

SUSE

LINUX ENTERPRISE SERVER – HA AND CLUSTERING SOLUTIONS



SUSE LINUX ENTERPRISE SERVER – HA AND CLUSTERING SOLUTIONS

OVERVIEW	3
APPLICATION TYPES		
	Stateless Clusters	3
	Web Farm	3
	Fail-Over Clusters.....	4
	File Server	4
	Database Server.....	4
	E-Mail Server	4
	Parallel Database Systems	4
	Blade Server and Grid Clusters	4
	Web Service Cluster	5
	Grid Computing	5
CONCLUSION	5

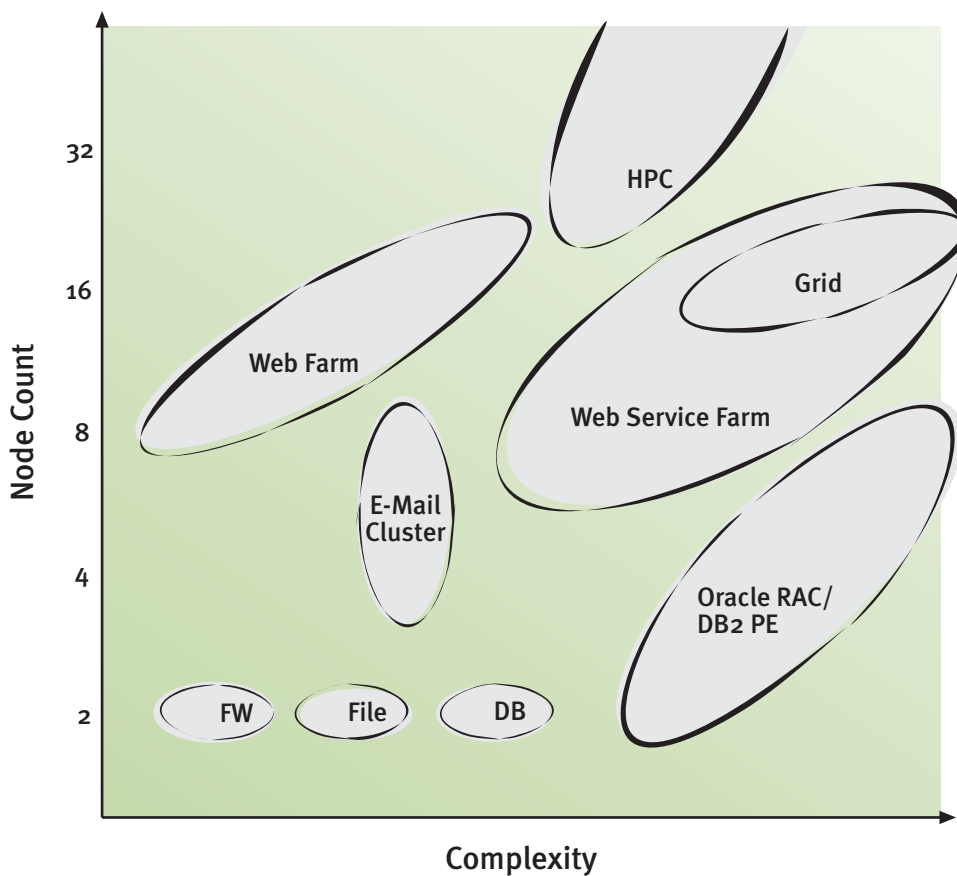


Figure 1:
Cloud chart with estimates of typical cluster sizes for applications of varying complexity

OVERVIEW

This document describes a number of cluster solutions for a selection of application types. The focus of this paper is mainly on clustering for high availability, with lower priority on scalability. In some configurations, scalability might be added by the clustering solution as a by-product, such as in load balancing situations, but the large domain of pure high-performance computing clusters will be regarded in a separate paper.

Figure 1 shows a distribution of some applications with typical cluster sizes. The complexity is roughly estimated as a function of the amount of application “state” that has to be transferred between the cluster nodes. The application “state” can for example be a database that is taken over by the other node on shared disks.

APPLICATION TYPES

Using the criteria of cluster size and “statefulness”, some typical groups of clustering types can be identified.

STATELESS CLUSTERS

An example of stateless or nearly stateless applications are web farms for static content or simple session-based applications.

Web Farm

An application that is almost stateless, like a web-based application used by large number of end-users with web browsers, can be clustered for availability of the service by a load balancing mechanism that regularly monitors the availability of the single node. In the case of a failure of a single node, some responsibility for the recovery of the application state may be placed on the user (who may have to log in again). This is accepted practice in all but the most critical applications.

The Linux Virtual Server (LVS) kernel module and utilities can be used as a layer 4 network switch together with mod backhand for more complex load-balancing applications. Typical multi-tier solutions with java servlets can be implemented with the Tomcat/Apache load balancing interface, for which extensions have already been evaluated. All these packages are included in the SuSE Linux Enterprise Server 8.

FAIL-OVER CLUSTERS

The classical fail-over cluster solution will include two or more cluster nodes connected to a shared set of storage devices. In most operating environments, the administration effort of managing a cluster of more than two nodes is avoided except where a load balancing solution offers added scalability, so the two-node cluster is frequently the best solution for the customer.

Replication of data and updates are also solutions and alternatives to expensive shared storage systems for some applications, typically characterized by a low update frequency to the data. DRBD¹ is a stable kernel-level implementation for storage replication over IP networks and is featured in SuSE Linux Enterprise Server 8. The downside of host-based replication is that network bandwidth quickly becomes a bottleneck which is unacceptable for many applications.

Another widely used application is the package heartbeat (included with SuSE Linux Enterprise Server 8). It has a few thousand installations up in production in the real world and also works well with the LVS and the DRBD project. Heartbeat implements serial, UDP and PPP/UDP heartbeats together with IP address takeover including a nice resource model and resource groups. It currently supports multiple IP addresses and a simple two-node primary/secondary model.

Another important feature for large environments is multi-pathing for the storage system access, which implements redundant and fault-tolerant routes between a host and the storage devices. SuSE Linux Enterprise Server 8 provides multi-pathing with help of the multiple devices layer (MD) in a general way. MD has support for failover and read/write balancing independent from the real devices.

¹ Distributed Replicated Block Device

File Server

The network file sharing protocols still contain some challenges in fail-over clustering as some parts of the session state may not be transferred easily to the surviving node.

Database Server

Almost all fail-over clustering frameworks offer modules for the most successful commercial and open source databases.

E-Mail Server

E-Mail server Clusters combine the requirement for scalability to serve thousands of mail users with the high availability requirement for the data storage.

Typically, the data storage is implemented as a load balancing system of fail-over clusters, with a reverse proxy system distributing the e-mail protocol requests to the correct server. A directory system manages the load and resource distribution.

Parallel Database Systems

Database systems that are designed as parallel systems are a special form of the fail-over cluster. A major characteristic is the concurrent access to the storage devices. These systems usually do not require an additional fail-over framework, but manage the availability of the nodes internally. They are usually clustered both for availability and scalability. The most prominent example is Oracle 9i "Real Application Cluster" (RAC). IBM's DB2 also has a parallel database which is designed as a "shared nothing" database. This system has to be augmented by a fail-over solution and has data partitioning requirements to scale well.

All above mentioned cases can be handled by SuSE Linux Enterprise Server 8. Also available are solutions from vendors like SteelEye or Polyserve.

BLADE SERVER AND GRID CLUSTERS

Grid computing is mostly described as using the idle processor capacity of user workstations for productive use. This currently does not seem to be a viable alternative to server systems for most commercial applications.

APPLICATION TYPES

For some application types, the main characteristics of grid computing can be salvaged. If the need for application-specific customization of the underlying operating platform can be reduced, a “cluster” of servers can be shared among several applications, and the application load can be shared by appropriate mechanisms.

Web Service Cluster

One example is the standardized environment that most Java-based web applications and web services require. The application server system (Servlet and EJB Container with standardized middleware components) simplifies deployment mostly to the point of distributing the program archive files and resources including the deployment descriptors.

The emerging web service architecture adds complexity to the already described web application cluster: the client for a web service usually will be another application, which needs a more sophisticated reaction by the server than can be expected from a human user in the case of a session failure.

Java security manager policy files can restrict the permitted activities of a deployed application on the target machines, so that multiple applications owned by different information owners can be safely deployed on the same machines.

Load balancing, resource allocation and node availability is controlled by a layer 4 switch like Linux Virtual server, or a reverse proxy layer that can be implemented with Apache servers and the corresponding proxy modules. Mod backhand can be used for complex measurement-based load balancing requirements.

As the computing resource, blade servers together with large database servers are a flexible hardware platform for this kind of application: the CPU and memory requirements of complex Java applications can be distributed on an extensible set of servers, session to server “stickyness” is managed internally by the application server, for example the Tomcat/ Apache adapter. This component can be extended to actually reject sessions that belong to failed nodes, giving the client application the opportunity to determine the failed actions immediately.

Grid Computing

Some applications can be managed as a series of batch jobs that have to be run with or without dependencies or specific timing requirements. For the flexible management of large amounts of batch jobs, both automatically and manually scheduled, a batch queue system is necessary. SuSE Linux Enterprise Server 8 includes the Software from The Globus Project which provides software tools that make it easier to build computational grids and grid-based applications. These tools are collectively called the Globus Toolkit™.

CONCLUSION

Clustering for availability is highly complex because of the variety of requirements in the different application types. For almost all application types, a successful clustering mechanism can be implemented, but a simple catch-all standard solution does not exist.

The SuSE Linux Enterprise Server 8 includes all the major components for a wide range of solutions and many other leading companies provide well known and proven solutions with SuSE Linux Enterprise Server 8 as a solid and tested platform on which their own solution has been ported for professional deployment in commercial organisations.

© SuSE Linux AG 2002

Linux is a registered trademark of Linus Torvalds.

UNIX is a registered trademark of The Open Group.

All other company, product, and service names or designations may be trademarks or service marks or registered trademarks or service marks of other companies around the world and shall be treated as such.