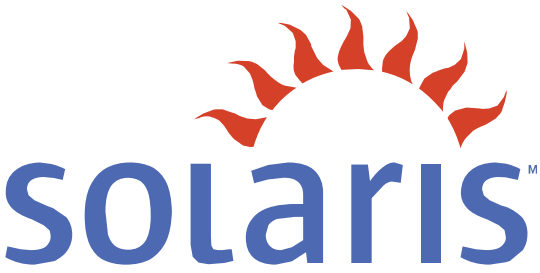
HIGH-AVAILABILITY



Working with Domains, Containers & Zones

Introduction

The industry is rapidly evolving, with new ways to *slice* your computing capacity, each vendor is introducing their own terminology. This paper is designed to provide the reader with a brief introduction at a high level to the terminology used by Sun.

RSF-1 has been tested extensively with these technologies and HAC are able to offer a value added solution by, for instance, failing zones between machines.

Domains

Introduced by Sun, following their acquisition of Cray, this technology is available on enterprise class machines like the E10,000 (E10k), E6900, E15k, E25k etc. Each (physical) domain is allocated physical resources, which may be brought on-line or off-line while the machine is running. The physical resources can not be shared between domains and conceptually a domain is essentially a stand alone enterprise class machine.

Each domain may have a different OS installed and is intended to be completely isolated from the other domains. Figure 1. shows an E10k with a number of different domains. Note that each CPU module, which may include up to 4 CPUs, must be allocated in it's entirety to a domain.

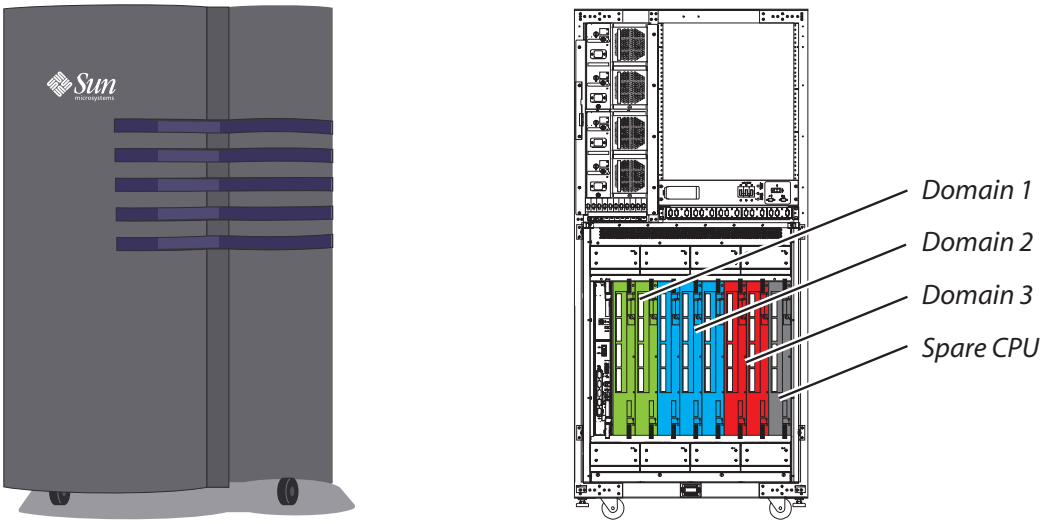


Figure 1.
E10,000 Domains

Each domain must have at least one CPU module to operate. The above example shows three domains, each with two or more CPU modules and a CPU module reserved as a spare. This is purely an example used to illustrate the technology and not a recommended configuration.

Logical Domains

Logical domains were introduced in 2006 with the release of the Niagra chip-set. A CPU or collection of CPUs can be shared by multiple independent operating systems (OSs). The operating systems can not only be different versions but also different *flavours*, for instance Linux, Windows and Solaris.

Figure 2. Illustrates how this might operate.

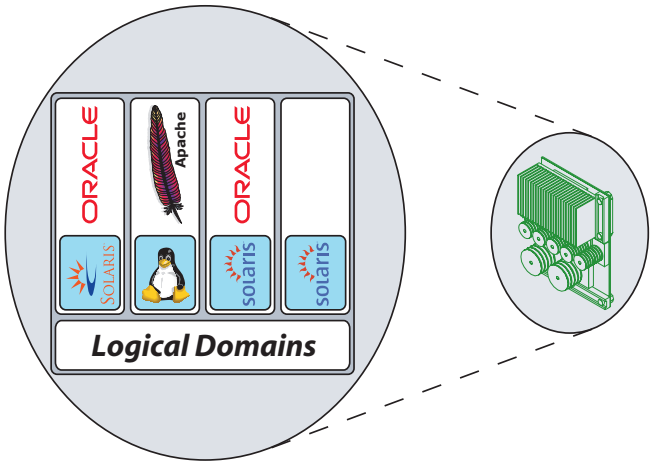


Figure 2.
Logical Domains

Containers

The old name for containers is Resource Manager, which better describes the concept. Containers are defined with various resources or proportion of resources and then processes are started using those allocated resources. The operating system and file system is common to all containers and there is no additional security imposed.

The benefit of containers is that you can allocate defined resources to an application and be confident that, even if another application has become very busy, your *contained* application will continue to operate with the resources you have defined. So one *contained* application will not be adversly affected by another application.

Figure 3. Illustrates the concept of containers.

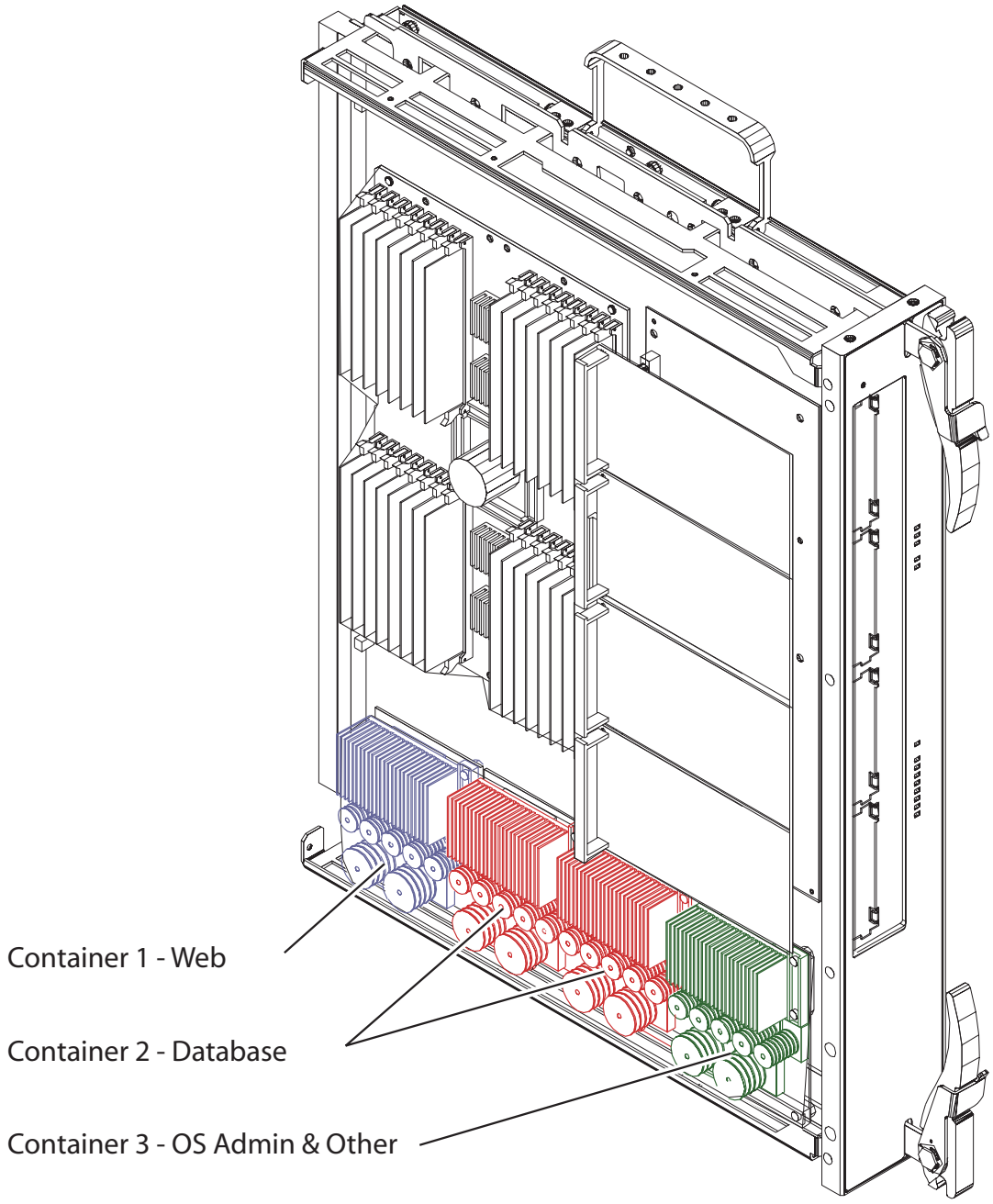


Figure 3.
Solaris Containers

In the above example, a web server has a dedicated CPU, a database has two dedicated CPUs and other applications and administrator activity will be run on a single CPU. Memory and other resources would also be assigned to each container based on rules defined by the administrator.

Containers are defined as ‘projects’ from an administrators point of view and each project is given a name. Then processes, or tasks, can be started by an administrator using the ***newtask*** command with a parameter specifying the appropriate project name. Any sub-processes that are *spawned* will be run *in* the same project/container.

Zones

Zones are an extension of containers according to the marketing material Sun publish. However, while zones can be placed *inside* a container, the principle of zones is rather different. Zones are like virtual operating systems running under a single operating system.

Zones provide security and a limited degree of isolation; zones have their own hostname, password files and IP addresses etc, so appear to remote users as a machine in their own right. Further, each zone can be rebooted and each has it's own file space. However, a fault that results in a kernel panic in one zone will cause the whole machine to reboot, along with all the zones it is running.

Each zone has a copy of the core OS files and these are placed in a sub-directory of the *global zone*. In a clustered environment, where there is a desire to failover zones, then the zone files will be stored on a shared area and each zone will be allocated an independent disk partition. Only a single copy of the operating system is actually running and so upgrades to the OS will affect all zones.

Figure 4. Illustrates how zones might operate.

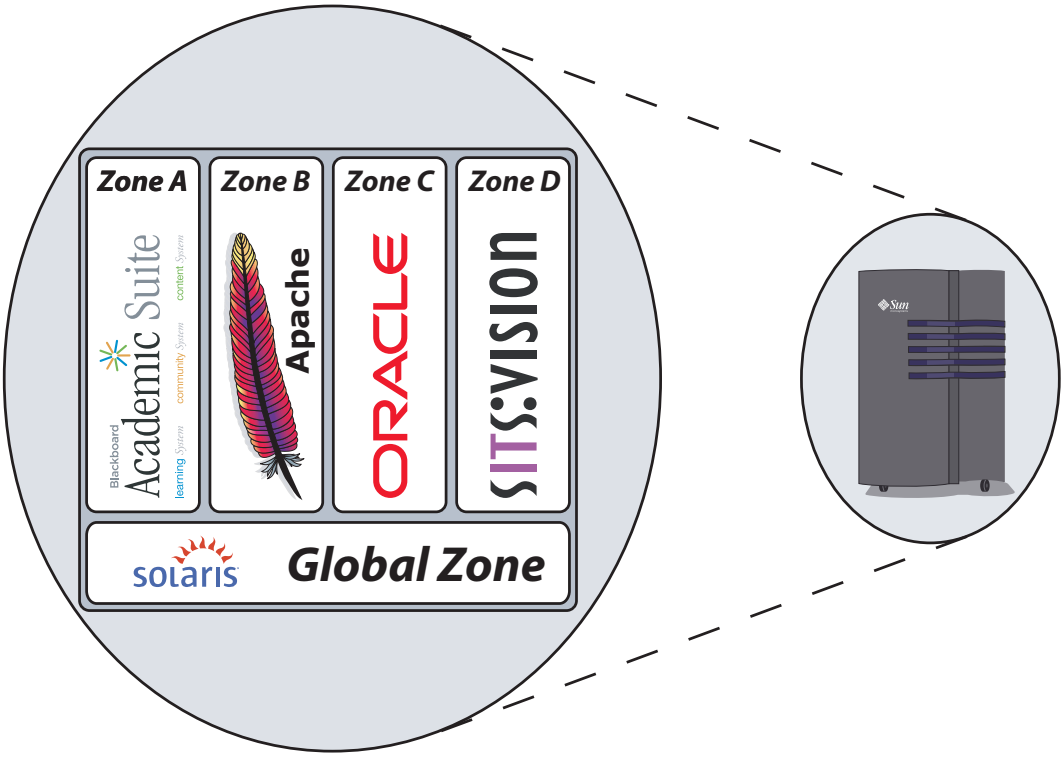


Figure 4.
Solaris Zones

To control the resources used by a zone then containers can be used.

RSF-1 with Domains, Containers & Zones

The environment available to RSF-1 within a logical or physical domain is no different to that provided on different machines. RSF-1 does not endeavour to control the domain resources, that is a task for the administrator, instead RSF-1 monitors the health of the domain, the environment and the application and acts if a fault should occur.

RSF-1 would normally operate outside a container but controls the applications within containers. RSF-1 could be *placed* within a container or even a zone if required but this is not recommended for production systems.

RSF-1 can be used either within a zone, for testing, or more commonly to control zones, enabling the failover of zones to another node if required.