IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 1, Issue 5, Oct-Nov, 2013
ISSN: 2320 - 8791
www.ijreat.org

Virtualization in Solaris with Shared Storage for High Availability

Faizul Hussain Ahamed Mansoor[1]

[1] Pervasive Computing Technology, Bharathidasan Institute of Technology, Anna University Regional Centre, Tiruchirappalli, Tamil Nadu 620024, India

Abstract

Virtualization is a technique that has evolved the industry of information technology to its zenith; because it reduces bottleneck of the physical server requirements like power supply, hardware maintenance, disk extensibility, datacenter space etc. However the current requirements of the business needs still encourage the engineers to come up with strong new ideas that fulfills the need of business which is High availability, this high availability is a feature that can achieved by implementing the Shared storage. Hence if an environment which comprise of Virtualization along with common shared storage that will be a tremendous innovation for achieving high availability of the web services that are hosted on the virtual environment. The features which are mentioned are available in the industry from different vendors which need to be collaborated appropriately to achieve the high availability of the resources.

Keywords: *Virtualization, Solaris, Zones, Storage, LUN, High availability, ZFS.*

1. Introduction

In Information technology service industries availability of the web services for the end users is a major factor that retains the business, if a banking web service runs on a physical server with storage and memory available only on the physical machine it leads to the single point failure of the resources which shall result in loss of millions of dollars in few minutes. But in the meanwhile if the same web service runs on a server which is virtual and the storage is separated from the machine with multiple connectivity to the machines that would avoid the single point of failure and other issues in the physical machine side like power supply, hardware maintenance, disk extensibility, datacenter space etc., that is why the virtualization technology alone is not well enough to achieve the high availability of the resources for the web services and different way of coupling the virtualization is trialed with different other technologies to achieve high availability and here it is coupled with common storage which is separated from the physical server.

2. Solaris Virtualization Overview

Major Solaris Containers Sun's operating system virtualization technique consists of several technologies that work together to foster improved resource management and isolate the environment from the underlying OS. With Solaris Containers, organizations can determine how to allocate resources among applications and services, and ensure they do not interfere with one another [1].

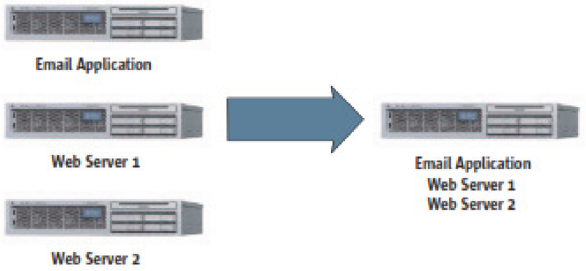


Fig. 1 Virtualized Servers in Physical Solaris Machine.

2.1 Solaris Zones

As part of the consolidation effort, it is important to evaluate the needs of the applications that will share the consolidated system. In this example, the email server and Web server applications need to run in isolated environments that make it appear as if they are running on physically separate machines [2].

Zone 1
Email Application
Zone 2
Web Server 1
Zone 3
Web Server 2

Fig. 2 Representation of Zones in Physical Solaris Machine

This is made possible by a Solaris Container technology called Solaris Zones, which provides separate environments on a machine and logically isolates applications from one another. Each application receives a dedicated namespace in which to run, and cannot see, monitor, or affect applications running in another zone.

3. Storage - NetApp

A Storage is a device which stores all the contents of the applications like coding, images, flash videos etc. sometimes storage shall also include the operating system if it is virtualized OS. The storage may be in size of terabyte hence it is always preferred to make it into several partitions in the devices like NetApp partitions are done by creating a volume then creating a logical unit number (LUN) which is unique, and then this shall be detected on the physical machine to which it is connected[3].

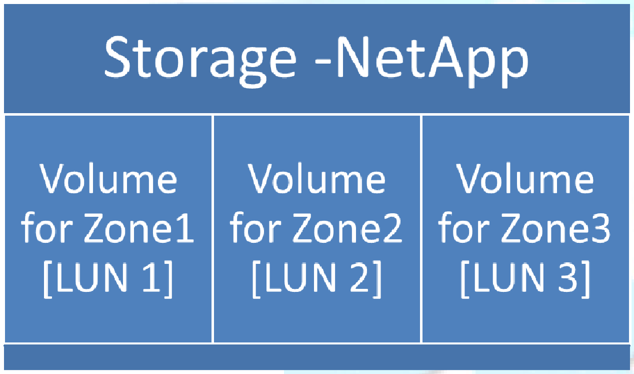


Fig. 3 Storage Architecture.

While creating the volume the size has to decided depending upon our application which will be hosted on the virtual OS, once it is detected on the machine file system can be created based on our requirements and there are different types of file system like Network file system (NFS), Unix file system (UFS), Zeta file system (ZFS) etc. The file system enables us to store the data in the disks of the storage.

4. Proposed System Architecture

As it was described about Solaris machine earlier the architecture is proposed here with single physical Solaris machine which is also called as global zone only when implementing the concept of virtualization, the global zone is the administering zone through which other non-global zones or the virtual servers are created. These non-global zones have their own arguments like number of CPU's, physical memory size, network interface and their IP address fixed by the commands so that each zone will be separated from each other and they shall have their own applications installed in it and may act as email application server or Web server etc.

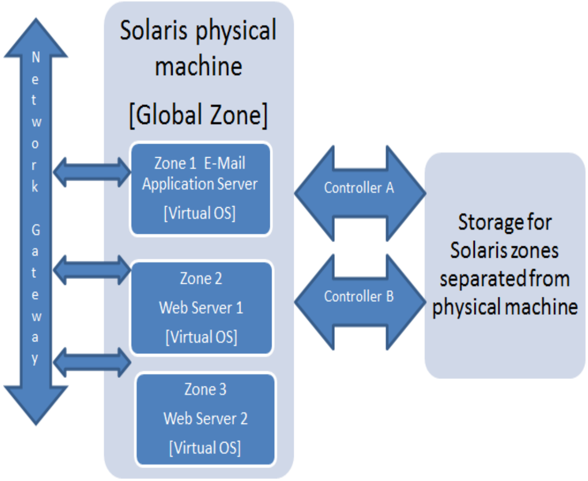


Fig. 4 Proposed Virtualized Architecture with storage separated from Physical Solaris machine.

This entire physical Solaris machine will be connected to the network via switches and router so that it will have the global reachability. On the other side of the Solaris machine is our storage device which will host all the contents of the entire machine including the zones but will be isolated from each other through the concept of creating the volume and Logical Unit Number (LUN) for each zone uniquely [4].

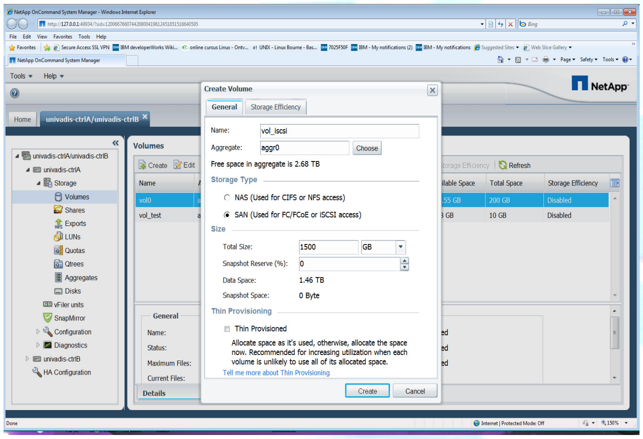
The Physical Solaris machine is connected to the storage via two controllers controller A and controller B that provides higher availability of the content from the storage as out zones are hosting the applications from the storage, this storage would in turn have the dual power supply option one from the normal power unit another from the alternative emergency power unit. So the content will be always available to the global reach via zones and network gateway.

5. Implementation

Solaris is more efficient in command line than operating via console but for the NetApp storage console is easier in terms of initial configuration. To implement this architecture which has been proposed configurations has to be made on both the Solaris physical machine and NetApp storage.

5.1 Storage Configuration on NetApp

LUN Creation: To create a Logical unit number in storage a volume has to be created first then new LUN shall be created from the storage console and the storage used is NetApp [5].



Create Volume

General | Storage Efficiency

Name: vol_iscsi

Aggregate: aggr0 [Choose]

Free space in aggregate is 2.68 TB

Storage Type

○ NAS (Used for CIFS or NFS access)

● SAN (Used for FC/FCoE or iSCSI access)

Size

Total Size: 1500 GB

Snapshot Reserve (%): 0

Data Space: 1.46 TB

Snapshot Space: 0 Byte

Thin Provisioning

□ Thin Provisioned

Allocate space as it's used, otherwise, allocate the space now. Recommended for increasing utilization when each volume is unlikely to use all of its allocated space.

Tell me more about Thin Provisioning

[Create] [Cancel]

Fig. 5 Storage Console

Once the LUN is created in the storage it has to be mapped to group called igroup sun so that the Solaris machine will detect the storage, in order to discover the LUN in Solaris machine following commands need to be executed in the server [6].

- root@solaris# devfsadm –v
- root@solaris#format –e

```
Searching for disks...done
AVAILABLE DISK SELECTIONS:
0.          c0t60A98000323766353352B42367530636Ed0
<NETAPP-LUN-810a-5.00GB>
/scsi_vhci/ssd@g60a98000323766353352b42367530636e
…
```

- Select : 0
- >partition
- >print
- >label
- devfsadm –v
- scdidadm –r
- cldevice populate

To look for the NetApp device in our server

- scdidadm –L

```
solaris:/dev/rdsk/c0t0d0   /dev/did/rdsk/d1
solaris:/dev/rdsk/c1t7d0   /dev/did/rdsk/d2
solaris:/dev/rdsk/c2t3E9108DBAE6F3437d0
/dev/did/rdsk/d3
solaris:/dev/rdsk/c5t5000CCA02532CF9Cd0
/dev/did/rdsk/d4
solaris:/dev/rdsk/c5t5000CCA025339310d0
/dev/did/rdsk/d5
solaris:/dev/rdsk/c5t5000CCA0252CDC08d0
/dev/did/rdsk/d6
solaris:/dev/rdsk/c5t5000CCA0250CA168d0
/dev/did/rdsk/d7
solaris:/dev/rdsk/c0t5000C50047BAEFBFd0
/dev/did/rdsk/d8
solaris:/dev/rdsk/c0t5000C50047BB0DABd0
/dev/did/rdsk/d9
solaris:/dev/rdsk/c0t5000C50047BA9E43d0
/dev/did/rdsk/d10
solaris:/dev/rdsk/c0t5000C50047BB4BE3d0
/dev/did/rdsk/d11
solaris:/dev/rdsk/c1t0d0   /dev/did/rdsk/d12
solaris:/dev/rdsk/c2t7d0   /dev/did/rdsk/d13
solaris:/dev/rdsk/c3t3952332D80AE5D89d0
/dev/did/rdsk/d14
```

Disk to be used for Zpool and ZFS creation

```
solaris:/dev/rdsk/c0t60A98000323766353352B4236753063
6Ed0 /dev/did/rdsk/d15
```

ZPOOL and ZFS file system creation: As the storage is detected on our Solaris machine Zpool and its respective ZFS has to be created on the storage so that the zones can be installed into it. In order to create those Zpool and ZFS following commands has to be executed on the Solaris machine.

- root@solaris # zpool create zpool-1 /dev/rdsk/c0t60A98000323766353352B42367530636Ed0
- root@solaris # zpool list

```
NAME      SIZE    ALLOC     FREE      CAP
HEALTH  ALTROOT
zpool-1  98.97G    91K  98.97G    0%  ONLINE
```

- root@solaris # zfs create zpool-1/zfs-1

- root@solaris # zfs list

```
NAME   AVAIL          USED          REFER
MOUNTPOINT
zpool-1        98970M  4.35G   32K  / zpool-A
zpool-1/zfs-1     98970M   98.35G    98970M   /
zpool-1/zfs-1
```

Create a directory in ZFS

- root@solaris# mkdir /zpool-1/zfs-1/dir-1
- root@solaris#chmod 700 /zpool-1/zfs-1/dir-1

5.2 Zone Configuration

Zones are of two categories they are whole root zone and sparse root zone. Whole root zones that does not share the resources like CPU, LWP, Physical memory, Swap memory etc. but in sparse root zone all these will be in pool and shall be utilized by any zones at any point of time provided there should not be resource scarcity at any instance.

Zone Configuration: Here requirement is not too critical so sparse root zone is created for this environment; the following commands would create a non-global sparse root zone[6].

- root@solaris# zonecfg -z zone1

zone1: No such zone configured
Use 'create' to begin configuring a new zone.

- zonecfg:zone1> create
- zonecfg:zone1> set zonepath=/zpool-1/zfs-1/dir-1

This is where the zone will be created. The above mentioned path points to the file system we mounted earlier. This restricts the zone size to the size of this file system. (i.e. 100GB in this case).

- zonecfg:zone1> add dedicated-cpu
- zonecfg:zone1:dedicated-cpu> set ncpus=1
- zonecfg:zone1:dedicated-cpu> end

One CPU is dedicated to this zone. Please note that the zone cannot be booted if adedicated CPU is not available. Also, on a multi-processor machine, only 1 CPU will be visible to the zone.

- zonecfg:zone1> add capped-memory
- zonecfg:zone1:capped-memory>set physical=100m
- zonecfg:zone1:capped-memory> set swap=512m
- zonecfg:zone1:capped-memory> end

Memory constraints for the available physical and swap memory.

- zonecfg:zone1> set scheduling-class=FSS

The zone will use a Fair Share Scheduler.

- zonecfg:zone1> set ip-type=explicit

The ip-type parameter supports two types i.e. shared and explicit. Here, we will consider the explicit type. For the explicit ip-type, the IP layer configuration and state is dedicated to the particular non global zone which is zone1.

- zonecfg:zone1> add net
- zonecfg:zone1:net> set address=141.89.226.31
- zonecfg:zone1:net> set physical=ce0
- zonecfg:zone1:net> end

By adding a network resource, we have provided the zone with a unique IP address, so that the zone can be uniquely identified from the outside world. That is, even though the IP layer state and configuration are shared with the global zone, traffic can still be routed explicitly to the non-global zone.

- zonecfg:zone1> add attr
- zonecfg:zone1:attr> set name=comment
- zonecfg:zone1:attr> set type=string
- zonecfg:zone1:attr> set value="Email Application server"
- zonecfg:zone1:attr> end

This is just a comment.

- zonecfg:zone1> verify

The verify command is used to ensure correctness of the configuration. However, certain aspects (e.g. path to the file system) can only be verified during zone installation. Please note that these paths must have been created prior to installing the zone.

- zonecfg:zone1> commit
- zonecfg:zone1> exit

Zone Installation and Booting: Once the above steps for configuring the zones are done an unique id for that specific zone is created and all the mentioned parameters for the zones are saved in XML format in file /etc/zones which is a configuration file for that zone. After the configuration of the zone the OS need to be installed into it so that a complete virtual environment will be available.

- root@solaris#zoneadm -z zone1 install

```
Preparing to install zone <my-zone>.
Creating list of files to copy from the global zone.
Copying <14339> files to the zone.
Initializing zone product registry.
Determining zone package initialization order.
Preparing to initialize <1203> packages on the zone.
Initialized      <1203>      packages      on      zone.
Zone <zone1> is initialized.
```

Installation of <1> packages was skipped.
The file </export/home/my-zone/root/var/sadm/system/logs/install_log> contains a log of the zone installation.

- root@solaris# zoneadm –z zone1 boot

Virtual OS called zone1 will be booted and shall be connected via IP 141.89.226.31 through any of the remote access tool like putty.

6. Conclusions

The applications which are hosted on the zones does not exhibit any difference to the end user or even to the developers as they would be working on the server as if it is a physical server but only the administrator who has control knows that this server has three virtual servers in it namely zone1, zone2, zone3 in which first zone has the email application server hosted on it, while the second and third zones has the web applications hosted on it. Thus it provides the isolation of the contents of the different virtual servers and storage is located away from the physical machine which gives dual controller access between them, thus it provides the High availability of the resources for the web services which are hosted and the end user shall not get the apology message like "server temporarily not available" on their browser, thus it reduces the single point of failure and increases the availability of the services that is one of the necessity of the web services era. Further this work shall be enhanced with advanced technologies like SAN storage, Clustering of Zones to increase the availability of web services by considering the entire environment.

References

[1] "Best practices for running oracle databases in oracle Solaris containers", Oracle White paper, Oracle, April 2010.
[2] "System Administration Guide : Oracle Solaris containers – Resource Management and Oracle Solaris Zones", Part No: 817–1592–21, Oracle, January 2013.
[3] Jorge Costa,"Clouds: Running Multiple Oracle Customers With Solaris Zones on NetApp MultiStore" NetApp, February 2010.
[4] Sajid Akhtar, and Robert Esker, "OpenStack Deployment And Operations Guide NetApp Inc., April 2013
[5] "NetApp storage deployment guide", NetApp & Cisco
[6] "NetApp® Block Access Management using Open Interfaces", Network Appliance Inc., March 2010.
[7] "System Administration Guide: Oracle Solaris 8 Containers", April 2011