文書番号: OSSTS20111007-001-001

LifeKeeper for Linux と Dell PowerVault MD3200 の連携

SCSI Reservation に依存しないクラスタの構成

第 1 版

サイオステクノロジー株式会社

目次

改版履歴

2011 年 10 月 7 日　　第 1 版

1　本ドキュメントの目的

このたび、LifeKeeper for Linux (以下、LifeKeeper)と Dell 社製ストレージ PowerVault MD3200 (以下、MD3200) の組み合わせにおいて、マルチパス構成がご利用いただけるようになりました。

これまで、当該ストレージはシングルパスで利用した場合のみサポート認定する形態をとっておりました。これは過去の認定テストにおいて、当該ストレージをマルチパスで構成した場合に、従来の LifeKeeper の標準であった SCSI Reservation による I/O Fencing が正しく行えないことが明らかになっていたためです。

LifeKeeper for Linux v7.3 以降のバージョンでは、SCSI Reservation に加えて 2 つの新たな I/O Fencing の方式がサポートされました。Quorum/Witness Server 方式、IPMI STONITH 方式です。

それぞれの方式について理解し、適切に構成することによって、LifeKeeper は RESERVE が有効に機能しないストレージであっても柔軟かつ効果的な I/O Fencing を実現することができます。

本ドキュメントは、弊社で実際に行った検証作業の構成をベースとして記述したものです。LifeKeeper で、マルチパス構成の MD3200 を共有ストレージとして利用した HA システムを構築する手法について、各機能の解説を交えながらご説明します。

2　本ドキュメントのサポート範囲

本ドキュメントは、LifeKeeper で MD3200 をクラスタの共有ストレージとして利用するために必要となる各種 I/O Fencing 機構の構成方法をご案内します。

■ 互換性について

本ドキュメントは MD3200 および Dell PowerEdge R710 サーバを用いた検証結果に基づくものです。以下に挙げるものを除き、他の機器への応用を想定したものではありません。

ストレージ

- **Dell PowerVault MD3200**
- **Dell PowerVault MD3220**

サーバ

- **Dell PowerEgde R710**

3 Quorum / Witness Server 方式の概要

Quorum/Witness Server 方式は、ノード障害を検知した際のフェイルオーバ先を多数決により決定する **Quorum Check** と、第三者サーバに問い合わせて、相手ノードの死活状態を再確認する **Witness Check** 機能によって構成されます。

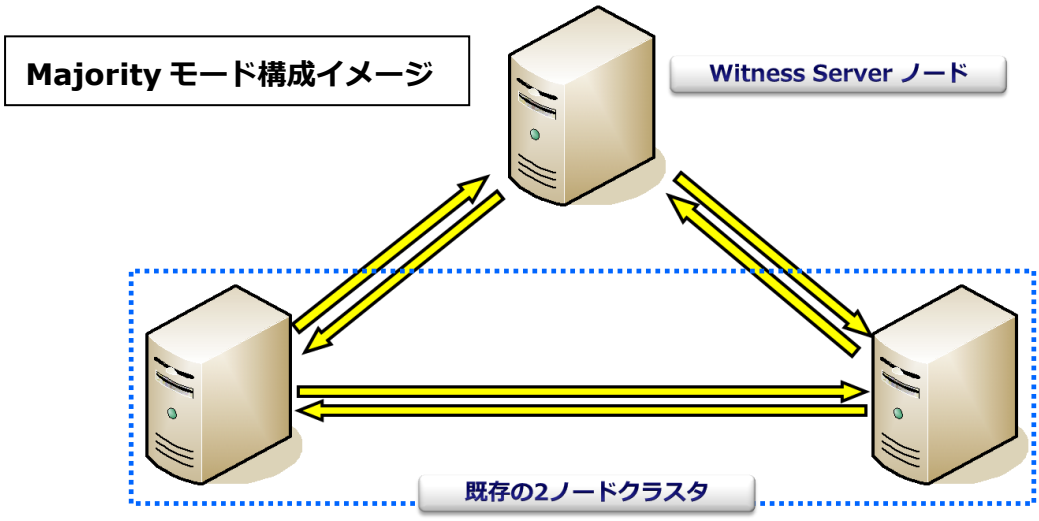
この機能を利用するには、LifeKeeper v7.3 以降で追加された Quorum/Witness Server Kit パッケージ(steeleye-lkQWK)をインストールする必要があります。なお、パッケージは Recovery Kit として製品メディアに同梱されています。

■ Quorum Check のモードについて

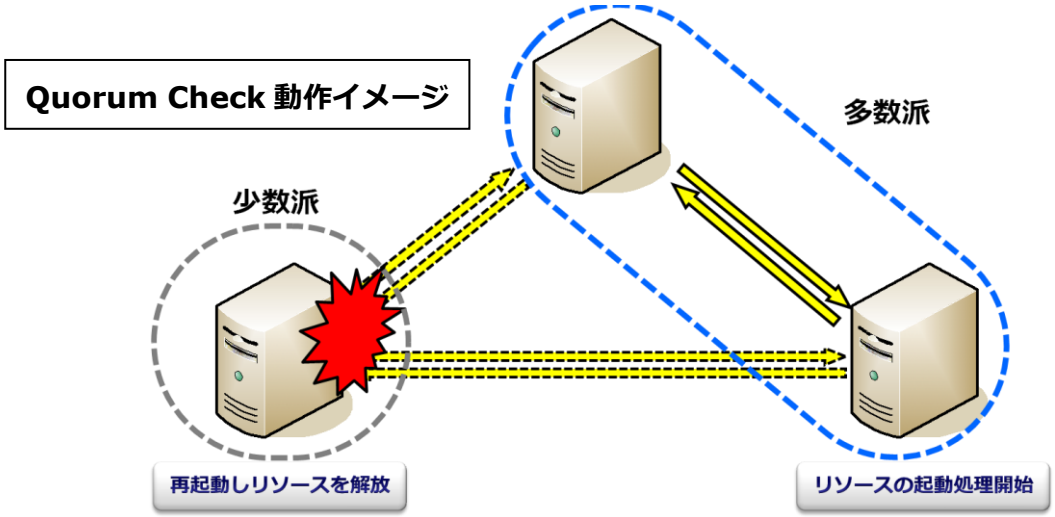
Quorum Check は、ノードの状態確認機能として 2 つのモードが用意されています。どちらのモードを選択するかによって、構成方法が大きく変わります。

3.1 Majority モード

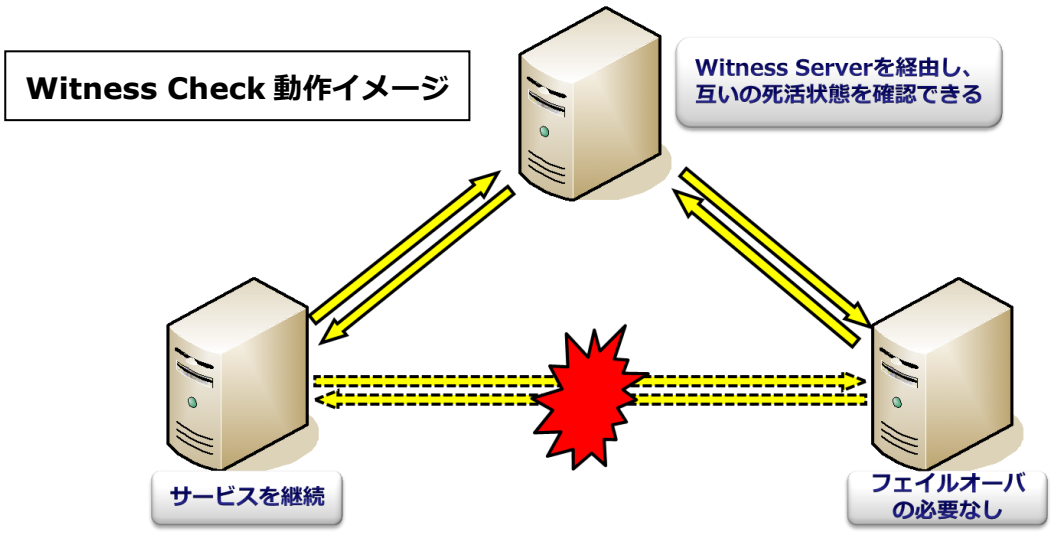
Witness Server と呼ばれる役割を持つ第三のノードをクラスタに参加させることによって、多数決を実現します。3 ノード以上の奇数台構成のクラスタであれば、追加のノードを用意する必要はありません。



Majority モードの Quorum Check では、あるノードの障害を検知した際に、クラスタを構成するノード全体の過半数と疎通可能かどうかをチェックし、自ノードが多数派であることを確認できた場合のみリソースを起動します。リソースを稼働しているノードが少数派となった場合には自ノードの電源を遮断し、スプリットブレインの発生を未然に防ぎます。



その後 Witness Check が開始され、ハートビートが途絶えたノードの死活状態を Witness Server に問い合わせます。この結果、対向ノードでリソースが稼動状態であることを確認できた場合は、フェイルオーバを中止しますので、スプリットブレインは発生しません。



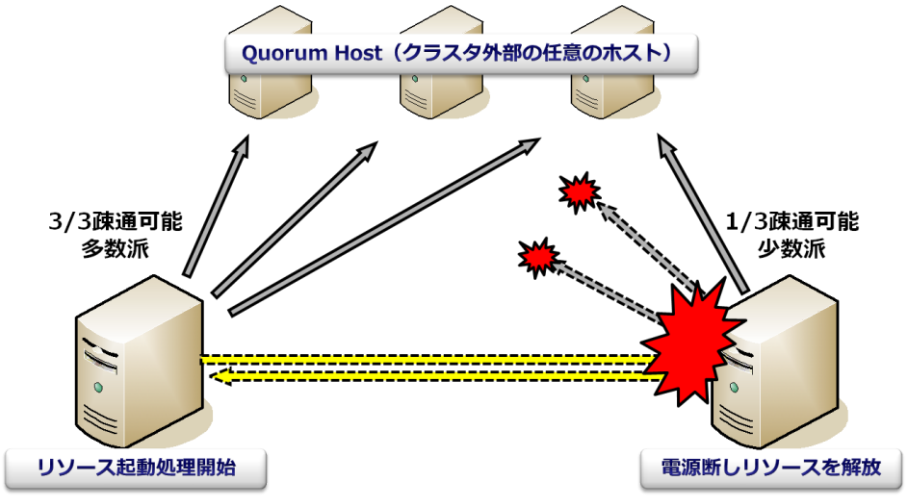
Quorum Check の結果、両ノードが同時に多数派となってしまった場合においても、Witness Check が相互の稼働状況を確認した上でフェイルオーバの可否が決定されるため、スプリットブレインは発生しません。

この方式では、クラスタを構成するノードは Witness Server を含めて奇数である必要がある点と、Witness Server となる第三のノードにも LifeKeeper のライセンスを適用する必要がある点に留意してください。

3.2 TCP Remote モード

Quorum Host に指定したホストのうち過半数に接続できるかどうかで、自ノードが多数派かどうかをチェックします。多数派であるノードのみがリソースを起動することができます。リソースを稼働しているノードが少数派となった場合には自ノードの電源を遮断し、スプリットブレインの発生を未然に防ぎます。

Quorum Check 動作イメージ



ノード間コミュニケーションパスの通信経路のどれか一つが、Quorum Host への通信経路と同じネットワークを利用するよう構成します。このように構成することで、全コミュニケーションパスの切断が発生した場合に、少なくとも一方のノードが少数派となる（スプリットブレインを回避できる）ことが期待できます。

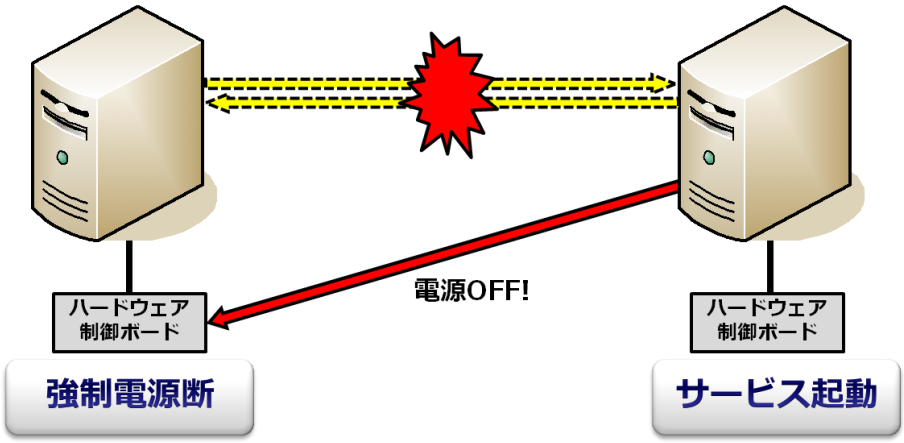
この方式では、Quorum Host の指定したポートへセッションが張れるかどうかで接続確認を行います。したがって、クラスタノードを追加する必要がありません。ノード数による制約がないため、現状の構成を維持したままの導入が可能であることも特徴です。一般的な 2 ノード構成のクラスタにも導入することができます。

しかし、2 ノード構成では第三者サーバを経由した Witness Check の効果が得られず、極稀にスプリットブレインが発生するおそれがあることから、以下に紹介する IPMI STONITH と併用してください。

なお、今回の構成のように、SCSI RESERVATOIN を停止し、Quorum/Witness Server Kit のみで I/O Fencing を行われる場合は、Majority モード、TCP remote モードのいずれの構成であっても、IPMI STONITH の併用が必須となります。

4　IPMI STONITH の概要

IPMI*(Intelligent Platform Management Interface)*は、リモートでサーバのハードウェアを管理する仕組みです。また、**STONITH** は *"Shoot The Other Node In The Head"*の略語であり、他ノードの電源を強制的に遮断する動作を指します。LifeKeeper は、IPMI の仕組みを利用して、フェイルオーバ開始時にフェイルオーバ元ノードの電源を強制的に遮断(STONITH)することで、スプリットブレインの発生を確実に抑止します。



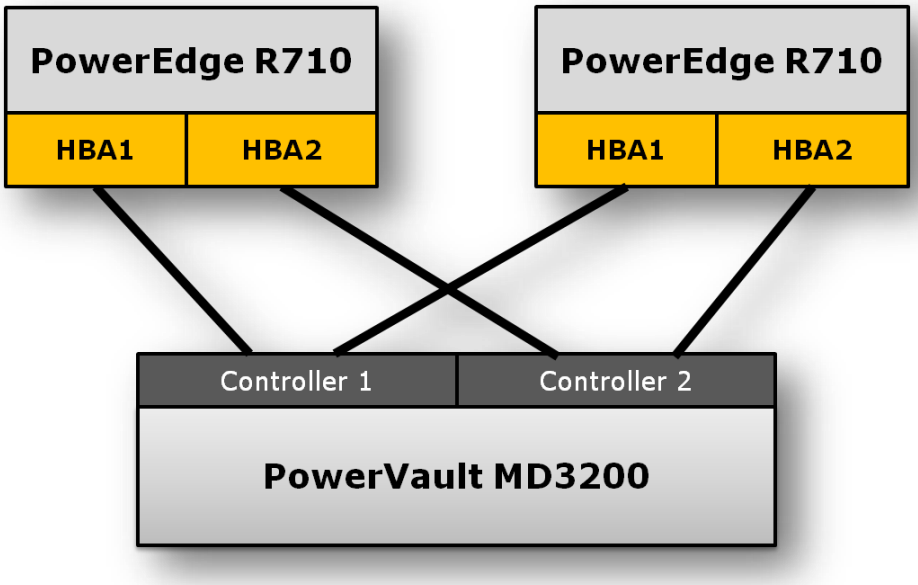
上述の Quorum/Witness Server Kit では、障害ノードが自発的に電源断を行い、スプリットブレインを回避します。しかし、カーネルのハングアップなど、障害ノードが完全に応答を停止してしまった場合は、OS による自発的な電源断は期待できません。そのような場合においても、IPMI STONITH を利用することで、外部から強制的に電源を遮断することができます。この機能を使用することで、スプリットブレインが発生する可能性を排除します。

5　構成方法

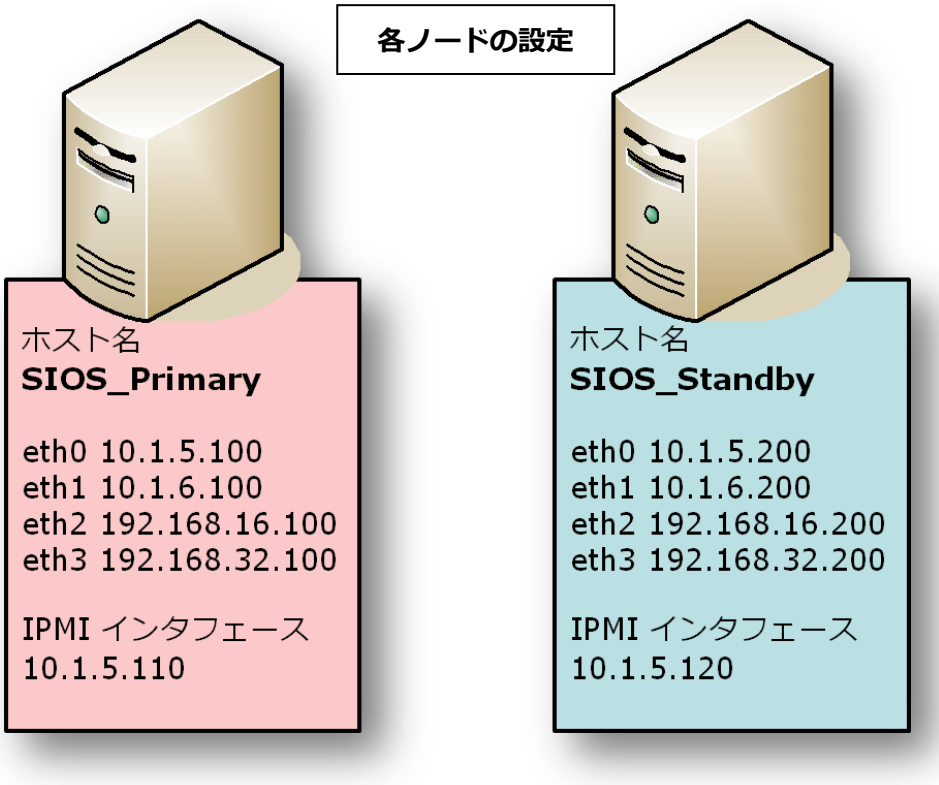
本項では、以下のような 2 ノード構成の Active/Stanby クラスタを構築します。

サーバ	Dell 社 PowerEdge R710
OS	Red Hat Enterprise Linux 5.5 (x86_64)
LifeKeeper	v7.3
共有ストレージ	Dell 社 PowerVault MD3200 (SAS 接続)
マルチパスドライバ	device-mapper multipath, RDAC

各サーバと MD3200 の間は SAS ケーブルで以下のように結線しています。(マルチパス接続)



上記の環境にて Quorum/Witness Server Kit および IPMI STONITH の設定を行い、MD3200 を共有ストレージとして用いたクラスタを構成します。なお、稼動系ノードを SIOS_Primary、待機系ノード名を SIOS_Standby として説明を行います。



5.1 構成の前提

- ✓ 各ノードに Red Hat Enterprise Linux 5.5 がインストールされている
- ✓ 各ノードに LifeKeeper v7.3 および DMMP ARK がインストールされている
- ✓ 各ノードに Quorum/Witness Server Kit がインストールされている
- ✓ 各ノードにドライバがインストールされ、マルチパスで接続した MD3200 上の LU にアクセス可能な状態である（現時点ではマウントを行わないでください）

5.2 SCSI Reservation の無効化

各ノードの LifeKeeper 設定ファイル/etc/default/LifeKeeper に以下のパラメータを追記し、SCSI Reservation によるストレージ制御を無効にします。

```
RESERVATIONS=none
```

この変更を有効にするために、LifeKeeper を再起動してください。

```
# /opt/LifeKeeper/bin/lkstop
# /opt/LifeKeeper/bin/lkstart
```

5.3 IPMI STONITH の設定

(1) ipmitool をインストールし、各ノードで相互に IPMI デバイスと通信できるよう設定を行います。

コマンド実行例

```
# ipmitool lan set 1 ipsrc static
# ipmitool lan set 1 ipaddr 10.1.5.110
# ipmitool lan set 1 netmask 255.255.0.0
# ipmitool lan set 1 defgw ipaddr 10.1.1.10
# ipmitool user set name 1 lifekeeper
# ipmitool user set password 1 secret
# ipmitool user priv 1 4
# ipmitool user enable 1
```

※ ipmitool の設定についての詳細は以下を参照してください。
http://ipmitool.sourceforge.net/manpage.html

(2) ipmitool を利用し、シェルコマンドで互いにノードの電源断(STONITH)を行えることを確認します。

(3) stonith-install スクリプトを実行し、LifeKeeper が各ノードに対して IPMI STONITH を実行するための仕組みをインストールします。

```
# /opt/LifeKeeper/samples/STONITH/stonith-install
```

(4) stonith.conf に、各ノードの電源断を実行する IPMI STONITH コマンドを記述します。各ノードからのフェイルオーバ処理を実施する前に、ここに記述した IPMI STONITH コマンドが実行されます。**フェイルオーバ元のノード名**と、**そのノードに対して実行する IPMI STONITH コマンド**をスペースで区切って記述します。

今回の例では、SIOS_Primary の stonith.conf には SIOS_Standby の電源断を実行するコマンド、SIOS_Standby の stonith.conf には SIOS_Primary の電源断を実行するコマンドをそれぞれ記述します。

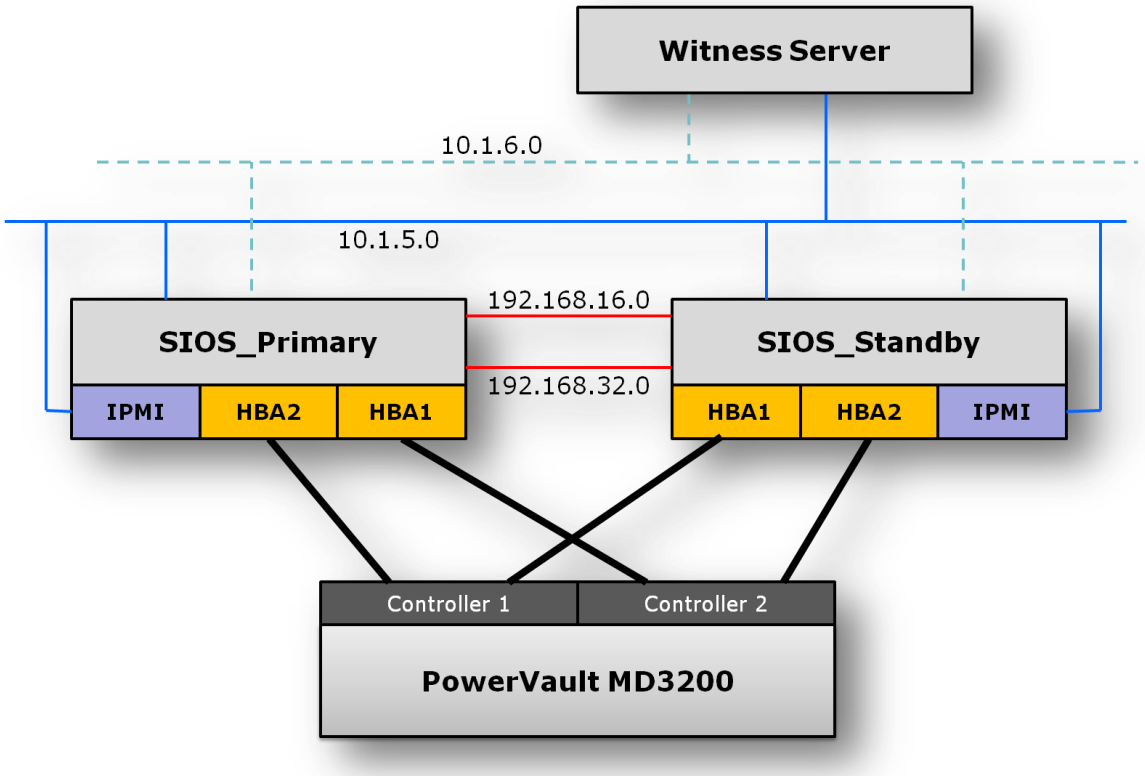
```
# vi /opt/LifeKeeper/config/stonith.conf

(SIOS_Primary の stonith.conf)
SIOS_Standby ipmitool -I lanplus -H 10.1.120 -U lifekeeper -P secret
chassis power off

(SIOS_Standby の stonith.conf)
SIOS_Primary ipmitool -I lanplus -H 10.1.5.110 -U lifekeeper -P secret
chassis power off
```

注 : 電源断の替わりに再起動するようコマンドを記述することもできます。しかし、再起動後にハートビート通信が復旧していない場合には、再起動したノードが STONITH を発動してしまい、ピンポン状態となる可能性があります。そのため、コマンドは電源断とすることを推奨します。

5.4 Quorum / Witness Server Kit の設定(1)

3 ノード以上の奇数台構成で Majority モードを利用する場合

Witness Server となるノードを新規に用意し、以下のようなネットワークを構成しました。



■ コミュニケーションパスの作成

Witness Server を含む各ノード間で、コミュニケーションパスを作成します。

- SIOS_Primary と SIOS_Standby の間には、192.168.16.0 と 192.168.32.0 ネットワークを利用したコミュニケーションパスを作成します。
- SIOS_Primary、SIOS_Standby の両ノードから、10.1.5.0 と 10.1.6.0 ネットワークを経由した Witness Server とのコミュニケーションパスをそれぞれ作成します。

■ パラメータの設定

/etc/default/LifeKeeper を編集し、パラメータを以下のように設定します。この設定は全てのノードで共通にしてください。

```
# vi /etc/default/LifeKeeper

QUORUM_MODE=majority                    # Quorum Check のモード
WITNESS_MODE=remote_verify              # Witness Check の有無
QUORUM_LOSS_ACTION=fastkill             # 少数派ノードの電源断動作
```

■ ハートビート通信途絶時の動作

A) SIOS_Primary のネットワーク機能が異常停止した場合

ハートビート通信が途絶するシチュエーションはいくつか考えられますが、ここでは SIOS_Primary にノード障害が発生し、ネットワーク機能に問題が生じた場合を想定します。

- ➔ SIOS_Primary は SIOS_Standby と通信できず、また Witness Server とも通信ができません。クラスタを構成する 3 台のノードのうち、自ノード 1 台しか疎通が取れないことから少数派となり、自ノードの電源断を行います。
- ➔ SIOS_Standby は SIOS_Primary との通信はできませんが、10.1.5.0 および 10.1.6.0 ネットワークを経由して Witness Server と通信可能な状態です。自ノードも含め、クラスタノード 3 台中の 2 台と疎通ができるので、Quorum Check の結果、自ノードが多数派であると判断し、サービスの起動を行います。

結論：SIOS_Primary は電源断、SIOS_Standby がリソース起動

B) コミュニケーションパスのネットワークに障害発生した場合

192.168.16.0 および 192.168.32.0 ネットワークのみに障害が発生して、ハートビート通信が途絶したケースです。各ノードに異常はなく、また SIOS_Primary で稼働中のサービスに影響がないものとします。

- ➔ SIOS_Primary と SIOS_Standby は互いに通信できない状態となりますが、どちらも Witness Server とは通信ができます。そのため、Quorum Check の結果、どちらのノードも自ノードが多数派であると判断します。
- ➔ Witness Check を実施します。SIOS_Primary と SIOS_Standby はそれぞれ相手ノードの死活状態を Witness Server に問合せ、相手ノードが実際には稼動していることを知ります。結果として、SIOS_Primary はサービスを継続し、

SIOS_Standby はフェイルオーバ処理を行いません。

結論：SIOS_Primary はサービスを継続、SIOS_Standby はフェイルオーバ処理を行わない

■ IPMI STONITH による電源断が必要となるシチュエーション

A) SIOS_Primary が完全に応答停止状態となった場合

SIOS_Primary の OS が完全に応答停止してしまったケースです。この場合、Quorum Check による SIOS_Primary の自発的な電源断は期待できません。

しかし、SIOS_Standby が Quorum Check の結果多数派となり、フェイルオーバ処理を開始します。フェイルオーバ処理の開始前に IPMI STONITH が発動して SIOS_Primary の電源を強制的に遮断します。

そのため、スプリットブレインが発生することなく、リソースのフェイルオーバを実施することができます。

B) Witness Check を無効にした場合

/etc/default/LifeKeepe の設定を以下のように変更することで、Quorum Check に引き続き行われる Witness Check を無効化することができます。なお、Majority モードを利用する場合に Witness Check を無効化することは**推奨しません**。

```
WITNESS_MODE=none
```

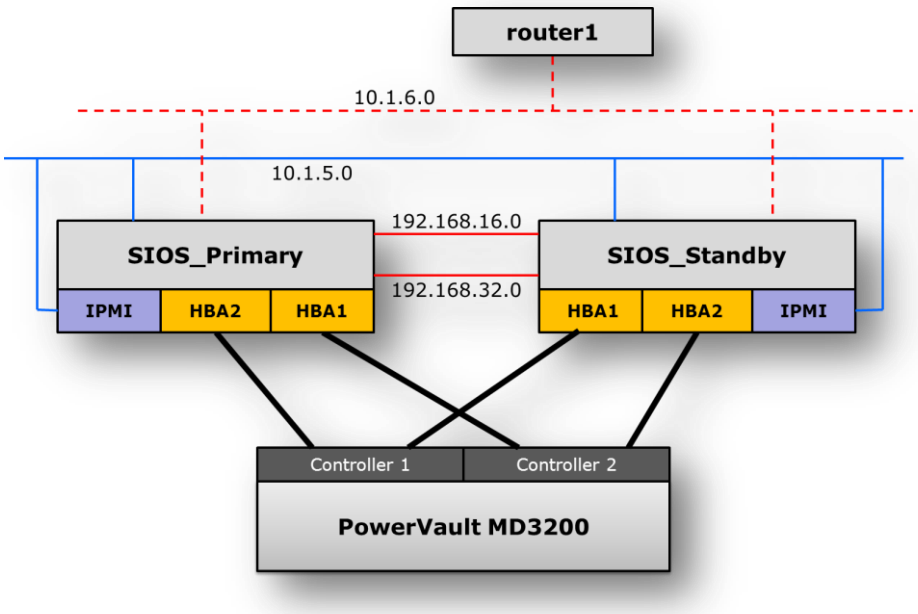
Witness Check を無効にした状態で、192.168.16.0 および 192.168.32.0 ネットワークに障害が発生したケースを想定します。上記「ハートビート通信途絶時の動作 B」とほぼ同じ状況ですが、Witness Check による死活状態の再確認が行われないため、両ノードが同時に多数派となり、フェイルオーバ処理を開始してしまいます。

この場合には、先にハートビートの途絶を検知し、フェイルオーバ処理を開始したノードが IPMI STONITH を発動し、相手ノードの電源を遮断します。

5.5 Quorum / Witness Server Kit の設定(2)

2 ノード構成を維持し、TCP_Remote モードを利用する場合

以下のようなネットワーク構成で、router1 を Quorum Host として指定します。



この構成のポイントは、コミュニケーションパスに使う経路が Quorum Host へのアクセス経路と重なっていることです。このことにより、ネットワーク障害発生時に障害ノードから Quorum Host へのアクセス経路が失われますので、両ノードが同時に多数派となってしまうことを回避できます。

■ コミュニケーションパスの作成

各ノード間で、コミュニケーションパスを作成します。コミュニケーションパスの少なくとも一つが Quorum Host へのネットワーク経路と重なるよう考慮します。

- SIOS_Primary と SIOS_Standby の間に、192.168.16.0 と 192.168.32.0 ネットワークを利用したコミュニケーションパスを作成します。
- さらに、SIOS_Primary と SIOS_Standby の間には Quorum Host への共通の経路である 10.1.6.0 ネットワークを利用したコミュニケーションパスを作成します。

■ パラメータの設定

/etc/default/LifeKeeper を編集し、パラメータを以下のように設定します。原則的に、この設定は全てのノードで共通にしてください。

```
# vi /etc/default/LifeKeeper

QUORUM_MODE=tcp_remote                  # Quorum Check のモード
WITNESS_MODE=none                       # Witness Check を行わない
QUORUM_HOSTS=router1:80                 # 問合せ先ホスト:ポート
QUORUM_TIMEOUT_SECS=20                  # 問合せのタイムアウト
QUORUM_LOSS_ACTION=fastkill             # 少数派ノードの電源断動作
```

■ ハートビート通信途絶時の動作

A) SIOS_Primary のネットワーク機能が異常停止した場合

ハートビート通信が途絶するシチュエーションはいくつか考えられますが、ここでは SIOS_Primary にノード障害が発生し、ネットワーク機能に問題が生じた場合を想定します。

- ➔ SIOS_Primary は Quorum Host である router1 への接続を試行します。しかし、ネットワーク機能に障害が発生しており通信が行えません。そのため自ノードが少数派であると判断し、自発的な電源断を行います。
- ➔ SIOS_Standby は 10.1.6.0 ネットワークを経由して Quorum Host である router1 への接続が可能です。そのため、自ノードが多数派であると判断し、サービスの起動を行います。

結論：SIOS_Primary は電源断、SIOS_Standby がリソース起動

B) コミュニケーションパスのネットワークに障害発生した場合

192.168.16.0 および 192.168.32.0、10.1.6.0 ネットワークのすべてに障害が発生して、ハートビート通信が途絶したケースです。

- ➔ SIOS_Primary と SIOS_Standby はどちらも Quorum Host である router1 への接続が行えません。したがって両ノードとも少数派となります。リソースが稼働している SIOS_Primary は自発的な電源断を行い、リソースが稼働していない SIOS_Standby は、リソースの起動は行いません。

結論：SIOS_Primary が電源断、SIOS_Standby はリソース起動せず

■ IPMI STONITH による電源断が必要となるシチュエーション

A) SIOS_Primary が完全に応答停止状態となった場合

SIOS_Primary の OS が完全に応答停止してしまったケースです。この場合、Quorum Check による SIOS_Primary の自発的な電源断は期待できません。

しかし、SIOS_Standby が Quorum Check の結果多数派となり、フェイルオーバ処理を開始します。フェイルオーバ処理の開始前に IPMI STONITH が発動して SIOS_Primary の電源を強制的に遮断します。

そのため、スプリットブレインが発生することなく、リソースのフェイルオーバを実施することができます。

B) ハートビート断にも関わらず両ノードが Quorum Host と通信できる場合

今回の構成ではこのような事象は発生しませんが、ハートビート通信の途絶時に両ノードとも Quorum Host と通信可能な状態であると、両ノードが多数派となり、リソースの起動処理を開始してしまいます。

このような場合には、先にハートビートの途絶を検知し、フェイルオーバ処理を開始したノードが IPMI STONITH を発動することで。相手ノードの電源を遮断してスプリットブレインを回避します。

5.6 ファイルシステムリソースの作成

(1) ファイルシステムのマウント

稼働系ノードである SIOS_Primary ノードで、MD3200 上の LU 領域にファイルシステムを作成し、任意のディレクトリにマウントします。

```
# fdisk /dev/mapper/mpath1
# mkfs.ext3 /dev/mapper/mpath1p1
# mount /dev/mapper/mpath1p1 /u01/
```

(2) ファイルシステムリソースの作成

LifeKeeper GUI 管理画面を起動します。

```
# /opt/LifeKeeper/bin/lkGUIapp > /dev/null 2>&1 &
```

その後、管理ガイドの手順に従って、マウントした領域をファイルシステムリソースとして保護します。管理ガイドは以下のリンクより参照することができます。

Creating a File System Resource Hierarchy - SIOS Technology Corp Wiki

6 お問い合わせ

本ドキュメントの記載内容についてのお問い合わせ先

■ LifeKeeper 製品の導入を検討中のお客様

弊社パートナー営業部までお問い合わせください。

TEL:03-6860-5111 (受付時間 9:00～17:00 土日祝祭日および弊社休業日は除く)

お問い合わせメールフォーム

https://www.sios.com/products/bcp/lkdk/contact/

■ LifeKeeper 製品をご購入済みのお客様

弊社 LifeKeeper 製品サポート窓口までお問い合わせください。

購入後のお問い合わせ

https://www.sios.com/products/bcp/lkdk/contact/support_lk.html

7 免責事項

本ドキュメントは 2011 年 9 月にサイオステクノロジー株式会社（以下、弊社）にて独自に実施した検証結果に基づき、参考情報としてお客様に提供することを目的として作成しております。