Best Practice on Data Migration with EMC Open Replicator for Symmetrix

EMC Proven™ Professional Knowledge Sharing
September, 2007

Henry Zhang
Henry.zhang@eds.com
EDS Canada
# Table of Contents

- Project background: ............................................................................................................... 3
- Project challenge: .................................................................................................................. 3
- EMC Open Replicator for Symmetrix ................................................................................... 3
- Data Migration with Open Replicator .................................................................................. 4
- Implement Data Migration with Open Replicator for Symmetrix ....................................... 5
  - Pre-migration planning ...................................................................................................... 5
  - Migration Operation ......................................................................................................... 6
  - Post-migration clean up ................................................................................................... 7
- Best practices on data migration with Open Replicator for Symmetrix ............................... 7
  - Pre-migration planning ...................................................................................................... 8
  - Migration Operation ......................................................................................................... 10
  - Post-migration clean up ................................................................................................... 11
- EDS migration tool and scripts ............................................................................................ 12
- EDS San Support Software .................................................................................................. 12
- Migration scripts: ............................................................................................................... 13

---

*Disclaimer: The views, processes or methodologies published in this compilation are those of the author. They do not necessarily reflect EMC Corporation’s views, processes, or methodologies.*
Project background:
We had three DMX1000 arrays at end of lease in December, 2006 and needed to migrate data to two new DMX3-24 arrays before the due date. There were more than 120 servers, nearly 1900 LUNs and approximately 50TB of data involved in this migration.

Project challenge:
The technical challenge was that the servers each represented different hardware platforms like SUN-Solaris, HP-UX, IBM-AIX, Window-intel; multiple Volume managers like Veritas Volume Manager, Solaris Volume Manager, HP LVM, IBM LVM; and a variety of popular cluster technologies such as VCS, Sun Cluster, HP MC/SG, MSCS.

But, there were also other challenges. Change management was significant since the migration would impact more than fifteen client production and development servers. We needed software to make data migration easy, fast, and reliable in the face of complex technical configurations.

EMC Open Replicator for Symmetrix
Open Replicator for EMC Symmetrix is a high-speed data mobility, migration and distribution solution software. It provides platform-independent replication between a Symmetrix DMX™ and any qualified storage system like Symmetrix DMX or CLARiiON® with full or incremental copy capabilities over the existing infrastructure. It is designed as part of the Symmetrix Enginuity™ operating environment, and is compatible with the EMC SRDF® and TimeFinder® families.
Data Migration with Open Replicator

High-speed online data migration is one of Open Replicator’s most significant features. It provides array-to-array high speed, block level replication through the existing SAN infrastructure.

Open Replicator migrates data at the array LUNs block level. As a result, migration is platform independent to any type of open system, no matter the host operating system or application.

There are many replication solutions in the market today. Some may select a host-based utility like Veritas Volume Manager that provides data replication at OS level. I was involved in a host-based data migration project three years ago that included servers with SUN Solaris Operating System, and Veritas Volume Manager using EMC arrays.

My comparisons on these two migrations are below.

Both Array-based migration and Host-based migration share features like:
- Simplicity and reliability
- Flexibility
- Shell/CLI based programmable
- Minimal business impact

For Veritas Volume Manager host-based data replication:
- Flexible on LUNs level: it allow the user to re-layout Volumes and File Systems over new LUNs physically (changing LUNs size, LUNs number) and logically (changing on raid type) during data migration.
- Migrations are on O.S/VXVM/VXFS level by Veritas Volume Manager
- Joint-effort between Storage Team and SA team
- Two reboots on server

For EMC Open Replicator array-based data migration:
- Flexible on LUNs logical level (changing on raid type) during data migration
- Migrations are on LUNs in Array at block level by Open Replicator for Symmetrix
- Symmetrix-based controls
- Faster
- Effort at Storage Team primarily; SA Team verifies.
- Single reboot on server
- Open – platform independent on both host and array level

In this project, we used Open Replicator to migrate existing data from three old DMX-1000 arrays to two new DMX3 arrays in a business critical environment.
Implement Data Migration with Open Replicator for Symmetrix

Open Replicator for Symmetrix offers multiple functions like Data Mobility, Remote Vaulting and Migration. These best practices only focus on the Data Migration segment that migrates data from a retired array to the new array.

Like most solutions, Data Migration with Open Replicator involved three phases:

- pre-migration planning
- migration operation
- post-migration clean-up

There are many tasks in each phase. The steps below describe a hot pull copy session of Open Replication. “Hot” means that the application/database remains online during data migration. “Pull” means that the copy direction is from the old to the new array.

In a “hot pull” Open Replicator copy session, the LUNs on the new array can be read/write enabled to the host, but LUNs on the old array should not be accessible from its host. The host could access data on the new Symmetrix array before the migration was completed.

**Pre-migration planning**

One key output of this phase is device file(s) that list the control and remote device pairs that can be used for migration operations detailed in step 5.2. **These device pairs could be documented by host or group of hosts.**

- Identify current host configuration
  Migration plans are host-based. We must identify host connected HBAs (number, WWN, Fabric), attached LUNs (number, size, raid-type), FAs mapped information, and LUN masked on HBAs.

- New devices planning, configuration and mapping
  Create LUN/Meta devices on the new array. These new LUNs should be the same size, same raid type (not necessary) as the current host configuration. They must also be mapped to new assigned front-end adapters.

- FA zoning
  Create a new zone if it does not exist in the Fabric that contains FA which server LUNs mapped to in old array; and FA which LUNS were just created and mapped to on new array.

- Host zoning
  Create new zones in the fabric that contain the server HBA and FA which new LUNs mapped to in new array.
- **FA masking**
  This step is important and can only be done through SYMCLI, not by EMC ControlCenter®. It would let new array FAs “see” all LUNs on the old array.

**Migration Operation**

- **Shutdown server**
  Take the server offline. This is a pre-requirement of changing host masking. In order to let the server drop LUNs on the old array and view LUNs from the new array, we need to shutdown the application/database and server one time.

- **Startup data migration**
  Create an Open Replicator copy session to establish connections between new LUNs and old LUNs from the previous device file. Activating the data migration session starts the data migration from the old array to the new array.

- **Host masking changing**
  This step would remove host LUNs masking from the old array FAs and add new LUNs masking to the new array FAs. It lets the server access LUNs to the new array and blocks LUNs from the old array to the server.

- **Boot server**
  You can boot the server up as long as the migration session has been activated; you don’t need to wait until the migration is complete. This reduces server/application down time. But, keeping the server offline makes migration speed much faster. Another option is to hold booting the server until LUNs migration is complete.

- **Host verification**
  We must ensure that the server can see all the LUNs which have just been assigned from the new array. Also, we need make sure that all the old device files have been cleaned from the OS.

- **Migration speed tuning**
  Set the session pace to designate how fast data is copied between devices. The range is between 0-9, with 0 being the fastest and 9 being the slowest.

  Set the ceiling value, the maximum percentage of bandwidth. The percentage for an FA port or all FA ports are between 0-100; you also could set it to NONE to turn off the ceiling function. The session pace is ignored unless the ceiling value is set to NONE.

- **Migration monitoring**
  You can monitor migration sessions until all LUNs are in “Copied” state. You can also find the migration session completion percentage and the estimated time of completion. Terminate the session after migration has been completed.
Migration verification
This is host verification that includes HBA verification, Device Manager verification, PowerPath® verification, Volume Manager verification, application/database verification, as well as cluster fail-over.

**Post-migration clean up**
We need to clean up all the related configurations in the old array and fabric.

- **Host zoning clean up**
  Remove each zone which contains server HBA to old array FA(s) from Fabric.

- **FA zoning clean up**
  Remove the FA (old array) to FA (new array) zones created during pre-migration from Fabric.

- **FA masking clean up**
  Clean previous FA masking that was set during pre-migration.

- **Devices clean up**
  Unmap the old array LUNs device from its FAs.

**Best practices on data migration with Open Replicator for Symmetrix**
There are many ways to perform array-based migration, such as migrating the whole array at one time or conducting an FA pairs-based migration.

However, we decided to this work by “host”. We migrated 10-20 servers on scheduled weekends by client-base. This made data migration well-planned, impacted the client less, and minimized resource workload.

Below, I will use one test server **camktest** as an example to illustrate the process that supported the successful migration of three arrays in the last year. It stands on Open Replicator hot pull copy session.

Illustration follows.
**Pre-migration planning**

All the steps in pre-migration planning can be done during regular business hours without impacting hosts that are running. This reduces the real migration change window.

- Identify host configuration:

  New target array: 1334  
  Old source array: 0528  
  Host type: Windows 2003  
  HBA type: Qlogic

<table>
<thead>
<tr>
<th>Server</th>
<th>Source Array</th>
<th>Source Vol</th>
<th>Source Type</th>
<th>Source Size</th>
<th>Source FA1</th>
<th>Source FA2</th>
<th>Source Fabric</th>
</tr>
</thead>
<tbody>
<tr>
<td>camktest</td>
<td>0528</td>
<td>667</td>
<td>R5</td>
<td>8.51</td>
<td>13A1</td>
<td>4A1</td>
<td>1/2</td>
</tr>
<tr>
<td>210000e08b1e3cf2/1</td>
<td>0528</td>
<td>688</td>
<td>R5</td>
<td>68.08</td>
<td>13A1</td>
<td>4A1</td>
<td>1/2</td>
</tr>
<tr>
<td>210100e08b3e3cf2/2</td>
<td>0528</td>
<td>690</td>
<td>R5</td>
<td>68.08</td>
<td>13A1</td>
<td>4A1</td>
<td>1/2</td>
</tr>
</tbody>
</table>

Host camktest has three LUNs on array 0528, there are 667(8.51G), 688(68.08G), 690(68.08G) and all are at raid 5. All LUNs are mapped to both FA 13A1 and FA 4A1 on array 0528.

FA 13A1 WWN 50060482cae0542c on array 0528 is connected to fabric 1,  
FA 4A1 WWN 50060482cae05423 on array 0528 is connected to fabric 2.

Host camktest has two HBA, all there LUNs are masked on both HBAs.  
HBA1 WWN 210000e08b1e3cf2 go to fabric 1,  
HBA2 WWN 210100e08b3e3cf2 go to fabric 2.
New devices planning, configuration and mapping:

<table>
<thead>
<tr>
<th>Server</th>
<th>Tgt Array</th>
<th>Tgt Vol</th>
<th>Tgt Type</th>
<th>Tgt size</th>
<th>Tgt FA1</th>
<th>Tgt FA2</th>
<th>Tgt Fabric</th>
</tr>
</thead>
<tbody>
<tr>
<td>camktest</td>
<td>1334</td>
<td>1035</td>
<td>R5</td>
<td>8.51</td>
<td>9A:0</td>
<td>8A:0</td>
<td>1/2</td>
</tr>
<tr>
<td>210000e08b1e3cf2/1</td>
<td>1334</td>
<td>6F2</td>
<td>R5</td>
<td>68.08</td>
<td>9A:0</td>
<td>8A:0</td>
<td>1/2</td>
</tr>
<tr>
<td>210100e08b3e3cf2/2</td>
<td>1334</td>
<td>6FA</td>
<td>R5</td>
<td>68.08</td>
<td>9A:0</td>
<td>8A:0</td>
<td>1/2</td>
</tr>
</tbody>
</table>

Create LUNs on new array 1334 with same LUN size and LUN raid-type: 1035(8.51G), 6F2(68.08G), 6FA(68.08G). All LUNs need to map to new array 1334 on FA9A:0 and FA8A:0

WWN of array 1334 FA 9A:0 is 50060482d52dd588. This FA connected in Fabric 1. WWN of array 1334 FA 8A:0 is 50060482d52dd587. This FA connected in Fabric 2.

Map new luns to assigned FAs(9A0,8A0) on array 1334

```
# cat camktest.map
map dev 6f2 to dir 09a:0, lun=01E;
map dev 6f2 to dir 08a:0, lun=01E;
map dev 6fa to dir 09a:0, lun=01F;
map dev 6fa to dir 08a:0, lun=01F;
map dev 1035 to dir 09a:0, lun=020;
map dev 1035 to dir 08a:0, lun=020;
```

# symconfigure -sid 1334 -file camktest.map -v -nop preview
# symconfigure -sid 1334 -file camktest.map -v -nop prepare
# symconfigure -sid 1334 -file camktest.map -v -nop commit

Create below device file for further migration:

```
# cat camktest.txt
symdev=000190101334:1035    symdev=000187400528:667
symdev=000190101334:6f2     symdev=000187400528:688
symdev=000190101334:6fa     symdev=000187400528:690
```

FA zoning -- Create zones with old array FA and new array FA:

Zone name OR_528_13A1_1334_9A0 in Fabric 1 with
FA-9A@000190101334 port 0 & FA-13A@000187400528 port 1;

Zone name OR_528_4A1_1334_8A0 in Fabric 2 with
FA-8A@000190101334 port 0 & FA-4A@000187400528 port 1;

Host zoning -- Create zones from server HBAs to new array FAs:

Zone name CAMKTEST_DMX1334_FA9A0 in fabric 1 with
FA-9A@000190101334 port 0 & adapter1@camktest(WWN 21000e08b1e3cf2);

Zone name CAMKTEST_DMX1334_FA8A0 in fabric 2 with
FA-8A@000190101334 port 0 & adapter2@camtest (WWN 210100e08b3e3cf2);
Activate the modified zone set on each fabric. Use the command below to verify that array 0528 FA 13A:1 could see array 1334 FA9:0.

```
# symmask -sid 528 -dir 13a -p 1 list logins
```

Symmetrix ID : 000187400528
Director Identification : FA-13A
Director Port : 1

<table>
<thead>
<tr>
<th>User-generated Identifier</th>
<th>Type</th>
<th>Node Name</th>
<th>Port Name</th>
<th>FCID</th>
<th>In</th>
<th>Fabric</th>
</tr>
</thead>
<tbody>
<tr>
<td>50060482d52dd588 Fibre</td>
<td></td>
<td>50060482d52dd588 50060482d52dd588 50060482d52dd588 080033 No</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- FA masking --- it would let new array FAs see all LUNs on the old array.

```
#  symmask –sid 0528 –dir 13a –p 1 –wwn 50060482d52dd588 add dev 667
#  symmask –sid 0528 –dir 13a –p 1 –wwn 50060482d52dd588 add dev 688
#  symmask –sid 0528 –dir 13a –p 1 –wwn 50060482d52dd588 add dev 690
#  symmask –sid 1334 –wwn 210000e08b1e3cf2 –dir 09a –p 0 add devs 1035
#  symmask –sid 1334 –wwn 210000e08b1e3cf2 –dir 09a –p 0 add devs 6f2
```

**Migration Operation**

All steps of below migration operation should be done within change window.

- Shutdown application/database and server
  For a Unix server, put new LUNs configuration file in place before shutting down the server (like on Solaris sd.conf, lpfc.conf); For a windows host, make sure to enable auto-mapping on HBA like emulex.

- Startup data migration
  Create an Open Replicator Copy Session:

  Activate Open Replicator Session:
  # symrcopy activate –file camktest.txt –v –nop

- Changing Host masking
  This step will enable read/write to the host on the new array, making the remote device on the old array inaccessible from its host. It is the key step for an Open Replicator hot pull copy session.

```
# symmask -sid 1334 -wwn 210000e08b1e3cf2 -dir 09a -p 0 add devs 1035
# symmask -sid 1334 -wwn 210000e08b1e3cf2 -dir 09a -p 0 add devs 6f2
# symmask -sid 1334 -wwn 210000e08b1e3cf2 -dir 09a -p 0 add devs 6fa
# symmask -sid 1334 -wwn 210100e08b3e3cf2 -dir 08a -p 0 add devs 1035
# symmask -sid 1334 -wwn 210100e08b3e3cf2 -dir 08a -p 0 add devs 6f2
```
Boot server
Boot the server up after the migration session has been activated or wait until the migration session is complete. The boot session will need to re-discover the devices tree on the new array.

Host verification
Make sure only LUNs from new array are visible to the host on each HBA.

Migration speed tuning
# symrcopy set ceiling 80 –sid 1334 –dir 09a –port 0
or
# symrcopy set pace 0 –file camktest.txt

Migration monitoring
# symrcopy query –file camktest.txt -i 30
This command will show you migration detail, like migration percentage, and will estimate time to completion.

# symrcopy verify –file camktest.txt
This command will verify that all the device-pairs in the control file camktest.txt are in “copied” state from “CopyInProg” state.

# symrcopy terminate –file camktest.txt
This command will let you terminate the migration copy session after migration is completed.

Migration verification
This is the last step of the migration process. It is host-based checking including all the components such as HBA, device manager/volume manager, PowerPath, cluster, application and database.

Post-migration clean up
Host zoning clean up
Remove host-FA zones which previous existing in the fabric:
Zone name: CAMKTEST_DMX528_FA13A1 in fabric 1 with FA-13A@000187400528 port 1 & adapter1@camktest (WWN 210000e08b1e3cf2);
Zone name: CAMKTEST_DMX528_FA4A1 in fabric 2 with FA-4A@000187400528 port 1 & adapter2@camktest (WWN 210100e08b3e3cf2);
FA zoning clean up
Remove FA-FA zones which had been created in migration planning phase from fabric:
Zone name OR_528_13A1_1334_9A0 in Fabric 1 with
FA-9A@000190101334 port 0 & FA-13A@000187400528 port 1;
Zone name OR_528_4A1_1334_8A0 in Fabric 2 with
FA-8A@000190101334 port 0 & FA-4A@000187400528 port 1;

FA masking clean up
Remove FA-LUNs masking which had been created in migration planning phase:
# symmask –sid 0528 –dir 13a –p 1 –wwn 50060482d52dd588 remove dev 667
# symmask –sid 0528 –dir 13a –p 1 –wwn 50060482d52dd588 remove dev 688
# symmask –sid 0528 –dir 13a –p 1 –wwn 50060482d52dd588 remove dev 690
# symmask –sid 0528 –dir 4a –p 1 –wwn 50060482d52dd587 remove dev 667
# symmask –sid 0528 –dir 4a –p 1 –wwn 50060482d52dd587 remove dev 688
# symmask –sid 0528 –dir 4a –p 1 –wwn 50060482d52dd587 remove dev 690

Devices clean up
Devices in old arrays had been removed from HBA masking in the migration operation.
We need to unmap these LUNs from FAs on the old array.

# cat camktest.unmap
Unmap dev 667 from dir 13a:1;
Unmap dev 688 from dir 13a:1;
Unmap dev 690 from dir 13a:1;
Unmap dev 667 from dir 4a:1;
Unmap dev 688 from dir 4a:1;
Unmap dev 690 from dir 4a:1;

# symconfigure -sid 1334 -file camktest.unmap -v -nop preview
# symconfigure -sid 1334 -file camktest.unmap -v -nop prepare
# symconfigure -sid 1334 -file camktest.unmap -v -nop commit

EDS migration tool and scripts

EDS San Support Software
The EDS storage team developed a storage report website. It provided all the necessary
details on current storage/host configuration and provided all the information on the old
array for migration.

The report for server camktest follows on the next page:
Migration scripts:
It is better to make shell scripts for each step to do the huge SYMCLI work of arrays during migration phases, especially as we faced more than 120 servers, nearly 1900 LUNs and about 50TB data in this migration.

Two examples of scripts are below:

- This script is for step FA masking:
  ```bash
  #!/bin/sh
  Host=$1
  cat FALunMasking-$Host.cfg | while read Line
  do
    set $Line
    Dir=$1
    Port=$2
    WWN=$3
    Array=$4
    LunNumber=0
    for Lun in `cat Volume-$Host`
    do
      LunNumber=`expr $LunNumber + 1`
      echo "Assign $Array $Dir $Port $Lun to $WWN"
      symmask -sid $Array -nop -dir $Dir -p $Port -wnn $WWN add dev $Lun
    done
    echo Totally $LunNumber have been added into $WWN
  done
  symmask -sid $Array refresh -nop
  ```
This script is for cleaning LUNs FA masking or undeleted host masking on old array after migration.

```bash
#!/bin/sh

echo # ./Lun_Clean.sh LUNFILE OLD_ARRAY_SID

File=$1
Sid=$2

for lun in `cat $File`
do
    LineNum=0
    symmaskdb list assignment -dev $lun -sid $Sid | grep FIBRE | while read Line
        do
            echo $Line
            LineNum=`expr $LineNum + 1`
            set $Line
            if [ "$LineNum" != 1 ]
    then
        WWN=$1
        FAs=$3
    else
        WWN=$2
        FAs=$4
    fi

    FAPorts=`echo $FAs | sed -e 's/FA-//' -e '/g' -e 's/;/ /g'`
    for FA in `echo $FAPorts`
do
        FA1=`echo $FA | sed 's/:/ /g'`
        set $FA1
        DIR=$1
        PORT=$2
        echo Remove Lun=$lun from dir $DIR port $PORT on $WWN at $Sid
        symmask -sid $Sid -dir $DIR -p $PORT -wwn $WWN remove devs $lun
        sleep 1
    done
done
```