# Migration vplex-Invista to PMAX on Serviceguard cluster

## Introduction

### Why

Vplex-Invista versus PMAX-Symetrix.

Vplex-Invista storage bay are near end of life.

Moreover, some incident occured on vplex-Invista storage bay.

### How

First, Storage team adds new PMAX LUN.

Next, Linux team switches the vplex lun with the pmax ones without any service interruption using mdadm.

Last, Storage team removes the vplex-invista lun.

## Adding PMAX LUN

Storage team adds new LUN from PMAX.

The new luns will be a little bigger than the vplex-Invista ones. (Ex : 128G -> 140G)

To see the new LUNs, you can wait for a server reboot or you can try rescan the disk :

for i in 0 1 2 3;do echo "- - -" > /sys/class/scsi\_host/host$i/scan;done

## Switching vplex and PMAX LUN

### Introduction bis

For each mdadm device, we will add a new PMAX disk while removing one vplex-Invista disk.

We update alias in multipath.

Then we wait for mdadm synchronization.

Next, we add the second PMAX disk while removing the last vplex-Invista disk.

We update alias in multipath.

Then we wait for mdadm synchronization.

Lastly, if we are able, we reboot the server and if not, we keep watch for the next server reboot.

### Pre-requisites

Storage team will send the matching table between old vplex-Invista luns and pmax new ones.

Linux team sees the new luns on both servers of the cluster.

There is no incident on the cluster you want to migrate (verify the status of all the packages).

### Migration script

There is a migration bash script to generate mdadm and multipath commands.

First, copy this script on one server of the cluster and make it executable. (chmod +x <script>)

#!/bin/bash

#set -x

echo "Generation commandes mdadm"

for ligne in `cat $1`

do

OLD\_WWN=`echo ${ligne}|awk -F";" '{print $1}' | tr '[:upper:]' '[:lower:]' `

NEW\_WWN=`echo ${ligne}|awk -F";" '{print $2}' | tr '[:upper:]' '[:lower:]' `

OLD\_DM=`multipath -ll| grep $OLD\_WWN | cut -d" " -f3`

NEW\_DM=`multipath -ll| grep $NEW\_WWN | cut -d" " -f3`

MD\_RAID=`cat /proc/mdstat |grep ^md | grep "${OLD\_DM}\["| cut -d" " -f1 `

echo "mdadm --manage /dev/${MD\_RAID} --add /dev/${NEW\_DM} && mdadm --manage /dev/${MD\_RAID} --fail /dev/${OLD\_DM}"

done

echo "Generation commandes multipath"

for ligne in `cat $1`

do

OLD\_WWN=`echo ${ligne}|awk -F";" '{print $1}'| tr '[:upper:]' '[:lower:]' `

NEW\_WWN=`echo ${ligne}|awk -F";" '{print $2}'| tr '[:upper:]' '[:lower:]' `

echo "sed -i -e \"s/${OLD\_WWN}/${NEW\_WWN}/g\" /etc/multipath.conf"

done

This script needs a argument file to work.

You will make two of them using the lun matching table provided by storage team.

Each one of these two files will list the old lun and the new lun from one datacenter room storage bay.

On homologation, one will list GSA luns and the other GSC.

On production, one will list HDI luns and the other HDM.

/ ! \ If you don’t distinguish luns from their storage bay localization, you may delete both disks of a mdadm raid device and lose all data !!!

These are example for the files to set in the migration script argument.

Format : old\_wwn ;new\_wwn

[root@slzuyncsgl39 ~]# cat /root/GSA.txt

6000144000000010A015272860B2ADB8;60000970000297600513533030343239

6000144000000010A015272860B2ADCC;60000970000297600513533030343241

6000144000000010A015272860B2ADE0;60000970000297600513533030343242

…

[root@slzuyncsgl39 ~]# cat /root/GSC.txt

6000144000000010A015A3158BFFE2AD;60000970000297600513533030343344

6000144000000010A015A3158BFFE2B8;60000970000297600513533030343345

6000144000000010A015A3158BFFE2B3;60000970000297600513533030343346

…

### Script execution

You will execute the script for one datacenter room and have to wait for synchronization before doing the other.

The script execution wil provide you a list of commands to execute :

This example is missing some inline parts due to be an already migrated cluster. (device names mostly)

[root@slzuyncsgl39 ~]# ./migre\_pmax.sh /root/GSA.txt

Generation commandes mdadm

mdadm --manage /dev/md8 --add /dev/dm-22 && mdadm --manage /dev/md9

md8 --fail /dev/

mdadm --manage /dev/md8 --add /dev/dm-2 && mdadm --manage /dev/md9

md8 --fail /dev/

mdadm --manage /dev/md8 --add /dev/dm-3 && mdadm --manage /dev/md9

md8 --fail /dev/

…

Generation commandes multipath

sed -i -e "s/6000144000000010a015272860b2adb8/60000970000297600513533030343239/g" /etc/multipath.conf

sed -i -e "s/6000144000000010a015272860b2adcc/60000970000297600513533030343241/g" /etc/multipath.conf

sed -i -e "s/6000144000000010a015272860b2ade0/60000970000297600513533030343242/g" /etc/multipath.conf

…

### Use mdadm commands from migration script

We recommand to execute the mdadm commands at the same times.

You have to run it on only the node of the cluster which is currently mounting the mdadm device.

The command for one disk should be :

mdadm --manage /dev/md8 --add /dev/dm-22 && mdadm --manage /dev/md9

md8 --fail /dev/xx

So you have to execute all the commands provided by the migration script, waiting between some of them to avoid running too many synchronization at the same time.

### Use multipath commands from migration script

The multipath commands will only modify /etc/multipath.conf to rename mpath alias.

**You have to run multipath;conf update on both nodes of the cluster.**

You can run it on the disk while you wait for its mdadm synchronization or after.

After the /etc/multipath.conf modification, you can refresh multipath cache through :

**(Warning : Don’t restart the multipath service in the server that manage packages)**

multipath –F

multipath –d

multipath –v2

The mpath in use will not be renamed and will have to wait for a package switch or a reboot.

### Wait for mdadm synchronization

You can follow the mdadm synchronization through :

cat /proc/mdstat | grep -E ‘active|recovery’

watch -n2 cat /proc/mdstat

/ !\ The time for synchronization can vary a lot from what this command return because of applicative usage/activities.

To speed up the synchronization if needed :  
echo 500000 >/proc/sys/dev/raid/speed\_limit\_max (by default : 200000).

echo 500000 >/proc/sys/dev/raid/speed\_limit\_max (by default : 1000)

At the end of operation, execute this command :  
echo 200000 >/proc/sys/dev/raid/speed\_limit\_max .

echo 1000 >/proc/sys/dev/raid/speed\_limit\_max

### Plan reboot or keep watch on it

When you have execute the migration for both storage bay room, you can go on in this part.

If one node of the cluster is not bearing any package.

You can reboot it. (Don’t forget nosup in production !)

When it comes back, check its multipath alias.

What the reboot should refresh : multipath cache and mdadm removed disk cleaning.

Next, if you have a switch-over downtime planned, you can switchover all package to this node and reboot the other one.

If you have not a switch-over downtime planned, keep a close watch on the other node reboot by PCL.

Once both node were rebooted and all is fine on the cluster, the cluster is migrated and the removing of vplex luns can proceed.

### Troubleshooting

In the recent past, we lost a pmax storage bay at this step.

One of the pmax disks became unavailable and mdadm set back the old vplex disk in the mdadm raid.

To solve this problem, we did the migration again using the same commands but we planned both servers reboot to clean mdadm removed disks.

## Remove vplex LUN

/ ! \ Before everything, you have to check that no vplex-Invista is in use by a SGL package (Both multipath and mdadm). (GO/NOGO from you.)

The Storage team will do most actions in this.

During intervention, multipath will be very slow to answer and there will be a lot of error logs in /var/log/messages.

When all the vplex-Invista luns are removed for a cluster from both sites (Chartres Diderot/D’Alembert or Gradignan Salle A/Salle C), the error logs will stop and vplex-Invista lun will remain with no path available in multipath.

Easy check : Multipath will reply as usual (quickly) and there will be no more path on Invista.

multipath -ll | grep Invista

multipath -ll | grep SYM

Then Linux team can remove the luns from multipath cache :

multipath -F

multipath –d

multipath –v2

Easy check : Multipath will reply as usual (quickly) and there will be no more lun on Invista.

multipath -ll | grep Invista

multipath -ll | grep SYM