the DNS name for the host resolves correctly, the devices for ASE Cluster Edition to be installed on are seen on the host(s) in the same location(s) and the interconnects are working properly) then only a few basic changes will need to be made to the operating system:

- Change the shared memory setting in /etc/sysctl.conf. On both RHEL and SLES, this file is located in the same place and controls the amount of shared memory that can be allocated at one time on the host. Depending on the version of Linux that you are running, this value can range from 32MB to the total amount of RAM on the host. You can check the current value by the following command:
  - cat/proc/sys/kernel/shmmax
- To set shared memory to the proper value, multiply the value that you want by 1024, twice. So, if you want to be able to allocate 20GB of RAM to the ASE instance running on your current host, then the value you would want to set shared memory to is (20480x1024x1024) = 21474836480. This can be set in one of two ways:
  - Echo "21474836480" > /proc/sys/kernel/shmmax (but be aware that this value will be reset to the default when the host is rebooted)
  - Insert kernel.shmmax = 21474836480 at the end of /etc/sysctl.conf, save the file and then run "sysctl -p" as root. The value will take effect immediately and will be reset to the value above when the system is booted.
  - Changing the I/O scheduler has been shown to improve I/O performance under heavy I/O load. Therefore if you expect that your cluster will be fairly I/O intensive, add "elevator = <deadline|noop|cfq>" to the boot prompt line and reboot your system. Which one you choose depends on your hardware; I recommend that you test the effect of this. It only works on later releases of RHEL and SLES, but if you see improvement from it then you can add it to your bootloader configuration file.
  - Create a 'sybase' user, belonging to a 'sybase' group, on each host. Make SURE that each of the user ID's and group ID's match on each host.
  - Adjust the following kernel parameters:
    - aio-max-nr – the default values of this parameter have been known to be low sometimes, limiting the maximum number of asynchronous I/Os that can be performed. You can adjust this by "echo <value> > /proc/sys/fs/aio-max-nr"
    - You can make this permanent by setting "fs.aio-max-nr=<value>" in the /etc/sysctl.conf file.
  - exec-shield – set this to '0' to disable it
    - Echo "0" > /proc/sys/kernel/exec_shield
  - Set "kernel.exec-shield=0" in the /etc/sysctl.conf file.

- randomize-va-space – set this to '0' to disable it.
  - Echo "0" > /proc/sys/kernel/randomize_va_space
  - Set "kernel.randomize-va-space=0" in /etc/sysctl.conf

Installing ASE Cluster Edition and Building Your First Cluster

Once the operating system has been configured and the system rebooted, it is time to install the ASE Cluster Edition binaries. This can be done in one of two different ways, depending on the actions taken in the first section of this paper.

Installing with a shared $SYBASE

If you are using a shared $SYBASE installation directory (that is, $SYBASE is mounted at the same location on each host, and the mount point is exported from the SAN or an NFS server) then the steps are as follows:

- Login to one of the hosts in the cluster.
- Un-tar the installation binaries, and run the setup program on this host
  - Note that once you do this, ALL of the hosts in the cluster will be able to see the same binaries.
- Once the installation completes, make sure that you install the appropriate licenses in either the $SYBASE/SYSAM-2_0/licenses directory (if you are using un-served licenses) or you have properly configured a license server and have SySAM pointing to it.
- Login to EACH host in the cluster, as the sybase user, and run "$SYBASE/UAF-2_5/bin/uafstartup.sh".
  - This needs to be run on each host, so that the UAF agent and daemons will be running on each node that you will be building the cluster on.

Installing with a local $SYBASE on each node

If you are planning to install the sybase binaries locally on each node (recommended), then you'll need to perform the following steps on each of the nodes:

- Login to the host
- Un-tar the installation binaries, and run the setup program
- Once the installation completes, make sure that you install the appropriate licenses in either the $SYBASE/SYSAM-2_0/licenses directory (if you are planning to use un-served licenses) or you have properly configured a license server and have SySAM pointing to it.
- Run "$SYBASE/UAF-2_5/bin/uafstartup.sh" on the node.

Again you will need to repeat the above steps on each of the nodes in the cluster. This is the preferred way to install ASE cluster edition.

When the binaries are installed and the UAF agents are running, you can run the 'sybcluster' application to build your cluster. You only need to run this program on ONE of the nodes because you can manage the cluster with this application from any node in the cluster.

When executing this application, you will need to pass the hostnames of the UAF agents that you are planning to use in your cluster. For example, "sybcluster -F node1, node2, node3".

Some things to keep in mind when running this application:

- The application doesn't check to see if the device sizes that you specify are large enough to successfully run the disk init program. If you don't know the exact size of the devices, you can run "dd" against them to determine the correct size of them.
- If you make a mistake when entering in any values, make sure at the end to save the information in a file (you will be given the option) and don't create the cluster yet. Exit out of sybcluster, edit the file that was created by the sybcluster program and then re-run 'sybcluster' as above. When creating the cluster, you can pass the filename in so that you don't have to re-type everything back in again.
- Linux doesn't allow you to name the raw devices anything intelligible, so everything is going to be sequentially named like "/dev/raw/raw<some number>". It is advisable to note which device belongs to which device before running 'sybcluster' to avoid any confusion.
- I advise creating multiple backupsevers to allow for redundancy. The default mode for each backupserver will be "round robin"—good for basic redundancy and high availability. You can change the mode that the backupserver runs in later when the installation completes.

After your cluster is up and running, the following configuration changes from within ASE are recommended:

- Have all user databases use 2K logio. This is different than the recommendation for the SMP version; within cluster edition we have shown that the 2K logio size is optimal for a 2K page server (vs the normal doubling of the page size for the logio).
- Turn on the statement cache and enable literal autoparameterization.
- Adjust the CPIC-related configuration parameters (monitor them with MDA) because the default message sizes tend to be too small for larger, OLTP-type of applications.
- Adjust the normal configuration parameters. Keep in mind that they affect each instance of the cluster, so if you have nodes within your cluster that have different resources available to them (memory, CPU's, etc), you will need to adjust them to fit that specific node.

TIPS, TRICKS AND SUGGESTIONS

UAF

The UAF framework is required for ASE cluster edition (because there isn't a RUNSERVER file created by default anymore), as well as for sybcluster and other processes to use. Therefore, it's critical that it be started when the host machine is booted. Otherwise you won't be able to start ASE Cluster Edition on the node. In rc.local (boot.local on SLES-based hosts) add the following line:

```
su – sybase -c "/opt/sybase/UAF-2_0/bin/uafstartup.sh &"
```

Cluster Startup Takes A Long Time

Sometimes cluster startup can take time, normally due to one of the following issues:

- Large size of local tempdb.
- Large shared memory allocation at instance startup (that is, each node is configured for a large amount of shared memory).
- Cluster was shutdown suddenly or with nowait, and the quorum device still indicates that instances are up and running.

In the case of issue #1 and #2 above, 'sybcluster' will issue messages to the console that it's waiting for the instance to finish starting up. When it finishes waiting, it will move on to the next instance (if there is one still remaining to startup) and try to start that up. But that next instance will not startup, nor will any subsequent instances start, until the first one completes. The only way to work around this problem at the current time is to either reduce the size of shared memory (or create smaller tempdb devices at cluster creation time) or start each instance individually.

With issue #3, the first instance will wait a configurable number of times to make sure that no other cluster instance is running, and if not, will attempt a "takeover" of the cluster and startup.

Understanding and Using "qrmutil"

'qrmutil' is a new binary that ships with ASE cluster edition. Since the quorum device isn't directly viewable and changeable from within ASE itself (remember, it's not a database; it's just a device that stores configuration and metadata information about the cluster itself), this application (located in $SYBASE/ASE-15_0/bin) is used to query the quorum device and potentially make changes to it.

'qrmutil' has many parameters that can be passed to it. You can just execute the binary with no options passed to it to see what it can do. For example, If you need to set permanent trace flags for the cluster, you would use this application to set them. With this application, you can also extract the latest copy of the configuration file, restore a deleted (or corrupted) quorum device—assuming you have a backup of it—and take a backup of the quorum device as part of your backup strategy.

Sybase, Inc.
Worldwide Headquarters
One Sybase Drive
Dublin, CA 94568-7902
U.S.A
1 800 8 sybase

www.sybase.com

SYBASE®