

OpenSource/Linux技術文書

LifeKeeper for Linux v8.1.1  
インストール・シヨンガイド  
Red Hat Enterprise Linux 6.2  
+ HP 3PAR StoreServ 7200 Storage FC  
マルチパス編（第1版）

日本ヒューレット・パッカード株式会社  
2013年3月22日

## 目次

[本ドキュメントについて].....	4
1. 環境 .....	5
2. 設定手順 .....	5
2-1. ノード1 とノード2 に Red Hat Enterprise Linux 6.2 をインストール.....	6
2-2. 3PAR StoreServ 7200 Storage の設定 .....	7
2-3. SPP (Service Pack for ProLiant) の適用.....	10
2-4. Device Mapper Multipath のインストール・設定 .....	11
2-5. ネットワークの確立.....	14
2-6. /etc/fstab ファイルの編集.....	14
2-7. LifeKeeper ソフトウェアのインストール .....	15
2-8. License Key のインストール .....	16
2-9. LifeKeeper の起動 .....	17
2-10. LifeKeeper GUI の起動.....	17
2-11. クラスタ設定、Volume (3PAR StoreServ 7200 Storage) リソース設定.....	18
3. 注意事項 .....	18

## 図表目次

図 1. Create Host の Welcome 画面 .....	7
図 2. Create Host の Host Settings 画面 .....	7
図 3. Create Host の Fibre Channel Settings 画面 .....	8
図 4. Create Virtual Volume 画面 .....	8
図 5. Create CPG 画面 .....	9
図 6. Export Virtual Volume の Virtual Volume 画面 .....	9
図 7. Export Virtual Volume の Virtual Volume Set 画面 .....	10

[本ドキュメントについて]

- 本ドキュメントの内容については充分チェックをしておりますが、その正確性を保証するものではありません。また、将来、予告なしに変更することがあります。
- 本ドキュメントの使用で生じるいかなる結果も利用者の責任となります。日本ヒューレット・パカード株式会社は、本ドキュメントの内容に一切の責任を負いません。
- 本ドキュメントの技術情報は、ハードウェア構成、OS、アプリケーションなど使用環境により大幅に数値が変化する場合がありますので、十分なテストを個別に実施されることを強くお勧め致します。
- 本ドキュメント内で表示・記載されている会社名・サービス名・商品名等は各社の商標又は登録商標です。
- 本ドキュメントで提供する資料は、日本の著作権法、条約及び他国の著作権法にいう著作権により保護されています。

本ドキュメントは、共有FCストレージにマルチパス構成の3PAR StoreServ 7200 Storageを使用したProLiantサーバーにRed Hat Enterprise Linux 6.2とLifeKeeper for Linux v8.1.1をインストールするためのガイドです。

注) 当資料が対象とするバージョン以外のOS、Device Mapper Multipath、LifeKeeper等をご使用の場合は、インストールや設定の手順が異なる場合があります。その場合は、ご使用のバージョンのマニュアルやRelease Notes等に記載された手順に従ってください。

## 1.環境

実際に使用した環境は、以下になります。

### H/W環境

サーバー : HP ProLiant BL460c Gen8

CPU : Xeon E5-2680 2.70GHz × 2

メモリ : 64GB

RAIDコントローラ : SmartアレイP220i/512MB FBWC コントローラ

ファイバチャネルアダプタ : HP 8Gb QMH2572 ファイバーチャネル アダプタ

ストレージシステム : HP 3PAR StoreServ 7200 Storage OS 3.1.2

ファイバチャネル集線装置 : HP SAN スイッチ 8/24 Base

### S/W環境

O/S : Red Hat Enterprise Linux 6.2 kernel-2.6.32-220.el6.x86\_64

クラスターソフトウェア : LifeKeeper for Linux v8.1.1

FCデバイスドライバ : qla2xxxドライバ v8.03.07.05.06.2-k

トポロジー : Fabric接続

なお、今回は2台のサーバーでクラスターを構成しましたが、各ノードとも、環境は同一です。以下、2台のノードをノード1、ノード2とします。

## 2.設定手順

実際の手順の流れは以下になります。

- 1) ノード1とノード2にRed Hat Enterprise Linux 6.2をインストール
- 2) 3PAR StoreServ 7200 Storageの設定
- 3) SPP (Service Pack for ProLiant) の適用
- 4) Device Mapper Multipathのインストール・設定
- 5) ネットワークの確立
- 6) /etc/fstabファイルの編集
- 7) LifeKeeperソフトウェアのインストール
- 8) License Keyのインストール
- 9) LifeKeeperの起動
- 10) LifeKeeper GUIの起動
- 11) クラスター設定、Volume (3PAR StoreServ 7200 Storage) リソース設定

各項目の具体的な作業内容を以降に記述します。

2-1. ノード 1 と ノード 2 に Red Hat Enterprise Linux 6.2 をインストール  
ノード1とノード2にRed Hat Enterprise Linux 6.2をインストールしてください。この時、  
以下のパッケージをインストールしてください。

```
device-mapper-multipath-0.4.9-46.el6.x86_64.rpm  
device-mapper-multipath-libs-0.4.9-46.el6.x86_64.rpm  
compat-libstdc++-296-2.96-144.el6.i686.rpm  
compat-libstdc++-33-3.2.3-69.el6.i686.rpm  
libgcc-4.4.6-3.el6.x86_64.rpm  
nss-softokn-freebl-3.12.9-11.el6.i686.rpm  
glibc-2.12-1.47.el6.i686.rpm  
libXau-1.0.5-1.el6.i686.rpm  
libxcb-1.5-1.el6.i686.rpm  
libX11-1.3-2.el6.i686.rpm  
libXext-1.1-3.el6.i686.rpm  
libXi-1.3-3.el6.i686.rpm  
libXtst-1.0.99.2-3.el6.i686.rpm  
zlib-1.2.3-27.el6.i686.rpm  
bzip2-libs-1.0.5-7.el6_0.i686.rpm  
pam-1.1.1-10.el6.i686.rpm  
libselinux-2.0.94-5.2.el6.i686.rpm  
cracklib-2.8.16-4.el6.i686.rpm  
audit-libs-2.1.3-3.el6.i686.rpm  
db4-4.7.25-16.el6.i686.rpm  
sg3_utils-1.28-4.el6.x86_64.rpm  
udev-147-2.40.el6.x86_64.rpm  
sysfsutils-2.1.0-6.1.el6.x86_64.rpm
```

## 2-2.3PAR StoreServ 7200 Storage の設定

3PAR Management Consoleを使って、3PAR StoreServ 7200 Storageの設定を行います。詳細については、「HP 3PAR Management Console User Guide」を参照して下さい。

A) ホストの作成（ホストの設定とファイバーチャネルの設定）を行います。

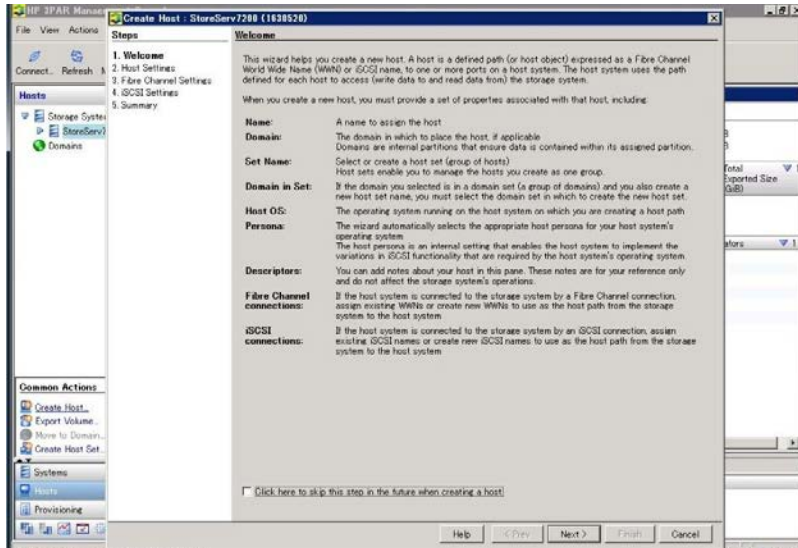


図 1. Create Host の Welcome 画面

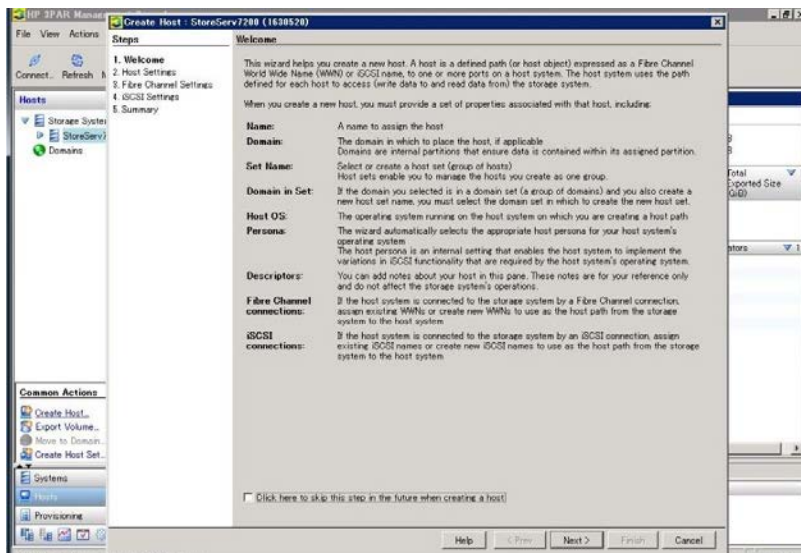


図 2. Create Host の Host Settings 画面

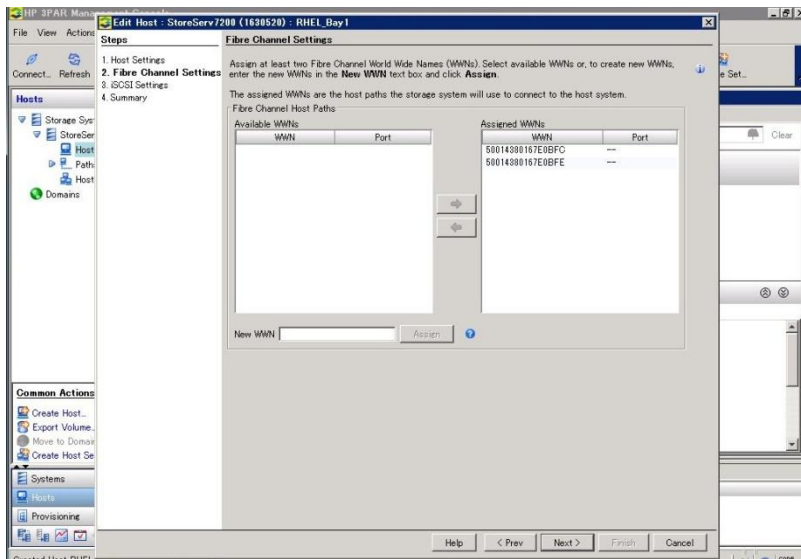


図 3. Create Host の Fibre Channel Settings 画面

B) 仮想ボリュームの作成を行います。

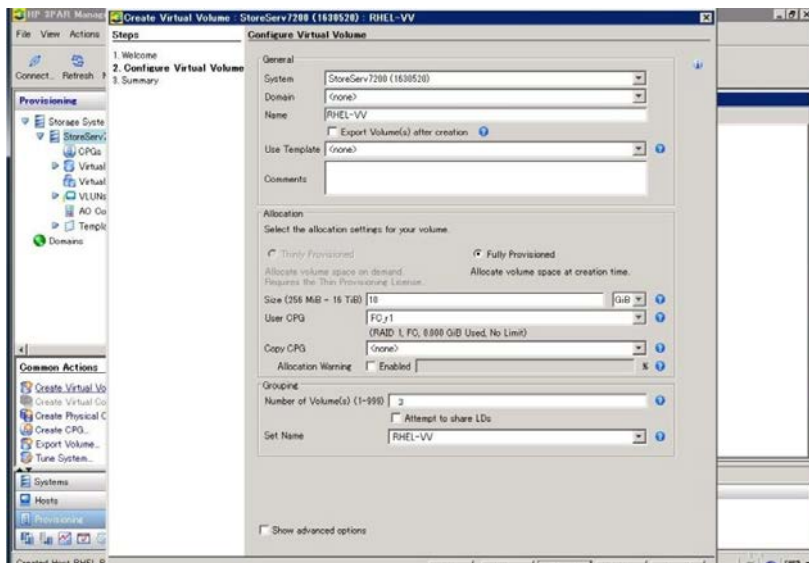


図 4. Create Virtual Volume 画面



C) 共通プロビジョニンググループ (CPG) を作成します。

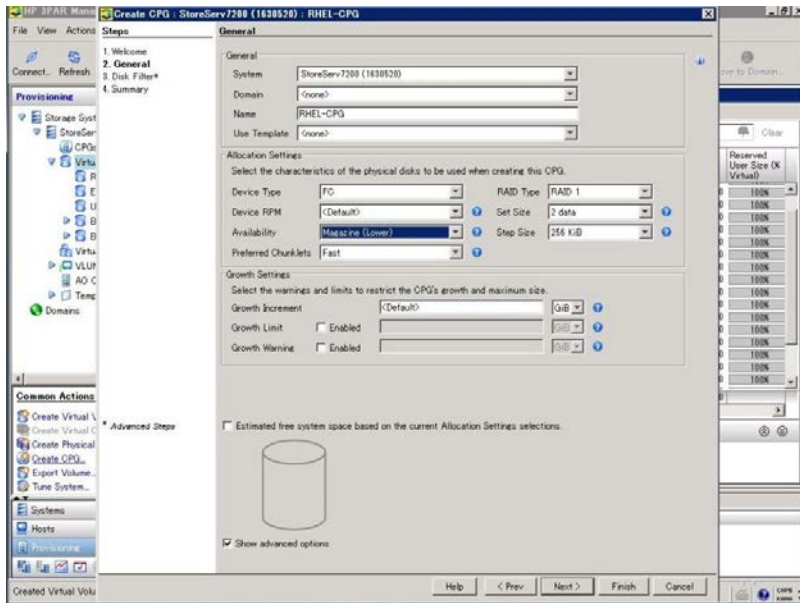


図 5. Create CPG 画面

D) エクスポートする仮想ボリュームを選択します。

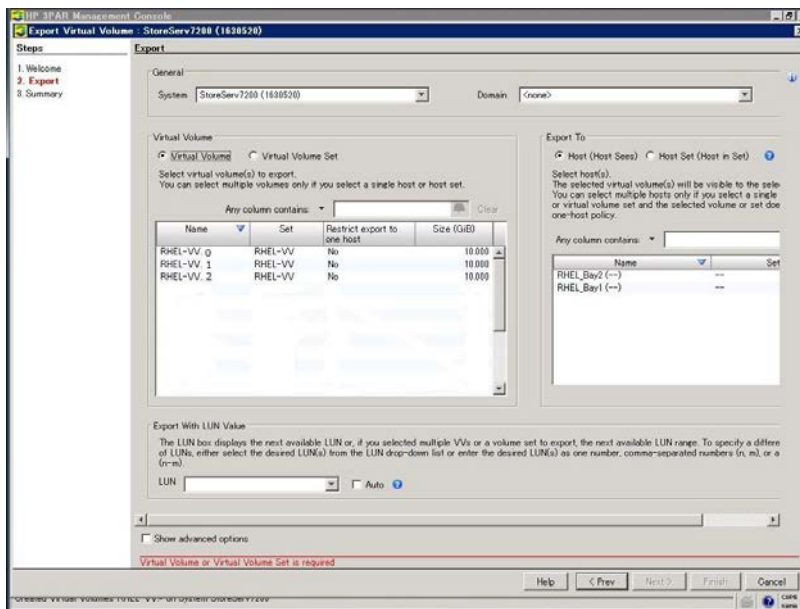


図 6. Export Virtual Volume の Virtual Volume 画面

E) エクスポートする仮想ボリュームセットを選択します。

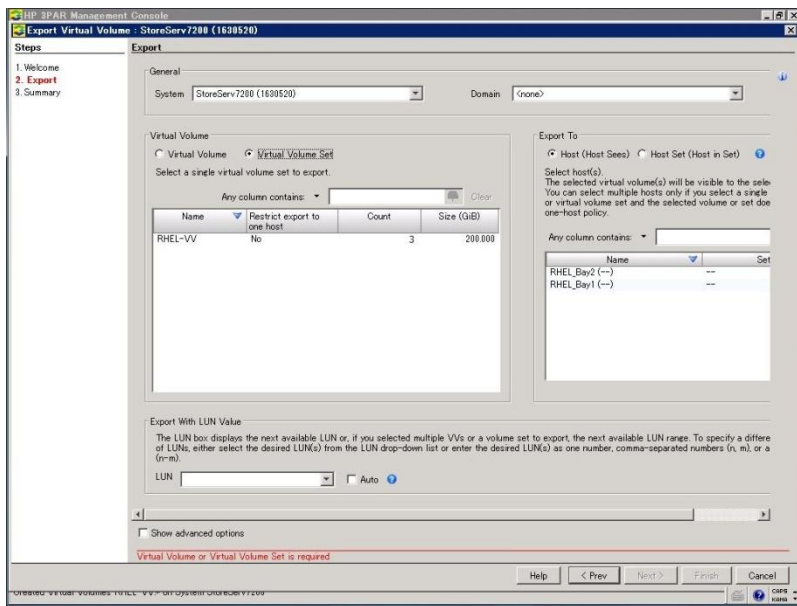


図 7. Export Virtual Volume の Virtual Volume Set 画面

### 2-3.SPP (Service Pack for ProLiant) の適用

以下のサイトを参考にして、各ノードにSPPを適用してください。この時、hp-fc-enablementキットをインストールしてください。なお、SPPに含まれているドライバよりも新しいバージョンのドライバが個別に提供されている場合は、新しいバージョンのドライバを適用することを推奨します。

<https://www.hpe.com/jp/ja/servers/linux/technical.html#spp>

SPP適用後、O/Sをrebootします。

```
# shutdown -r now
```

O/Sが起動されたら、hp-fc-enablementキットを使って、qla2xxxドライバのパラメータを変更します。

以下を実行します。

A) ドライバのパラメータ変更

```
# /opt/hp/hp-fc-enablement/set_parm -m
```

B) Initial RAM Diskの更新

```
# /opt/hp/hp-fc-enablement/make_initrd
```

C) O/Sのreboot

```
# shutdown -r now
```

## 2-4.Device Mapper Multipath のインストール・設定

Device Mapper Multipathのインストール・設定を行います。以下を実行してください。

- A) ノード1にrootでlogin
- B) device-mapper-multipathとdevice-mapper-multipath-libsの2つがインストールされているか確認して、入っていないければインストールしてください。

確認:

```
# rpm -qa | grep multipath
device-mapper-multipath-0.4.9-46.el6.x86_64
device-mapper-multipath-libs-0.4.9-46.el6.x86_64
```

インストール:

```
# rpm -ivh device-mapper-multipath-0.4.9-46.el6.x86_64.rpm
# rpm -ivh device-mapper-multipath-libs-0.4.9-46.el6.x86_64.rpm
```

- C) mpathconfユーティリティを使って、マルチパスを設定します。  
# mpathconf --enable
- D) /etc/multipath.confファイルが作成されるので、そのファイルの「devices」セクション内に、以下の様に3PAR用設定を追加します。「HP 3PAR RedHat and Oracle Linux Implementation Guide」に掲載されている設定に従ってください。

```
defaults {
    user_friendly_names    yes
    polling_interval      10
    max_fds                8192
}

devices {
    device {
        vendor              "3PARdata"
        product             "VV"
        no_path_retry       18
        features            "0"
        hardware_handler    "0"
        path_grouping_policy multibus
        getuid_callout      "/lib/udev/scsi_id --whitelisted --device=/dev/%n"
        path_selector       "round-robin 0"
        rr_weight           uniform
        rr_min_io           100
        path_checker        tur
        failback            immediate
    }
}
```

- E) multipathdデーモンを起動します  
# service multipathd start

- F) 3PAR StoreServ 7200 Storageに作成したLUN (volume) に対応したmultipath device が、/dev/mapper下に生成されている事を確認します。下記はLUNを3個作成した場合の例です。今回の環境では、multipath device名に、システム定義による user\_friendly\_name (mpatha、mpathb、mpathc・・・mpath + アルファベットの形式) を使用しています。

```
# ll /dev/mapper/
total 0
lrwxrwxrwx 1 root root      7 Jan 29 19:51 VolGroup-lv_home -> ../dm-5
lrwxrwxrwx 1 root root      7 Jan 29 19:51 VolGroup-lv_root  -> ../dm-0
lrwxrwxrwx 1 root root      7 Jan 29 19:51 VolGroup-lv_swap -> ../dm-1
crw-rw---- 1 root root 10, 58 Jan 29 19:51 control
lrwxrwxrwx 1 root root      7 Jan 29 19:51 mpathb -> ../dm-2           ← LUN #1
lrwxrwxrwx 1 root root      7 Jan 29 19:51 mpathc -> ../dm-3           ← LUN #2
lrwxrwxrwx 1 root root      7 Jan 29 19:51 mpathd -> ../dm-4           ← LUN #3
```

- G) 各multipath device (/dev/mapper/mpathb等) 毎に、4本のpath、即ち4個のblock device (/dev/sdb等) が構成されている事を確認します。なお、ゾーニング構成によって、1つのmultipath deviceを構成するpathの数は変わります。

```
# multipath -ll
mpathd (360002ac0000000000000000400007738) dm-4 3PARdata, VV           ← LUN #3
size=10G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
  |-- 5:0:0:2 sdd 8:48  active ready running           ← path
  |-- 5:0:1:2 sdg 8:96  active ready running           ← path
  |-- 6:0:0:2 sdj 8:144 active ready running           ← path
  `-- 6:0:1:2 sdm 8:192 active ready running           ← path
mpathc (360002ac0000000000000000300007738) dm-3 3PARdata, VV           ← LUN #2
size=10G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
  |-- 5:0:0:1 sdc 8:32  active ready running
  |-- 5:0:1:1 sdf 8:80  active ready running
  |-- 6:0:0:1 sdi 8:128 active ready running
  `-- 6:0:1:1 sdl 8:176 active ready running
mpathb (360002ac0000000000000000200007738) dm-2 3PARdata, VV           ← LUN #1
size=10G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
  |-- 5:0:0:0 sdb 8:16  active ready running
  |-- 5:0:1:0 sde 8:64  active ready running
  |-- 6:0:0:0 sdh 8:112 active ready running
  `-- 6:0:1:0 sdk 8:160 active ready running
```

- H) multipath deviceを構成しているblock deviceの1つにパーティションを作成するので、まずは、block deviceを確認します。

```
# multipath -l mpathb
mpathb (360002ac0000000000000000200007738) dm-2 3PARdata, VV
size=10G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=0 status=active
  |-- 5:0:0:0 sdb 8:16  active undef running
  |-- 5:0:1:0 sde 8:64  active undef running
  |-- 6:0:0:0 sdh 8:112 active undef running
  `-- 6:0:1:0 sdk 8:160 active undef running
```

- I) 確認したblock deviceとmultipath deviceに対してパーティションを作成します。
- ```
# fdisk -c -u /dev/sdb
# kpartx -a -p p /dev/mapper/mpathb
```
- J) 残りのmultipath device (mpathc、mpathd) についても、構成するblock deviceを確認して (multipath -l)、パーティションを作成します (fdisk -c -uとkpartx -a -p)。
- K) パーティションが正常に作成されているか確認します。
- ```
# ll /dev/mapper/
total 0
lrwxrwxrwx 1 root root      7 Jan 29 19:51 VolGroup-lv_home -> ../dm-5
lrwxrwxrwx 1 root root      7 Jan 29 19:51 VolGroup-lv_root -> ../dm-0
lrwxrwxrwx 1 root root      7 Jan 29 19:51 VolGroup-lv_swap -> ../dm-1
crw-rw---- 1 root root 10, 58 Jan 29 19:51 control
lrwxrwxrwx 1 root root      7 Jan 29 19:51 mpathb -> ../dm-2
lrwxrwxrwx 1 root root     11 Jan 29 20:11 mpathbp1 -> ../dm-6
lrwxrwxrwx 1 root root      7 Jan 29 19:51 mpathc -> ../dm-3
lrwxrwxrwx 1 root root     15 Jan 29 20:15 mpathcp1 -> ../dm-7
lrwxrwxrwx 1 root root      7 Jan 29 19:51 mpathd -> ../dm-4
lrwxrwxrwx 1 root root     17 Jan 29 20:17 mpathdp1 -> ../dm-8
```
- L) 該当デバイスに対してファイルシステムを作成します。下記はext4ファイルシステムを作成する例です。
- ```
# mkfs.ext4 /dev/mapper/mpathbp1
# mkfs.ext4 /dev/mapper/mpathcp1
# mkfs.ext4 /dev/mapper/mpathdp1
```
- M) マウントポイントを作成します。
- ```
# mkdir /mnt1 /mnt2 /mnt3
```
- N) ファイルシステムをmountし、正常にマウント出来る事を確認します。
- ```
# mount /dev/mapper/mpathbp1 /mnt1
# mount /dev/mapper/mpathcp1 /mnt2
# mount /dev/mapper/mpathdp1 /mnt3
```
- O) 正常にファイルを作成できる事を確認します。
- P) 動作確認終了後はアンマウントしておきます。
- ```
# umount /mnt1
# umount /mnt2
# umount /mnt3
```
- Q) ノード2にrootでログイン
- R) device-mapper-multipathとdevice-mapper-multipath-libsの2つがインストールされているか確認して、入っていない場合はインストールしてください。
- 確認:
- ```
# rpm -qa | grep multipath
device-mapper-multipath-0.4.9-46.el6.x86_64
device-mapper-multipath-libs-0.4.9-46.el6.x86_64
```
- インストール:
- ```
# rpm -ivh device-mapper-multipath-0.4.9-46.el6.x86_64.rpm
# rpm -ivh device-mapper-multipath-libs-0.4.9-46.el6.x86_64.rpm
```
- S) ノード1の/etc/multipath.confファイルをノード2にコピーします。以下はscpコマンドでコピーする例です。
- ```
# scp ノード1のホスト名:/etc/multipath.conf /etc
```

- T) multipathdデーモンを起動します  
# service multipathd start
- U) システム起動時、multipathdデーモンが自動起動されるように設定します。  
# chkconfig multipathd on
- V) multipath deviceとパーティションが正しく認識されているか確認します。  
# ll /dev/mapper/  
total 0  
lrwxrwxrwx 1 root root 7 Jan 29 19:51 VolGroup-lv\_home -> ../dm-5  
lrwxrwxrwx 1 root root 7 Jan 29 19:51 VolGroup-lv\_root -> ../dm-0  
lrwxrwxrwx 1 root root 7 Jan 29 19:51 VolGroup-lv\_swap -> ../dm-1  
crw-rw---- 1 root root 10, 58 Jan 29 19:51 control  
lrwxrwxrwx 1 root root 7 Jan 29 20:23 mpathb -> ../dm-2  
lrwxrwxrwx 1 root root 7 Jan 29 20:23 mpathbp1 -> ../dm-6  
lrwxrwxrwx 1 root root 7 Jan 29 20:23 mpathc -> ../dm-4  
lrwxrwxrwx 1 root root 7 Jan 29 20:23 mpathcp1 -> ../dm-8  
lrwxrwxrwx 1 root root 7 Jan 29 20:23 mpathd -> ../dm-3  
lrwxrwxrwx 1 root root 7 Jan 29 20:23 mpathdp1 -> ../dm-7
- W) マウントポイントを作成します。マウントポイント名は、必ずノード1と同一にしてください。  
# mkdir /mnt1 /mnt2 /mnt3
- X) ファイルシステムをmountし、正常にマウント出来る事を確認します。  
# mount /dev/mapper/mpathbp1 /mnt1  
# mount /dev/mapper/mpathcp1 /mnt2  
# mount /dev/mapper/mpathdp1 /mnt3
- Y) 先ほど、ノード1で作成したファイルが正常に見える事を確認します。
- Z) 動作確認終了後はアンマウントします。  
# umount /mnt1  
# umount /mnt2  
# umount /mnt3
- AA) 再度、ノード1（プライマリサーバー）からファイルシステムをmountします。  
# mount /dev/mapper/mpathbp1 /mnt1  
# mount /dev/mapper/mpathcp1 /mnt2  
# mount /dev/mapper/mpathdp1 /mnt3
- BB) 両ノードから、ファイルシステムがマウント可能で、アクセス可能である事が確認できたならば、ファイルシステムをマウントするのは、ノード1（プライマリサーバー）だけにしてください。

## 2-5.ネットワークの確立

両ノードのセットアップが完了したら、それぞれのノードに対して、pingが可能か確認してください。また、それぞれのノードで、DNSサービスもしくは/etc/hostsを使用して名前解決できるようにしてください。

## 2-6./etc/fstab ファイルの編集

/etc/fstabファイルにラベル名を使用している場合は、ブロックデバイス名に変更します。LifeKeeperでは/etc/fstab内では、ラベル名ではなく、ブロックデバイス名を使用することが推奨されています。

## 2-7.LifeKeeper ソフトウェアのインストール

LifeKeeper for Linux v8.1.1 をインストールするために、両ノード上で、以下のことを実行してください。

- A) 製品の配布メディアを任意のディレクトリに mount し、続いて sps.img ファイルを任意のディレクトリに mount します。  
# mount /dev/cdrom /media/cdrom  
# mount /media/cdrom/sps.img /work -t iso9660 -o loop
- B) sps.img を mount したディレクトリに移動します。
- C) ./setup の実行
- D) 画面に表示される質問に、応答してください。
- E) 途中、以下のように unique host ID が表示されます。この ID は、後で License Key を取得する時に必要になるので、正確に記録しておいてください。

The unique host ID for this system is listed below.

00XX7856XXX0

- F) 「Would you like to install a license key now? (y/n) [n] ?」と質問されます。License Key は後でインストールするので、ここではリターン（もしくは n で応答）。
- G) Select optional kits for SPS for Linux 画面が表示されるので、「lkDMMP LifeKeeper Device Mapper (DM) multipath Recovery Kit」を選択して、インストールしてください。また、他に必要なオプションの Recovery Kit があれば、選択して、インストールしてください。
- H) 「Setup has completed successfully.」と表示されます。

- I) LifeKeeper for Linux v8.1.1 では、デフォルトで以下のパッケージがインストールされます。

```
# rpm -qa | grep steel | sort
steeleye-curl-7.21.7-3.i386
steeleye-gnutls-2.8.6-3.i386
steeleye-gnutls-utils-2.8.6-3.i386
steeleye-libcurl-7.21.7-3.i386
steeleye-libcrypt-1.5.0-2.i386
steeleye-libgpg-error-1.10-2.i386
steeleye-libxml2-2.7.8-7.i386
steeleye-libxml2-static-2.7.8-7.i386
steeleye-lighttpd-1.4.26-2.8.i386
steeleye-lighttpd-fastcgi-1.4.26-2.8.i386
steeleye-lk-8.1.1-5620.i386
steeleye-lkCCISS-8.1.1-5620.i386
steeleye-lkGUI-8.1.1-5620.i386
steeleye-lkIP-8.1.1-5620.noarch
steeleye-lkLIC-8.1.1-5620.i386
steeleye-lkMAN-8.1.1-5620.noarch
steeleye-lkRAW-8.1.1-5620.noarch
steeleye-lkRHAS-8.1.1-5620.noarch
steeleye-lkapi-8.1.1-5620.i386
steeleye-lkapi-client-8.1.1-5620.i386
steeleye-openssl-0.9.7a-43.3.i386
steeleye-openssl-perl-0.9.7a-43.3.i386
steeleye-pcre-4.5-2.i386
steeleye-pdksh-5.2.14-780.7.i386
steeleye-perl-addons-5.8.8-18.i386
steeleye-perl32-5.8.8-7.i386
steeleye-readline-4.3-14.i386
steeleye-runit-2.0.0-4.4.i386
```

- J) 環境変数 PATH および MANPATH を以下のように設定しておくとう便利です。

```
PATH=$PATH:/opt/LifeKeeper/bin
```

```
MANPATH=$MANPATH:/opt/LifeKeeper/man
```

- K) /var/log/LK\_install.log ファイルにログが書かれます。

- L) /etc/default/LifeKeeper ファイルに次のパラメータを追加します。これは、3PAR StoreServ 用の設定です。

```
DMMP_REGISTRATION_TYPE=hba
```

## 2-8. License Key のインストール

以下の手順で、各ノードごとに License Key をインストールしてください。

- A) 2-7. LifeKeeper ソフトウェアのインストールの手順 E) で表示された unique host ID と LifeKeeper ソフトウェアに同梱されている Entitlement ID (Authorization Code) で、米国サイオステクノロジー社 (<http://us.sios.com> の LICENSE KEY リンク) から各ノードの License Key を入手してください。
- B) 各ノードに root で login



- C) 以下のコマンドを実行  
# /opt/LifeKeeper/bin/lkkeyins
- D) 画面に表示される質問に、適切に応答してください。
- E) 「LifeKeeper license key installation was successful!」と表示されれば、License Key のインストールは成功です。

## 2-9.LifeKeeper の起動

LifeKeeper を起動するために、両ノードで以下のことを実行してください。

- A) LifeKeeper を起動します。  
# /opt/LifeKeeper/bin/lkstart
  - B) lktest コマンドを実行して、LifeKeeper の起動を確認します。  
# /opt/LifeKeeper/bin/lktest
- | F | S | UID  | PID   | PPID | C | CLS | PRI | NI  | SZ   | STIME | TIME     | CMD           |
|---|---|------|-------|------|---|-----|-----|-----|------|-------|----------|---------------|
| 4 | S | root | 52159 | 3571 | 0 | TS  | 39  | -20 | 1458 | 20:28 | 00:00:00 | lcm           |
| 4 | S | root | 52163 | 1    | 0 | TS  | 39  | -20 | 1442 | 20:28 | 00:00:00 | events/lcm -l |
| 4 | S | root | 52166 | 3574 | 0 | TS  | 39  | -20 | 1445 | 20:28 | 00:00:00 | ttymon/lcm    |
| 4 | S | root | 52169 | 3569 | 0 | TS  | 29  | -10 | 1731 | 20:28 | 00:00:00 | lcd           |
- C) LifeKeeper を停止する場合は、以下のコマンドを実行してください。  
# /opt/LifeKeeper/bin/lkstop

## 2-10.LifeKeeper GUI の起動

LifeKeeper GUI を起動するために、両ノードで、以下のことを実行してください。なお、今回の構成では、LifeKeeper GUI のクライアント/サーバーは同一マシンです。

- A) LifeKeeper GUI パッケージがインストールされているか rpm コマンドで確認します。  
# rpm -qa |grep steeleye-lkGUI
- B) 以下のように表示されれば OK。  
steeleye-lkGUI-8.1.1-5620.i386
- C) LifeKeeper GUI サーバーが起動されているか ps コマンドで確認します。  
# ps -ef|grep runG
- D) 以下のように表示されれば OK。  
root 52919 3573 0 20:28 ? 00:00:00 /bin/sh /opt/LifeKeeper/bin/runGuiServer  
root 52920 52919 0 20:28 ? 00:00:00 /opt/LifeKeeper/bin/lklogmsg -p java -l LK\_INFO -s GUI\_Server -t -a -i 999 -c /opt/LifeKeeper/bin/runGuiServer
- E) もし、LifeKeeper GUI サーバーが起動されていなければ、以下のコマンドを実行して起動します。  
# /opt/LifeKeeper/bin/lkGUIserver start
- F) LifeKeeper GUI クライアントを起動します。X Window 上から以下のコマンドを実行。  
# /opt/LifeKeeper/bin/lkGUIapp
- G) Cluster Connect Dialog が表示されます。
- H) LifeKeeper GUI サーバーを停止する場合は、以下のコマンドを実行してください。  
# /opt/LifeKeeper/bin/lkGUIserver stop
- I) 初期インストール後、一度、LifeKeeper GUI サーバーを起動すれば、LifeKeeper の起動/停止に伴い、LifeKeeper GUI サーバーも起動/停止されます。

J) LifeKeeper は、以下のポート番号を使用しているため、この点を考慮して、パケットフィルタリングして下さい。

- 7365 (TCP) : コミュニケーションパス (TCP) の通信用
- 81 (TCP) : GUI サーバーの通信用
- 82 (TCP) : GUI サーバーの通信用
- 1024 (TCP) ~ : GUI サーバー、クライアント間の RMI 通信用

2-11. クラスター設定、Volume (3PAR StoreServ 7200 Storage) リソース設定  
LifeKeeper GUI クライアントからクラスターの設定を行います。以下の作業は、ノード 1 でのみ行います。

A) クラスターの構成

ノード 2 とコミュニケーションパスをはり、クラスターを構築します。

Edit > Server > Create Comm Path から 2 本のコミュニケーションパスをノード 2 とはり、クラスターを構築します。

B) Volume リソースの作成

DMMP 環境でも、DMMP ARK インストール後は、通常の FileSystem リソース作成と同手順で、DMMP の共有ディスクリソースが作成可能です。

Edit > Server > Create Resource Hierarchy を選びます。

Select Recovery Kit で File System を選びます。Mount Point 項目で、マルチパスデバイスがマウントされているディレクトリが表示され、Wizard を進んでいくと、リソースが作成されます。

### 3. 注意事項

A) /etc/fstab ファイルの編集

/etc/fstab ファイルにラベル名を使用している場合は、ブロックデバイス名に変更してください。LifeKeeper では/etc/fstab 内では、ラベル名ではなく、ブロックデバイス名を使用することが推奨されています。

B) LifeKeeper GUI が使用するポート

LifeKeeper は、以下のポート番号を使用しているため、この点を考慮して、パケットフィルタリングして下さい。

- 7365 (TCP) : コミュニケーションパス (TCP) の通信用
- 81 (TCP) : GUI サーバーの通信用
- 82 (TCP) : GUI サーバーの通信用
- 1024 (TCP) ~ : GUI サーバー、クライアント間の RMI 通信用

C) /etc/default/LifeKeeper ファイルにパラメータを追加

/etc/default/LifeKeeper ファイルに次のパラメータを追加してください。これは 3PAR StoreServ 用の設定になります。

```
DMMP_REGISTRATION_TYPE=hba
```

以上