



Red Hat Enterprise Linux 7 DM Multipath

DM Multipath の設定と管理

DM Multipath の設定と管理

法律上の通知

Copyright © 2015 Red Hat, Inc. and others.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, MetaMatrix, Fedora, the Infinity Logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux® is the registered trademark of Linus Torvalds in the United States and other countries.

Java® is a registered trademark of Oracle and/or its affiliates.

XFS® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack® Word Mark and OpenStack Logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

概要

本ガイドは Red Hat Enterprise Linux 7 の Device-Mapper Multipath の使い方について説明しています。

目次

第1章 Device Mapper を使ったマルチパス設定	2
1.1. Red Hat Enterprise Linux 7.1 で新たに追加された機能および変更点	2
1.2. DM Multipath の概要	2
1.3. ストレージレイの対応	5
1.4. DM Multipath のコンポーネント	5
1.5. DM Multipath 設定の概要	6
第2章 マルチパスデバイス	7
2.1. マルチパスデバイス識別子	7
2.2. クラスタ内では整合性のあるマルチパスデバイス名を維持する	7
2.3. マルチパスデバイスの属性	8
2.4. 論理ボリューム内のマルチパスデバイス	8
第3章 DM Multipath の設定	10
3.1. DM Multipath の設定	10
3.2. マルチパスデバイスの作成時にローカルのディスクを無視する	11
3.3. ストレージデバイスを設定する	13
3.4. initramfs ファイルシステムでマルチパスを設定する	14
第4章 DM Multipath 設定ファイル	15
4.1. 設定ファイルの概要	15
4.2. 設定ファイルの blacklist セクション	16
4.3. 設定ファイルの defaults セクション	18
4.4. 設定ファイルの multipaths セクション	24
4.5. 設定ファイルの devices セクション	27
第5章 DM Multipath の管理とトラブルシューティング	33
5.1. オンラインのマルチパスデバイスのサイズ変更	33
5.2. root ファイルシステムをシングルパスデバイスからマルチパスデバイスへ移行	33
5.3. swap ファイルシステムをシングルパスデバイスからマルチパスデバイスへ移行	35
5.4. マルチパスデーモン	35
5.5. 多数の LUN に伴う問題	35
5.6. queue_if_no_path 機能に関する問題	36
5.7. Multipath コマンドの出力	36
5.8. multipath コマンドを使ったマルチパスクエリー	37
5.9. multipath コマンドのオプション	38
5.10. dmsetup コマンドを使ってデバイスマッパーのエントリーを特定する	38
5.11. 対話式コンソール multipathd を使用してトラブルシューティングを行う	39
付録A 改訂履歴	40
索引	41

第1章 Device Mapper を使ったマルチパス設定

DM-Multipath (Device mapper multipath) を使用すると複数のサーバーノードと複数のストレージレイ間の複数の I/O パスをひとつのデバイスに設定することができます。I/O パスとは物理的な SAN 接続のことで各ケーブルやスイッチ、コントローラーなどが含まれます。複数の I/O パスを集め、集めたパスで構成される新しいデバイスをひとつ作成するのがマルチパス機能です。

この章では Red Hat Enterprise Linux 7 の初期リリースの後に追加された DM-Multipath の機能についてまとめています。また、DM-Multipath および各コンポーネントについての大きな概要、DM-Multipath の設定についての概要についても説明しています。

1.1. Red Hat Enterprise Linux 7.1 で新たに追加された機能および変更点

Red Hat Enterprise Linux 7.1 ではドキュメントおよび機能に以下のような更新または変更が行われています。

- ※ [表5.1 「役立つ multipath コマンドのオプション」](#) に `multipath` コマンドの `-w` と `-W` のオプションに関する説明が加えられています。このオプションを使用すると `wwids` ファイルの管理が容易になります。
- ※ `multipath.conf` ファイルの `features` パラメーターの `values` 引数に使用できる追加オプションについての説明が [4章DM Multipath 設定ファイル](#) に加えられています。
- ※ [表4.1 「マルチパス設定の defaults セクション」](#) に `force_sync` パラメーターの説明が加えられています。このパラメーターを「yes」に設定するとパスチェッカーが非同期モードでは実行されなくなります。

さらに、ドキュメント全体にわたり技術的な内容の若干の修正と明確化を行いました。

1.2. DM Multipath の概要

DM Multipath は以下のような目的に使用できます。

※ 冗長性

DM-Multipath を active/passive 設定にするとフェイルオーバーを行うことができます。active/passive 設定の場合、常に全体の半分のパスだけが使用されます。I/O パス上のいずれかの構成要素(ケーブル、スイッチ、コントローラーなど)に障害が発生すると DM-Multipath により別のパスへの切り替えが行われます。

※ パフォーマンスの向上

DM-Multipath を active/active モードに設定すると I/O はラウンドロビン式ですべてのパス全体に分散されます。I/O パスの負荷を検出し、動的に負荷分散の再調整を行うことができる設定もあります。

サーバーから RAID デバイスへの I/O パスが 2 つある active/passive 設定を [図1.1 「active/passive マルチパス設定 - RAID デバイスが1つ」](#) に示します。サーバーに 2 つの HBA、2 つの SAN スイッチ、2 つの RAID コントローラーがあります。

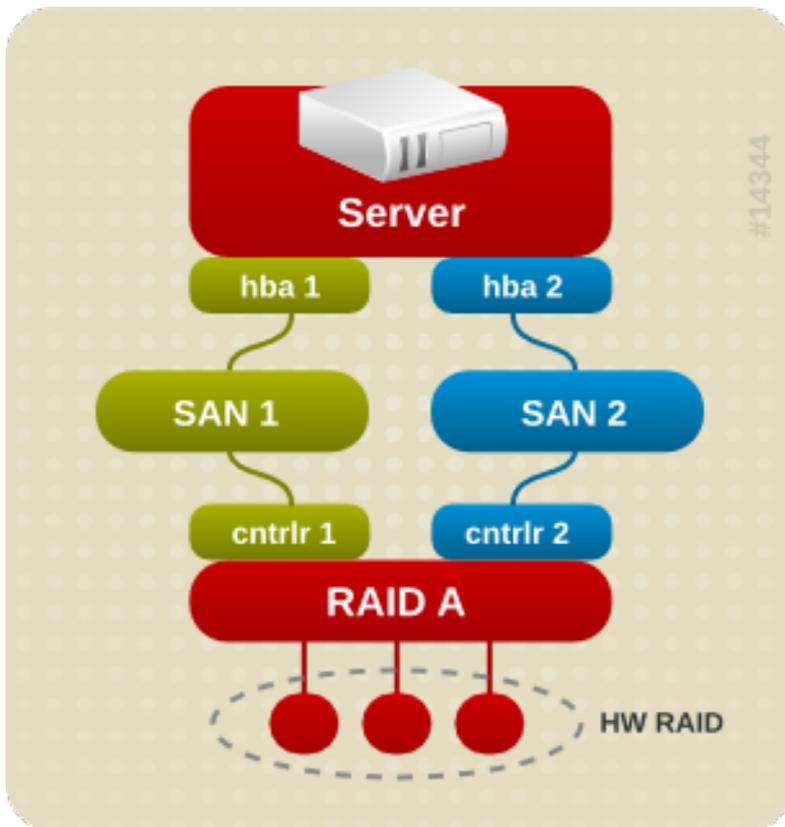


図1.1 active/passive マルチパス設定 - RAID デバイスが1つ

この設定では、hba1、SAN1、およびコントローラー 1 を通る 1 つ目の I/O パスと、hba2、SAN2、およびコントローラー 2 を通る 2 つ目の I/O パスがあります。この設定には、障害が発生する可能性のあるポイントが数多く存在します。

- ✧ HBA の障害
- ✧ FC ケーブルの障害
- ✧ SAN スイッチの障害
- ✧ アレイコントローラーポートの障害

DM-Multipath が設定されていると、いずれのポイントで障害が発生しても DM-Multipath により代替 I/O パスへの切り替えが行われます。

[図1.2 「active/passive マルチパス設定 - RAID デバイスが2つ」](#)ではもう少し複雑な active/passive 設定を示します。サーバーに 2 つの HBA があり、2 つの SAN スイッチ、RAID コントローラーが 2 つずつ付いた 2 つの RAID デバイスがあります。

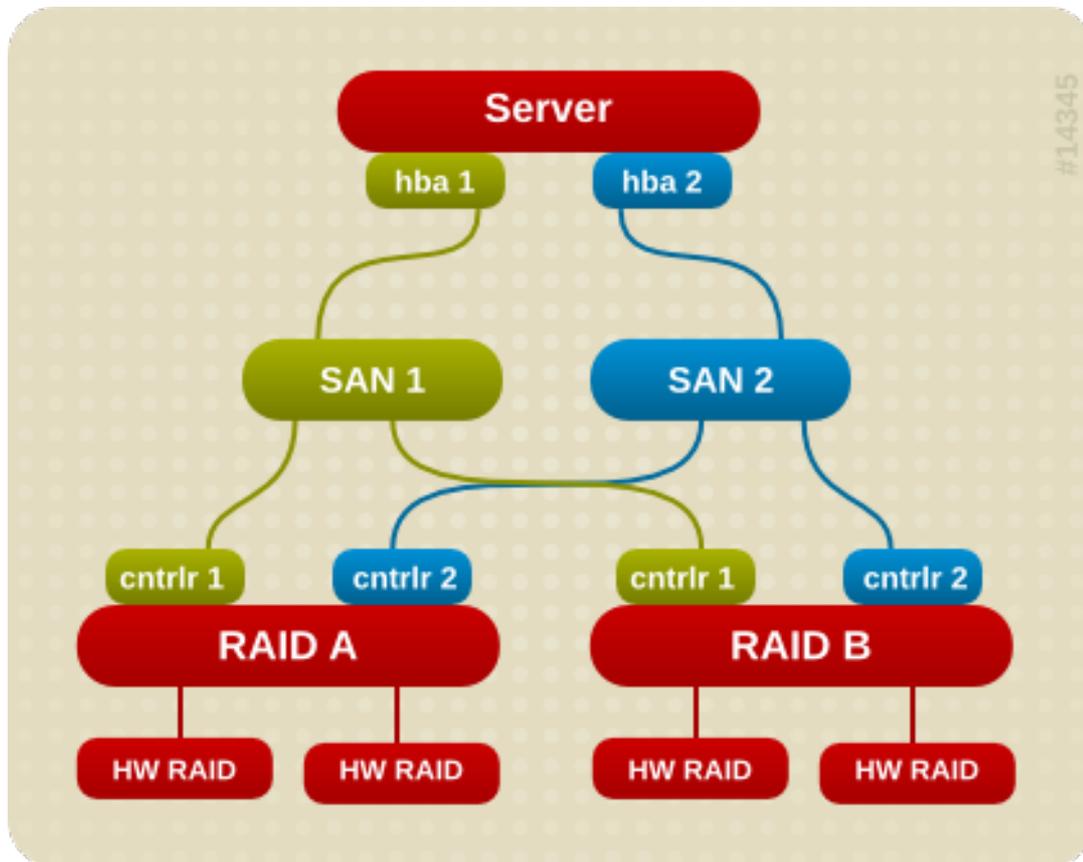


図1.2 active/passive マルチパス設定 - RAID デバイスが2つ

図1.2 「active/passive マルチパス設定 - RAID デバイスが2つ」に示した例では、それぞれの RAID デバイスに2つの I/O パスがあります (図1.1 「active/passive マルチパス設定 - RAID デバイスが1つ」に示した例と同様)。DM-Multipath が設定されていると、どちらかの RAID デバイスへの I/O パスのいずれのポイントで障害が発生しても DM-Multipath によりそのデバイスへの代替 I/O パスへの切り替えが行われます。

図1.3 「active/active マルチパス設定 - RAID デバイスが1つ」ではサーバーに2つの HBA があり、1つの SAN スイッチ、2つの RAID コントローラーがある active/active 設定を示します。サーバーからストレージデバイスへの I/O パスは4つあります。

- ※ hba1 から controller1 まで
- ※ hba1 から controller2 まで
- ※ hba2 から controller1 まで
- ※ hba2 から controller2 まで

この設定では、I/O をこれらの4つのパスに分散することが可能です。

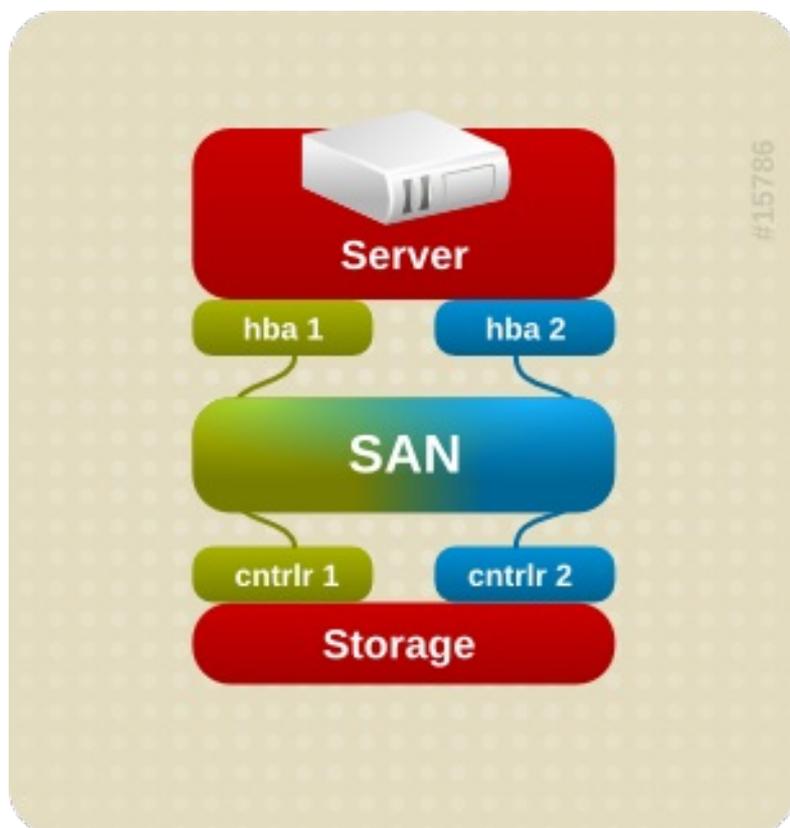


図1.3 active/active マルチパス設定 - RAID デバイスが1つ

1.3. ストレージレイの対応

DM-Multipath に対応しているストレージレイで最も一般的なものについては DM-Multipath の設定にデフォルトで含まれています。対応デバイスなどデフォルトの設定値については以下のいずれかのコマンドを実行して確認してください。

```
# multipathd show config
# multipath -t
```

DM Multipath に対応しているがデフォルトの設定には含まれていないようなストレージレイの場合は DM Multipath 設定ファイルの `multipath.conf` にそのストレージレイを追加する必要があります。DM Multipath 設定ファイルの詳細は [4章DM Multipath 設定ファイル](#) を参照してください。

I/O エラーとパスの切り替えに特殊な処理を必要とするストレージレイがあります。こうしたストレージレイの場合には別途、ハードウェアハンドラーカーネルモジュールが必要になります。

1.4. DM Multipath のコンポーネント

[表1.1 「DM Multipath のコンポーネント」](#) で DM-Multipath のコンポーネントを説明します。

表1.1 DM Multipath のコンポーネント

コンポーネント	説明
<code>dm_multipath</code> カーネルモジュール	I/O を再ルーティングして、パス群とパスグループのフェイルオーバーをサポートします。
<code>mpathconf</code> ユーティリティ	デバイスマッパーのマルチパスを設定し、有効化します。

コンポーネント	説明
multipath コマンド	マルチパスデバイスの表示、設定をします。通常、 <code>/etc/rc.sysinit</code> で起動されますが、ブロックデバイスを追加した場合は udev プログラムで起動させることもできます。
multipathd デーモン	パスを監視します。パスに障害が発生したあと復旧した際、パスグループスイッチを起動できます。対話式でマルチパスデバイスに変更を行うことができます。 <code>/etc/multipath.conf</code> ファイルに変更が行われた場合は常にデーモンを再起動する必要があります。
kpartx コマンド	デバイス上のパーティション用にデバイスマッパーデバイスを作成します。DM Multipath の DOS ベースパーティションの場合はこのコマンドを使用する必要があります。 kpartx はそれ自体のパッケージで提供されますが、 device-mapper-multipath パッケージはそのパッケージに依存します。

1.5. DM Multipath 設定の概要

DM Multipath には一般的なマルチパス設定に適したデフォルト設定がコンパイルされています。DM Multipath をシステムに設定する基本的な手順を以下に示します。

1. **device-mapper-multipath** rpm をインストールします。
2. **mpathconf** コマンドで設定ファイルを作成しマルチパスを有効にします。設定ファイルを編集する必要がない場合は、このコマンドでマルチパスデーモンを起動することもできます。
3. 必要に応じて **multipath.conf** 設定ファイルを編集し、デフォルト値の変更、更新したファイルの保存を行います。
4. マルチパスデーモンを起動します。

マルチパス設定の手順については [3章DM Multipath の設定](#) をご覧ください。

第2章 マルチパスデバイス

DM-Multipath を使用しない場合、サーバーノードからストレージコントローラーへの各パスはシステムでは別々のデバイスとして扱われます。同じサーバーノード、同じストレージコントローラーをつなぐ I/O パスであっても同様です。DM-Multipath は複数のデバイスをまとめて一つのマルチパスデバイスを作成し、複数の I/O のパスを論理的に整理する方法を提供しています。

2.1. マルチパスデバイス識別子

マルチパスデバイスはそれぞれグローバルに固有で不変となる World Wide Identifier (WWID) を持っています。マルチパスデバイス名はデフォルトでは WWID が付けられます。マルチパス設定ファイル内の **user_friendly_names** オプションを設定すると **mpathn** 形式のノード固有の名前にエイリアスを作成することができます。

例えば、2 つの HBA を搭載したノードと 2 つポートがあるストレージコントローラーをゾーニングされていない 1 つの FC スイッチで接続した場合、**/dev/sda**、**/dev/sdb**、**dev/sdc**、**/dev/sdd** の 4 つのデバイスが認識されます。DM-Multipath はマルチパス設定に従ってこれら 4 つのデバイスへ I/O を再ルーティングする 1 つのデバイスを固有の WWID を付けて作成します。**user_friendly_names** 設定オプションを **yes** に設定していた場合はマルチパスデバイスの名前が **mpathn** になります。

新しいデバイスを DM-Multipath の管理下に置くと **/dev** ディレクトリ配下の **/dev/mapper/mpathn** と **/dev/dm-n** の 2 ヶ所に表示されます。

- ※ **/dev/mapper** のデバイスは起動プロセスの初期に作成され、論理ボリュームの作成時などマルチパス設定したデバイスへのアクセスに使用されます。
- ※ **/dev/dm-n** 形式のデバイスはすべて内部的な使用を目的としているため、管理者の方は絶対に使用しないでください。

user_friendly_names 設定オプションなどマルチパス設定のデフォルト値については「[設定ファイルの defaults セクション](#)」を参照してください。

マルチパスデバイスに任意の名前を付けることもできます。マルチパス設定ファイルの **multipaths** セクションにある **alias** オプションを使用します。マルチパス設定ファイルの **multipaths** セクションについての詳細は「[設定ファイルの multipaths セクション](#)」を参照してください。

2.2. クラスター内では整合性のあるマルチパスデバイス名を維持する

user_friendly_names 設定オプションを **yes** に設定するとマルチパスデバイスの名前は 1 ノードに対しては固有となりますが、そのマルチパスデバイスを使用している全ノードで同じ名前になる保証はありません。同様に、**multipath.conf** 設定ファイルの **multipaths** セクションでデバイスに **alias** オプションを設定する場合、クラスター内の全ノードでその名前が自動的に同じにはなりません。LVM を使用してマルチパスデバイスから複数の論理デバイスを作成している場合は問題にならないはずですが、すべてのノードでマルチパスデバイス名を同じ名前にする必要がある場合は **user_friendly_names** オプションを **yes** に設定しない、またデバイスにエイリアスを設定しないことをお勧めします。**user_friendly_names** を **yes** に設定したりデバイスにエイリアスを設定しなければデバイス名はデフォルトでそのデバイスの WWID になるため整合性を維持することができます。

ただし、システムで定義される **user_friendly_names** をクラスター内の全ノードに対して固有にしたい場合には以下の手順に従うと整合性を維持することができます。

1. すべてのマルチパスデバイスの設定をひとつのマシンで行います。
2. 次のコマンドを使って他のマシンのマルチパスデバイスをすべて無効にします。

```
# service multipathd stop
# multipath -F
```

3. クラスター内の 1 番目のマシンの `/etc/multipath/bindings` ファイルを他のすべてのマシンにコピーします。
4. 先ほど `multipathd` デモンを無効にしたすべてのマシンで以下のコマンドを実行して再度デモンを有効にします。

```
# service multipathd start
```

新しいデバイスを追加する場合はこの手順を繰り返す必要があります。

エイリアスも同様にクラスター内のすべてのノードに対して固有となるよう設定する場合は、同じ手順でクラスター内の各ノードの `/etc/multipath.conf` ファイルが同じになるようにします。

1. 1つのマシンの `multipath.conf` ファイルでマルチパスデバイスのエイリアスを設定します。
2. 次のコマンドを使って他のマシンのマルチパスデバイスをすべて無効にします。

```
# service multipathd stop
# multipath -F
```

3. そのマシンの `/etc/multipath.conf` ファイルをクラスター内の他のマシンすべてにコピーします。
4. 先ほど `multipathd` デモンを無効にしたすべてのマシンで以下のコマンドを実行して再度デモンを有効にします。

```
# service multipathd start
```

新しいデバイスを追加する場合はこの手順を繰り返す必要があります。

2.3. マルチパスデバイスの属性

`user_friendly_names` や `alias` オプションの他にもマルチパスデバイスには多くの属性があります。マルチパス設定ファイルの `multipaths` セクションに特定のマルチパスデバイス用のエントリーを作成するとそのデバイスの属性を変更することができます。マルチパス設定ファイルの `multipaths` セクションについては [「設定ファイルの multipaths セクション」](#) を参照してください。

2.4. 論理ボリューム内のマルチパスデバイス

マルチパスデバイスを作成した後は、LVM 物理ボリュームを作成する時に物理デバイス名を使用するのと同じように、マルチデバイス名を使用することができます。例えば、マルチパスデバイスの名前が `/dev/mapper/mpath0` の場合、以下のコマンドを実行すると `/dev/mapper/mpatha` は物理ボリュームとして認識されます。

```
pvcreate /dev/mapper/mpatha
```

上記のコマンドで得られた LVM 物理デバイスは、他の LVM 物理デバイスを使用するのと同じように、LVM ボリュームグループを作成する時に使用することができます。



注記

複数のパーティションが設定されたデバイス全体に LVM 物理ボリュームを作成しようとするとき、**pvcreate** コマンドは失敗します。特に指定しない限り、Anaconda や Kickstart インストールプログラムはすべてのブロックデバイスに対して空のパーティションテーブルを作成するので注意してください。パーティションではなくデバイス全体を使用したい場合には、そのデバイスから既存のパーティションを削除する必要があります。既存パーティションの削除は **kpartx -d** や **fdisk** のコマンドを使用すると行うことができます。システムに 2TB 以上のブロックデバイスがある場合は **parted** コマンドを使用してパーティションを削除します。

複数の物理デバイスで active/passive のアレイを構成しマルチパスを使用させる LVM 論理ボリュームを作成する場合は、**/etc/lvm/lvm.conf** ファイルにフィルターを含ませこのような複数デバイスで構成されマルチパスを使用するディスクを除外してください。フィルターを使って除外しないと、I/O の受信時に active パスが自動的に passive パスに変更され LVM でその passive パスがスキャンされる度、マルチパスによりフェイルオーバーとフェイルバックが繰り返されることになります。passive パスを active パスに変更する際にコマンドを必要とする active/passive アレイの場合、このような状態が発生すると LVM で警告メッセージが表示されます。

LVM 設定ファイル (**lvm.conf**) 内で全ての SCSI デバイスをフィルターするには、そのファイルの **devices** セクションに以下のフィルターを組み込みます。

```
filter = [ "r/block/", "r/disk/", "r/sd.*/", "a/.*/" ]
```

第3章 DM Multipath の設定

この章では、DM-Multipath を設定するため以下のような手順を順を追って説明していきます。

- ※ 基本的な DM-Multipath のセットアップ
- ※ ローカルディスクを無視する
- ※ 設定ファイルにデバイスを追加する
- ※ `initramfs` ファイルシステムでマルチパスを起動する

3.1. DM Multipath の設定

システムに DM-Multipath をセットアップする前に、システムが更新され `device-mapper-multipath` パッケージが含まれているか確認してください。

`mpathconf` ユーティリティを使用してマルチパスをセットアップします。マルチパス設定ファイルの `/etc/multipath.conf` が作成されます。

- ※ `/etc/multipath.conf` ファイルが既に存在する場合は `mpathconf` ユーティリティによりファイルに編集が行われます。
- ※ `/etc/multipath.conf` ファイルが存在しない場合は、`/usr/share/doc/device-mapper-multipath-0.4.9/multipath.conf` ファイルが開始ファイルとして使用されます。
- ※ `/usr/share/doc/device-mapper-multipath-0.4.9/multipath.conf` ファイルが存在しない場合は、`mpathconf` ユーティリティにより `/etc/multipath.conf` ファイルがゼロから作成されます。

`/etc/multipath.conf` ファイルを編集する必要がない場合は、以下のコマンドを実行すると基本的なフェイルオーバー設定用の DM-Multipath をセットアップすることができます。このコマンドでマルチパス設定が有効になり、`multipathd` デーモンが起動されます。

```
# mpathconf --enable --with_multipathd y
```

`multipathd` デーモンを起動する前に `/etc/multipath.conf` ファイルを編集する必要がある場合は、以下の手順に従い基本的なフェイルオーバー設定用の DM-Multipath をセットアップします。

1. `--enable` オプションを指定して `mpathconf` コマンドを実行します。

```
# mpathconf --enable
```

`mpathconf` コマンドにこれ以外のオプションが必要な場合には `mpathconf` の man ページをご覧ください。また、`mpathconf` コマンドに `--help` オプションを指定して実行してください。

```
# mpathconf --help
usage: /sbin/mpathconf <command>

Commands:
Enable: --enable
Disable: --disable
Set user_friendly_names (Default y): --user_friendly_names <y|n>
```

```
Set find_multipaths (Default y): --find_multipaths <y|n>
Load the dm-multipath modules on enable (Default y): --with_module
<y|n>
start/stop/reload multipathd (Default n): --with_multipathd <y|n>
```

- 必要に応じて `/etc/multipath.conf` ファイルを編集します。DM-Multipath 用のデフォルト設定はシステムにコンパイルされているため、`/etc/multipath.conf` ファイルに明示的にセットする必要はありません。

`path_grouping_policy` のデフォルト値は `failover` にセットされています。このため、この例ではデフォルト値を変更する必要はありません。設定ファイル内のデフォルト値の変更に関する詳細は [4章DM Multipath 設定ファイル](#) をご覧ください。

設定ファイルの初期デフォルトセクションによりマルチパスデバイス名が `mpathn` の形式になるよう設定されます。この設定がないと、マルチパスデバイスの名前はそのデバイスの WWID にエイリアス設定されます。

- 必要に応じて設定ファイルを保存し、エディターを終了します。
- 以下のコマンドを実行します。

```
# service multipathd start
```

設定ファイルで `user_friendly_name` の値が `yes` にセットされているため、マルチパスデバイスは `/dev/mapper/mpathn` として作成されます。別の名前でエイリアスを設定する方法については [4章DM Multipath 設定ファイル](#) を参照してください。

`user_friendly_name` を使用したくない場合は以下のコマンドを実行します。

```
# mpathconf --enable --user_friendly_names n
```



注記

マルチパスデーモンを起動した後、マルチパス設定ファイルの編集が必要になった場合は `service multipathd reload` コマンドを使って行った変更を反映させてください。

3.2. マルチパスデバイスの作成時にローカルのディスクを無視する

内蔵ディスクにローカルの SCSI カードが搭載されているマシンがあります。DM-Multipath はこれらのデバイスには推奨できません。`find_multipaths` 設定パラメーターを `yes` にセットすると、こうしたデバイスをブラックリストに指定する必要がなくなります。`find_multipaths` 設定パラメーターについては「[設定ファイルの defaults セクション](#)」を参照してください。

`find_multipaths` 設定パラメーターを `yes` にセットしない場合は、以下の手順に従ってマルチパス設定ファイルを修正するとマルチパスの設定時にローカルのディスクを無視することができます。

- 内蔵ディスクを見つけてブラックリストに指定するディスクとして印を付けます。

この例では `/dev/sda` が内蔵ディスクです。デフォルトのマルチパス設定ファイル内で元々設定されているように、`multipath -v2` を実行するとマルチパスマップ内にローカルディスク `/dev/sda` が表示されます。

`multipath` コマンドの出力に関する詳細は「[Multipath コマンドの出力](#)」をご覧ください。

```
# multipath -v2
create: SIBM-ESXSST336732LC____F3ET0EP0Q000072428BX1 undef
WINSYS,SF2372
size=33 GB features="" hwhandler="" wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 0:0:0:0 sda 8:0  [-----]

device-mapper ioctl cmd 9 failed: Invalid argument
device-mapper ioctl cmd 14 failed: No such device or address
create: 3600a0b80001327d80000006d43621677 undef WINSYS,SF2372
size=12G features='0' hwhandler='0' wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 2:0:0:0 sdb 8:16  undef ready  running
  ` - 3:0:0:0 sdf 8:80 undef ready  running

create: 3600a0b80001327510000009a436215ec undef WINSYS,SF2372
size=12G features='0' hwhandler='0' wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 2:0:0:1 sdc 8:32 undef ready  running
  ` - 3:0:0:1 sdg 8:96 undef ready  running

create: 3600a0b80001327d800000070436216b3 undef WINSYS,SF2372
size=12G features='0' hwhandler='0' wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 2:0:0:2 sdd 8:48 undef ready  running
  ` - 3:0:0:2 sdg 8:112 undef ready  running

create: 3600a0b80001327510000009b4362163e undef WINSYS,SF2372
size=12G features='0' hwhandler='0' wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 2:0:0:3 sdd 8:64 undef ready  running
  ` - 3:0:0:3 sdg 8:128 undef ready  running
```

2. デバイスマッパーによる `/dev/sda` のマルチパスマップへのマッピングを阻止するため、`/etc/multipath.conf` ファイルの `blacklist` セクションを編集してこのデバイスを追加します。 `devnode` タイプで `sda` デバイスをブラックリスト指定することも可能ですが、`/dev/sda` が再起動後に同じく `/dev/sda` になる保証はないため安全な方法ではありません。デバイスごとにブラックリスト指定する場合はそのデバイスの WWID を使用します。

`multipath -v2` コマンドに対する出力を見ると、`/dev/sda` デバイスの WWID は `SIBM-ESXSST336732LC____F3ET0EP0Q000072428BX1` になっています。このデバイスをブラックリスト指定する場合は `/etc/multipath.conf` ファイルに以下の記述を含めます。

```
blacklist {
    wwid SIBM-ESXSST336732LC____F3ET0EP0Q000072428BX1
}
```

3. `/etc/multipath.conf` ファイルを更新したら `multipathd` デーモンにファイルを再ロードするよう手作業で指示する必要があります。以下のコマンドで更新した `/etc/multipath.conf` ファイルを再ロードします。

```
# service multipathd reload
```

4. 以下のコマンドを実行してマルチパスデバイスを削除します。

```
# multipath -f SIBM-ESXSST336732LC_____F3ET0EP0Q000072428BX1
```

5. **multipath -ll** コマンドを実行して現在のマルチパス設定を表示させ、削除が正しく行われたか確認します。**multipath -ll** コマンドについては「[multipath コマンドを使ったマルチパスクエリー](#)」を参照してください。

ブラックリスト指定されたデバイスが再度追加されていないことを確認する場合は、以下の例のように **multipath** コマンドを実行します。**-v** オプションを指定しない場合、**multipath** コマンドのデフォルト詳細レベルは **v2** になります。

```
# multipath

create: 3600a0b80001327d80000006d43621677 undef WINSYS,SF2372
size=12G features='0' hwhandler='0' wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 2:0:0:0 sdb 8:16  undef ready  running
  ` - 3:0:0:0 sdf 8:80  undef ready  running

create: 3600a0b80001327510000009a436215ec undef WINSYS,SF2372
size=12G features='0' hwhandler='0' wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 2:0:0:1 sdc 8:32  undef ready  running
  ` - 3:0:0:1 sdg 8:96  undef ready  running

create: 3600a0b80001327d800000070436216b3 undef WINSYS,SF2372
size=12G features='0' hwhandler='0' wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 2:0:0:2 sdd 8:48  undef ready  running
  ` - 3:0:0:2 sdg 8:112 undef ready  running

create: 3600a0b80001327510000009b4362163e undef WINSYS,SF2372
size=12G features='0' hwhandler='0' wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 2:0:0:3 sdd 8:64  undef ready  running
  ` - 3:0:0:3 sdg 8:128 undef ready  running
```

3.3. ストレージデバイスを設定する

DM-Multipath に対応しているストレージレイで最も一般的なものについては DM-Multipath の設定にデフォルトで含まれています。対応デバイスなどデフォルトの設定値については以下のいずれかのコマンドを実行して確認してください。

```
# multipathd show config
# multipath -t
```

既知のマルチパスデバイスとしてデフォルトではサポートされていないストレージデバイスを追加する必要がある場合は、**/etc/multipath.conf** ファイルを編集して適切なデバイス情報を記入します。

例えば、HP Open-V シリーズに関する情報を追加する場合は以下のようなエントリを記述します。この例ではすべてのパスに障害が発生した場合には 1 分間 (または 5 秒ごとの再試行で 12 回の再試行を行う) のキュー待ちを行うようデバイスを設定しています。

```
devices {
    device {
```

```
        vendor "HP"  
        product "OPEN-V"  
        no_path_retry 12  
    }  
}
```

設定ファイルの **devices** セクションについての詳細は [「設定ファイルの devices セクション」](#) をご覧ください。

3.4. **initramfs** ファイルシステムでマルチパスを設定する

initramfs ファイルシステムでマルチパスを設定することができます。マルチパスの設定後、以下のオプションを使用して **dracut** を実行し、マルチパス設定ファイルで **initramfs** ファイルシステムを再構築することができます。

```
# dracut --force --add multipath --include /etc/multipath /etc/multipath
```

initramfs ファイルシステムからマルチパスを実行してマルチパス設定ファイルに変更を加えた場合、変更を反映するには **initramfs** ファイルシステムを再構築する必要があります。

第4章 DM Multipath 設定ファイル

DM Multipath ではマルチパスでもっともよく使用する設定値がデフォルトで提供されています。また DM Multipath に対応しているストレージアレイで最も一般的なものについては DM-Multipath の設定にデフォルトで含まれています。対応デバイスなどデフォルトの設定値については以下のいずれかのコマンドを実行して確認してください。

```
# multipathd show config
# multipath -t
```

DM-Multipath の設定値は `/etc/multipath.conf` 設定ファイルを編集すると無効にすることができます。また、必要に応じて対応ストレージアレイだがデフォルトでは設定ファイルに含まれていないストレージアレイを設定ファイルに追加することもできます。

注記

initramfs ファイルシステムでマルチパスの設定を実行することができます。**initramfs** ファイルシステムでマルチパスを実行してマルチパス設定ファイルに変更を加えた場合は変更が反映されるよう **initramfs** ファイルシステムを再構築する必要があります。マルチパスで**initramfs** ファイルシステムを再構築する方法については [「initramfs ファイルシステムでマルチパスを設定する」](#) を参照してください。

本章では `multipath.conf` ファイルの解析や変更について説明しています。以下のトピックに関する項が含まれています。

- ※ 設定ファイルの概要
- ※ 設定ファイルの `blacklist` セクション
- ※ 設定ファイルの `defaults` セクション
- ※ 設定ファイルの `multipaths` セクション
- ※ 設定ファイルの `devices` セクション

マルチパス設定ファイルで指定するのは使用状況に応じて必要とされるセクションやデフォルト値の変更が必要なセクションのみです。使用環境には無関係なセクションやデフォルト値を無効にする必要がないセクションなどがある場合は初期ファイルに指定されている通りコメントアウトしたまま残しておいて構いません。

設定ファイルでは正規表現の記述構文を使用することができます。

設定ファイルの詳細については `multipath.conf(5)` の man ページをご覧ください。

4.1. 設定ファイルの概要

マルチパス設定ファイルは以下のセクションに分かれています。

blacklist

マルチパス設定の対象として考慮しないデバイスの一覧

blacklist_exceptions

ブラックリストの例外としてマルチパス設定の対象となるデバイスの一覧、この例外セクション

に指定されていなければ `blacklist` セクションのパラメーターに従ってブラックリスト指定されるデバイス

defaults

DM Multipath のデフォルトの全般設定

multipaths

マルチパスデバイスの特性に関する個別設定、ここで指定する値は設定ファイルの `defaults` と `devices` のセクションで指定されている値より優先されます。

devices

ストレージコントローラーの個別設定、ここで指定する値は設定ファイル内の `defaults` セクションで指定されている値より優先されます。対応ストレージレイだがデフォルトでは設定ファイルに含まれていないストレージレイを使用する場合はそのレイ用の `devices` サブセクションを作成する必要があります。

システムでマルチパスデバイスの属性を特定する場合、最初に `multipath` セクションの設定をチェックし、次に `devices` セクションの設定をチェック、その後 `defaults` セクションの設定をチェックします。

4.2. 設定ファイルの `blacklist` セクション

マルチパス設定ファイルの `blacklist` セクションには、システムによるマルチパスデバイス設定の際に使用しないデバイスを指定します。ブラックリスト指定されたデバイスはマルチパスデバイスへはグルーピングされません。

以前の Red Hat Enterprise Linux のリリースでは、明示的にブラックリスト指定されていないパスすべてにマルチパスデバイスの作成を試行していました。Red Hat Enterprise Linux 6 からは `find_multipaths` 設定パラメーターを `yes` に設定すると、マルチパスによるデバイスの作成は以下の 3 つの条件のうちのいずれかを満たす場合に限られます。

- ✧ ブラックリストに指定されていず同一の WWID を持つパスが少なくとも 2 つある場合
- ✧ ユーザーが手作業で `multipath` コマンドを使用しデバイスを指定して作成を強制している場合
- ✧ 任意のパスが以前に作成されたマルチパスデバイスと同一の WWID を持っている場合 (そのマルチパスデバイスがその時点で存在していない場合も含む)。マルチパスデバイスを作成すると常にそのデバイスの WWID が記憶されるため、マルチパスは同一の WWID を持つパスを認識すると自動的にそのデバイスを再度作成します。これによって、マルチパスのブラックリストを編集しなくても、マルチパスが自動的に正しいパスを選択してマルチパスデバイスにすることができます。

`find_multipaths` パラメーターを使用しないでマルチパスデバイスを作成した後、パラメーターを `yes` に設定し直した場合は、`/etc/multipath/wwids` ファイルからマルチデバイスとして作成したくないデバイスの WWID を削除しなければならない可能性があります。以下は、`/etc/multipath/wwids` ファイルの例を示しています。スラッシュ (`/`) で囲まれている部分が WWID です。

```
# Multipath wwids, Version : 1.0
# NOTE: This file is automatically maintained by multipath and
multipathd.
# You should not need to edit this file in normal circumstances.
#
# Valid WWIDs:
/3600d0230000000000e13955cc3757802/
/3600d0230000000000e13955cc3757801/
```

```

/3600d0230000000000e13955cc3757800/
/3600d02300069c9ce09d41c31f29d4c00/
/SWINSYS SF2372 0E13955CC3757802/
/3600d0230000000000e13955cc3757803/

```

find_multipaths パラメーターを **yes** に設定すれば、複数のパスがあるけれど意図的にマルチパスの設定をしたくないデバイス以外はブラックリスト指定をする必要がなくなります。このため、一般的にはラバイスのブラックリスト指定は必要ありません。

デバイスのブラックリスト指定が必要な場合は以下の基準に従って行います。

- ※ WWID で指定する、[「WWID でブラックリスト指定する」](#)の説明を参照してください
- ※ デバイス名で指定する、[「デバイス名でブラックリスト指定する」](#)の説明を参照してください
- ※ デバイスタイプで指定する、[「デバイスタイプでブラックリスト指定する」](#)の説明を参照してください

デフォルトでは、設定ファイルの初期ブラックリストセクションをコメントアウトした後でも、各種のデバイスタイプがブラックリストに指定されます。詳細は [「デバイス名でブラックリスト指定する」](#) でご覧ください。

4.2.1. WWID でブラックリスト指定する

設定ファイルの **blacklist** セクションで **wwid** エントリーを使って World-Wide IDentification によるデバイスのブラックリスト指定を行うことができます。

以下では、26353900f02796769 の WWID を持つデバイスをブラックリスト指定している行の例を示します。

```

blacklist {
    wwid 26353900f02796769
}

```

4.2.2. デバイス名でブラックリスト指定する

設定ファイルの **blacklist** セクションで **devnode** エントリーを使ってデバイス名でデバイスタイプのブラックリスト指定を行い、マルチパスデバイスにグループ化されないようにすることができます。

SCSI デバイスはすべてブラックリスト指定する行を以下に示します。sd* デバイスすべてをブラックリスト指定しています。

```

blacklist {
    devnode "^sd[a-z]"
}

```

blacklist セクションで **devnode** エントリーを使用すると、特定タイプに該当する全デバイスではなくデバイスを個別にブラックリスト指定することができます。しかしデバイスが **udev** ルールで静的にマッピングされていない限り再起動後にそのデバイス名が同じ名前になる保証がないためこの方法は推奨できません。再起動するとデバイス名は **/dev/sda** から **/dev/sdb** などに変わる可能性があります。

デフォルトでは以下の **devnode** エントリーがデフォルトの **blacklist** にコンパイルされています。エントリーでブラックリスト指定しているデバイスは通常 DM-Multipath に対応していません。これらのデバイスでマルチパスを有効にする場合は [「ブラックリストから除外する」](#) で説明しているように設定ファイルの **blacklist_exceptions** セクションで除外の指定をする必要があります。

```
blacklist {
    devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
    devnode "^(td|ha)d[a-z]"
}
```

4.2.3. デバイスタイプでブラックリスト指定する

設定ファイルの **blacklist** セクション内で **device** セクションを使用し特定のデバイスタイプを指定することができます。IBM DS4200 デバイスと HP デバイスをすべてブラックリスト指定している例を以下に示します。

```
blacklist {
    device {
        vendor "IBM"
        product "3S42"          #DS4200 Product 10
    }
    device {
        vendor "HP"
        product "*"
    }
}
```

4.2.4. ブラックリストから除外する

設定ファイルの **blacklist_exceptions** セクションを使用して、デフォルトではブラックリスト指定されているデバイスでのマルチパスを有効にすることができます。

例えば、多数のデバイスがありその中の1つのデバイス (WWID が 3600d0230000000000e13955cc3757803) だけマルチパスを有効にしたい場合、このデバイス以外のすべてのデバイスをひとつひとつブラックリスト指定するのではなく、一旦すべてのデバイスをブラックリスト指定してから **/etc/multipath.conf** ファイルに以下の行を追加して目的のデバイスだけ除外しマルチパスの有効化を許可することができます。

```
blacklist {
    wwid "*"
}

blacklist_exceptions {
    wwid "3600d0230000000000e13955cc3757803"
}
```

設定ファイル内の **blacklist_exceptions** セクションでデバイスを指定する場合、**blacklist** で指定したのと同じ方法で指定しなければなりません。例えば、**blacklist** で **devnode** エントリーを使ってデバイスを指定している場合、**exception** でそのデバイスの正しい WWID を使って除外指定を行っても適用されません。同様に **devnode** による除外は **devnode** エントリーでブラックリスト指定したデバイスにしか適用されません。**device** による除外は **device** エントリーでブラックリスト指定したデバイスにしか適用されません。

4.3. 設定ファイルの defaults セクション

/etc/multipath.conf 設定ファイルには **defaults** セクションがあり、以下のように **user_friendly_names** パラメーターを **yes** にセットすることができます。

```
defaults {
    user_friendly_names yes
}
```

上記は `user_friendly_names` パラメーターのデフォルト値を上書きしています。

設定ファイルには `defaults` のテンプレートが含まれています。このセクションは以下のようにコメントアウトされています。

```
#defaults {
#    udev_dir                /dev
#    polling_interval        10
#    selector                "round-robin 0"
#    path_grouping_policy    multibus
#    uid_attribute           ID_SERIAL
#    prio                    alua
#    path_checker            readsector0
#    rr_min_io               100
#    max_fds                 8192
#    rr_weight               priorities
#    failback                immediate
#    no_path_retry           fail
#    user_friendly_names     yes
#}
```

いずれかの設定パラメーターのデフォルト値を上書きする場合は、このテンプレートからの行を `defaults` セクションにコピーしてコメントを外します。例えば、`path_grouping_policy` パラメーターをデフォルト値の `failover` から `multibus` に上書き設定するには、以下のようにテンプレートから該当行を見つけて設定ファイルの初期 `defaults` セクションにコピーしそのコメントを外します。

```
defaults {
    user_friendly_names    yes
    path_grouping_policy    multibus
}
```

`multipath.conf` 設定ファイルの `defaults` セクションに設定する属性を [表4.1「マルチパス設定の defaults セクション」](#) で説明します。`multipath.conf` ファイルの `devices` および `multipaths` セクション内の属性指定で上書きされない限り、これらの値が DM-Multipath で使用されます。

表4.1 マルチパス設定の defaults セクション

属性	説明
<code>polling_interval</code>	2つのパスチェックの間隔を秒数で指定します。適正に機能するパスでは、チェックの間隔は $(4 * \text{polling_interval})$ まで徐々に増加します。デフォルト値は 5 です。
<code>udev_dir</code>	<code>udev</code> デバイスノードが作成されるディレクトリです。デフォルト値は、 <code>/dev</code> です。
<code>multipath_dir</code>	動的共有オブジェクトが保管されるディレクトリです。デフォルト値は、システムに依存しており、一般的には <code>/lib/multipath</code> です。

属性	説明
find_multipaths	<p>マルチパスデバイス設定用モードを指定します。このパラメーターを yes に設定するとブラックリストに指定されていないパスすべてにはデバイスを作成しなくなり、以下の3つの条件のうちのいずれかを満たす場合に限定してデバイスを作成するようになります。</p> <ul style="list-style-type: none"> - ブラックリストに指定されていず同一の WWID を持つパスが少なくとも2つある場合 - ユーザーが手作業で multipath コマンドを使用しデバイスを指定して作成を強制している場合 - 任意のパスが以前に作成されたマルチパスデバイスと同一の WWID を持っている場合。 find_multipaths を設定した状態でマルチパスデバイスを作成すると常にそのデバイスの WWID が記憶されるため、マルチパスは同一の WWID を持つパスを認識すると自動的にそのデバイスを再度作成します。これによって、マルチパスのブラックリストを編集しなくても、マルチパスが自動的に正しいパスを選択してマルチパスデバイスにすることができます。 find_multipaths パラメーターを設定しないで既にマルチパスデバイスを作成している場合は「設定ファイルの blacklist セクション」の説明を参照してください。 <p>デフォルト値は yes です。</p>
verbosity	<p>デフォルトの詳細度です。値が高いほど詳細レベルが高くなります。使用できるレベルは0から6の間です。デフォルト値は 2 です。</p>
path_selector	<p>次の I/O 動作に使用するパスの決定に使うデフォルトのアルゴリズムを指定します。使用できる値を以下に示します。</p> <p>round-robin 0: パスグループ内の全てのパスをループスルーしそれぞれに同負荷の I/O を送ります。</p> <p>queue-length 0: 未処理の I/O 要求数が最も少ないパスに次の I/O 群を送ります。</p> <p>service-time 0: 推定処理時間が最も短いパスに次の I/O 群を送ります。推定処理時間は各パスに対する未処理の I/O の合計サイズをその相対スループットで割ることにより求められます。</p> <p>デフォルト値は round-robin 0 です。</p>

属性	説明
path_grouping_policy	<p>未指定のマルチパスに適用するデフォルトのパスグループポリシーを指定します。使用できる値を以下に示します。</p> <p>failover: 優先グループ毎に1つのパス</p> <p>multibus: 1つの優先グループ内の全パスが有効</p> <p>group_by_serial: 検出されたシリアル番号毎に1つの優先グループ</p> <p>group_by_prio: パス優先値毎に1つの優先グループ。優先度はグローバル、コントローラ毎、またはマルチパスオプション毎として指定されているコールアウトプログラムによって決定されます。</p> <p>group_by_node_name: ターゲットノード名毎に1つの優先グループ。ターゲットノード名は <code>/sys/class/fc_transport/target*/node_name</code> 内でフェッチされます。</p> <p>デフォルト値は failover です。</p>
prio	<p>パスの優先値を得るため呼び出すデフォルトの関数を指定します。例えば、SPC-3 の ALUA ビットは悪用可能な prio 値を提供します。使用できる値を以下に示します。</p> <p>const: 全てのパスに優先度1をセットします。</p> <p>emc: EMC アレイのパス優先度を生成します。</p> <p>alua: SCSI-3 ALUA セットアップに基づいて、パスの優先度を生成します。</p> <p>ontap: NetApp アレイのパスの優先度を生成します。</p> <p>rdac: LSI/Engenio RDAC コントローラのパスの優先度を生成します。</p> <p>hp_sw: active/standby モードの Compaq/HP コントローラ用のパスの優先度を生成します。</p> <p>hds: Hitachi HDS Modular ストレージアレイのパスの優先度を生成します。</p> <p>デフォルト値は const です。</p>

属性	説明
features	<p>マルチパスデバイスの追加デフォルト features です。次の形式を取ります。 "<i>number_of_features_plus_arguments feature1 ...</i>".</p> <p>features に使用できる値:</p> <p>queue_if_no_path: no_path_retry を queue に設定するのと同じです。この機能の使用時に発生する可能性がある問題については「queue_if_no_path 機能に関する問題」を参照してください。</p> <p>retain_attached_hw_handler: このパラメーターを yes に設定したとき既に SCSI 層によりハードウェアハンドラーがパスデバイスに接続されていた場合は、multipath.conf ファイルで指定している hardware_handler の使用をデバイスに強制しません。SCSI 層によるハードウェアハンドラーの接続が行われていない場合は、通常通り、マルチパスで設定されているハードウェアハンドラーの使用が継続されます。デフォルト値は no です。</p> <p>pg_init_retries <i>n</i>: 失敗するまでパスグループの初期化を最高 <i>n</i> 回再試行します。ここで、$1 \leq n \leq 50$ です。</p> <p>pg_init_delay_msecs <i>n</i>: パスグループの初期化の再試行の間隔として <i>n</i> ミリ秒待機します。ここで、$0 \leq n \leq 60000$ です。</p>
path_checker	<p>パスの状態を判断するのに使用されるデフォルトのメソッドを指定します。使用できる値を以下に示します。</p> <p>readsector0: デバイスの最初のセクターを読み取ります。</p> <p>tur: デバイスに対して TEST UNIT READY コマンドを発行します。</p> <p>emc_clariion: EMC Clariion 固有のEVPDページ 0xC0 の問い合わせを行ってパスを特定します。</p> <p>hp_sw: Active/Standby のファームウェアを搭載した HP ストレージアレイのパスの状態をチェックします。</p> <p>rdac: LSI/Engenio RDAC ストレージコントローラーのパスの状態をチェックします。</p> <p>directio: 直接 I/O を使用する最初のセクターを読み取ります。</p> <p>デフォルト値は directio です。</p>

属性	説明
failback	<p>パスグループのフェイルバックを管理します。</p> <p>immediate の値を設定すると、active パスを含んでいる最も優先度の高いパスグループに直ちにフェイルバックします。</p> <p>manual の値を指定すると、直ちにフェイルバックはさせずオペレーター介入がある場合のみにフェイルバックします。</p> <p>followover の値を指定すると、パスグループの最初のパスが active になった時点で自動フェイルバックを実行させます。これによって、別のノードがフェイルオーバーを要求している場合にノードが自動的にフェイルバックしないようにします。</p> <p>「0」以上の数値を使ってフェイルバックの秒単位の遅延を指定します。</p> <p>デフォルト値は manual です。</p>
rr_min_io	<p>現在のパスグループ内の次のパスに切り替えるまでにルーティングする I/O 要求数を指定します。この設定は 2.6.31 より旧式のカーネルを実行しているシステム専用になります。これより新しいシステムには rr_min_io_rq を使用してください。デフォルト値は 1000 です。</p>
rr_min_io_rq	<p>現在のパスグループ内の次のパスに切り替えるまでにルーティングする I/O 要求数を指定します。要求ベースの device-mapper-multipath を使用しています。最近のカーネルを実行しているシステムで使用してください。2.6.31 より旧式のカーネルを実行しているシステムの場合は rr_min_io を使用してください。デフォルト値は 1 です。</p>
rr_weight	<p>priorities に設定すると、次のパス選択のため selector を呼び出すまでに送信する要求数に rr_min_io を使用せず、prio 関数で決定されるパスの優先度に rr_min_io をかけることで送信する要求数を求めます。uniform に設定すると全てのパスの重みは等しくなります。デフォルト値は uniform です。</p>
no_path_retry	<p>この属性の数値でキュー待ちを無効にするまでに障害の発生したパスの使用を試行させる回数を指定します。</p> <p>fail の値を指定するとキュー待ちせず直ちに失敗します。</p> <p>queue の値を指定するとパスが修復されるまでキュー待ちを停止させません。</p> <p>デフォルト値は 0 です。</p>
user_friendly_names	<p>yes に設定すると /etc/multipath/bindings ファイルを使った永続的で固有となるエイリアスが mpathn の形式でマルチパスに割り当てられます。no に設定すると WWID がマルチパスのエイリアスに使用されます。いずれの設定であっても設定ファイルの multipaths セクションで指定されるデバイス固有のエイリアスの方が優先されることとなります。デフォルト値は no です。</p>
queue_without_daemon	<p>no に設定すると multipathd デーモンによりシャットダウン時にすべてのデバイスのキュー待ちが無効にされます。デフォルト値は no です。</p>
flush_on_last_del	<p>yes に設定すると multipathd デーモンによりデバイスへの最後のパスが削除された時点でキュー待ちが無効にされます。デフォルト値は no です。</p>

属性	説明
max_fds	マルチパスおよび multipathd デーモンで開くことが可能なオープンファイル記述子の最大数をセットします。これは、 ulimit -n コマンドに相当します。Red Hat Enterprise Linux 6.3 リリース以降、デフォルト値は max になっています。つまり、 /proc/sys/fs/nr_open に記述されたシステムの上限に設定されます。これより旧式のリリースでは、値を設定しないとオープンファイル記述子の最大数は呼び出しのプロセスによって取得され、通常は 1024 です。この数が 1024 を超える場合は、念のためパスの最大数に 32 を加えた数に設定してください。
checker_timeout	明示的なタイムアウトで SCSI コマンドを発行するパス checker および prioritizer に使用する秒単位のタイムアウトです。デフォルト値は sys/block/sdx/device/timeout から取得します。
fast_io_fail_tmo	FC リモートポート上における問題検出後に、そのリモートポート上でデバイスへの I/O に失敗するまでの SCSI レイヤーの待機時間の秒数です。この値は dev_loss_tmo よりも小さくすることをお勧めします。これを off にセットすると、タイムアウトが無効になります。デフォルト値は OS により決定します。
dev_loss_tmo	FC リモートポート上における問題検出後に、それがシステムから削除されるまでの SCSI レイヤーの待機時間の秒数です。無限にセットするには 2147483647 秒または 68 年にセットします。デフォルト値は OS により決定します。
hwttable_string_match	multipath.conf ファイルの devices セクション内の各デバイス設定により独自のデバイス設定を作成するか、組み込みデバイス設定のいずれか一つを修正します。 hwttable_string_match を yes に設定すると、ユーザーのデバイス設定内のベンダー、製品、およびリビジョンの文字列が組み込みのデバイス設定の文字列と完全に一致する場合には組み込みの設定がユーザー設定オプションで修正されます。完全一致しない場合はユーザーのデバイス設定は新しい設定として処理されます。 hwttable_string_match を no に設定すると文字列による一致ではなく正規表現による一致が使用されます。 hwttable_string_match はデフォルトでは no に設定されます。
retain_attached_hw_handler	このパラメーターを yes に設定したとき既に SCSI 層によりハードウェアハンドラーがパスデバイスに接続されていた場合は、 multipath.conf ファイルで指定している hardware_handler の使用をデバイスに強制しません。SCSI 層によるハードウェアハンドラーの接続が行われていない場合は、通常通り、マルチパスで設定されているハードウェアハンドラーの使用が継続されます。デフォルト値は no です。
detect_prio	yes に設定すると最初にデバイスで ALUA に対応しているかをチェックします。対応している場合は alua prioritizer をデバイスに割り当てます。デバイスが ALUA に対応していない場合は通常通り prioritizer の特定を行います。デフォルト値は no です。
reload_readwrite	yes に設定すると multipathd デーモンはパスデバイスの変更イベントをリッスンし、デバイスが読み取り/書き込みになるとマルチパスデバイスの再ロードを行います。
uid_attribute	固有のパス識別子を与えます。デフォルト値は ID_SERIAL です。
force_sync	「yes」に設定するとパスチェッカーによる非同期モードでの実行を阻止します。

4.4. 設定ファイルの **multipaths** セクション

multipath.conf 設定ファイルの **multipaths** セクションで各特定マルチパスデバイスに対して設定できる属性を [表4.2「Multipathsの属性」](#) に示します。属性が適用されるのは指定したマルチパスデバイスのみです。デフォルト値がDM-Multipathによって使用され、**multipath.conf** ファイルの **defaults** と **devices** セクションに設定した属性より優先されます。

表4.2 Multipathsの属性

属性	説明
wwid	multipath 属性を適用するマルチパスデバイスの WWID を指定します。このパラメーターは、 multipath.conf ファイルのこのセクションに必須となります。
alias	multipath 属性が適用されるマルチパスデバイスのシンボリック名を指定します。 user_friendly_names を使用する場合には、この値を mpathn に設定しないでください。自動的に割り当てられる user_friendly_names と競合してしまうため誤ったデバイスノード名が与えられる可能性があります。
path_grouping_policy	未指定のマルチパスに適用するデフォルトのパスグループリングポリシーを指定します。使用できる値を以下に示します。 failover = 優先グループ毎に1つのパス multibus = 1つの優先グループ内の全パスが有効 group_by_serial = 検出されたシリアル番号毎に1つの優先グループ group_by_prio = パス優先値毎に1つの優先グループ group_by_node_name = ターゲットノード名毎に1つの優先グループ
path_selector	次の I/O 動作に使用するパスの決定に使うデフォルトのアルゴリズムを指定します。使用できる値を以下に示します。 round-robin 0 : パスグループ内の全てのパスをループスルーしそれぞれに同負荷の I/O を送ります。 queue-length 0 : 未処理の I/O 要求数が最も少ないパスに次の I/O 群を送ります。 service-time 0 : 推定処理時間が最も短いパスに次の I/O 群を送ります。推定処理時間は各パスに対する未処理の I/O の合計サイズをその相対スループットで割ることにより求められます。

属性	説明
failback	<p>パスグループのフェイルバックを管理します。</p> <p>immediate の値を設定すると、active パスを含んでいる最も優先度の高いパスグループに直ちにフェイルバックします。</p> <p>manual の値を指定すると、直ちにフェイルバックはさせずオペレーター介入がある場合のみにフェイルバックします。</p> <p>followover の値を指定すると、パスグループの最初のパスが active になった時点で自動フェイルバックを実行させます。これによって、別のノードがフェイルオーバーを要求している場合にノードが自動的にフェイルバックしないようにします。</p> <p>「0」以上の数値を使ってフェイルバックの秒単位の遅延を指定します。</p>
prio	<p>パスの優先値を得るため呼び出すデフォルトの関数を指定します。例えば、SPC-3 の ALUA ビットは悪用可能な prio 値を提供します。使用できる値を以下に示します。</p> <p>const: 全てのパスに優先度 1 をセットします。</p> <p>emc: EMC アレイのパス優先度を生成します。</p> <p>alua: SCSI-3 ALUA セットアップに基づいて、パスの優先度を生成します。</p> <p>ontap: NetApp アレイのパスの優先度を生成します。</p> <p>rdac: LSI/Engenio RDAC コントローラーのパスの優先度を生成します。</p> <p>hp_sw: active/standby モードの Compaq/HP コントローラー用のパスの優先度を生成します。</p> <p>hds: Hitachi HDS Modular ストレージアレイのパスの優先度を生成します。</p>
no_path_retry	<p>この属性の数値でキュー待ちを無効にするまでに障害の発生したパスの使用を試行させる回数を指定します。</p> <p>fail の値を指定するとキュー待ちせず直ちに失敗します。</p> <p>queue の値を指定するとパスが修復されるまでキュー待ちを停止させません。</p>
rr_min_io	<p>現在のパスグループ内の次のパスに切り替えるまでにルーティングする I/O 要求数を指定します。この設定は 2.6.31 より旧式のカーネルを実行しているシステム専用です。これより新しいシステムには rr_min_io_rq を使用してください。デフォルト値は 1000 です。</p>
rr_min_io_rq	<p>現在のパスグループ内の次のパスに切り替えるまでにルーティングする I/O 要求数を指定します。要求ベースの device-mapper-multipath を使用しています。最近のカーネルを実行しているシステムで使用してください。2.6.31 より旧式のカーネルを実行しているシステムの場合は rr_min_io を使用してください。デフォルト値は 1 です。</p>

属性	説明
rr_weight	priorities に設定すると、次のパス選択のため path_selector を呼び出すまでに送信する要求数に rr_min_io を使用せず、 prio 関数で決定されるパスの優先度に rr_min_io をかけることで送信する要求数を求めます。 uniform に設定すると全てのパスの重みは等しくなります。
flush_on_last_del	yes に設定するとデバイスへの最後のパスが削除された時点でキュー待ちを無効にします。
user_friendly_names	yes に設定すると /etc/multipath/bindings ファイルを使った永続的で固有となるエイリアスが mpathn の形式でマルチパスに割り当てられます。 no に設定すると WWID がマルチパスのエイリアスに使用されます。いずれの設定であっても設定ファイルの multipaths セクションで指定されるデバイス固有のエイリアスの方が優先されることとなります。

設定ファイルで2つの特定マルチパスデバイスに対して **multipath** 属性を指定している例を以下に示します。1番目のデバイスの WWID は **3600508b4000156d70001200000b0000** で **yellow** のシンボリック名を持たせています。

2番目のマルチパスデバイスの WWID は **1DEC_____321816758474** で **red** のシンボリック名を持たせています。この例では **rr_weight** 属性に **priorities** を設定しています。

```

multipaths {
    multipath {
        wwid                3600508b4000156d70001200000b0000
        alias                yellow
        path_grouping_policy multibus
        path_selector        "round-robin 0"
        failback              manual
        rr_weight             priorities
        no_path_retry        5
    }
    multipath {
        wwid                1DEC_____321816758474
        alias                red
        rr_weight             priorities
    }
}

```

4.5. 設定ファイルの **devices** セクション

multipath.conf 設定ファイルの **devices** セクションで個別ストレージデバイスごとに設定できる属性を [表4.3「devicesの属性」](#) に示します。そのデバイスを含むパスの **multipath.conf** ファイルの **multipaths** セクションに指定されている属性で上書きされない限りこの属性が使用されます。この属性は **multipath.conf** ファイルの **defaults** セクションに設定されている属性より優先されます。

マルチパスに対応しているデバイスの多くはデフォルトでマルチパス設定に含まれています。対応デバイスなどデフォルトの設定値については以下のいずれかのコマンドを実行して確認してください。

```

# multipathd show config
# multipath -t

```

こうしたデバイスの値はおそらく修正の必要はありませんが、必要が生じた場合は設定ファイルにエンターリーを含ませてデフォルト値を上書きすることができます。**multipathd show config** コマンドで表示される目的デバイスのデフォルト値をコピーして変更したい値に書き換えることができます。

設定ファイルのこのセクションにデフォルトでは自動的に設定されないデバイスを追加する場合は **vendor** と **product** のパラメーターを設定する必要があります。値は **/sys/block/device_name/device/vendor** と **/sys/block/device_name>/device/model** で検索できます。device_name はマルチパス設定対象のデバイスです。以下に例を示します。

```
# cat /sys/block/sda/device/vendor
WINSYS
# cat /sys/block/sda/device/model
SF2372
```

追加で指定するパラメーターは使用するデバイスによって異なります。デバイスが active/active なら通常、追加でパラメーターを設定する必要はありません。**path_grouping_policy** には **multibus** に設定する必要性が生じることがあるかもしれません。これ以外で設定の必要性が生じる可能性のあるパラメーターとして **no_path_retry** や **rr_min_io** があります。これらについては [表4.3 「devices の属性」](#) で説明しています。

デバイスが active/passive であるのに I/O を持つパスが自動的に passive パスに切り替えられる場合は、チェッカーの関数を変更して I/O をパスに送信しない関数にし、それが正しく動作するか検証する必要があります (これを行わないとデバイスはフェイルオーバーし続けます)。つまり、ほとんどの場合 **path_checker** を **tur** に設定するということになります。この設定は Test Unit Ready コマンドに対応するすべての SCSI デバイスで機能します。Test Unit Ready コマンドは大半のデバイスで対応しています。

パスの切り替えに特殊なコマンドを必要とするデバイスの場合、マルチパスを設定するにはハードウェアハンドラーカーネルモジュールが必要になります。現在、利用可能なハードウェアハンドラーは **emc** です。このハンドラーが目的のデバイスに使用できない場合はそのデバイスへのマルチパス設定はできない可能性があります。

表4.3 devices の属性

属性	説明
vendor	device 属性が適用されるストレージデバイスのベンダー名 (例: COMPAQ) を指定します。
product	device 属性が適用されるストレージデバイスの製品名 (例: HSV110 (C)COMPAQ) を指定します。
revision	ストレージデバイスの製品リビジョン識別子を指定します。
product_blacklist	製品別にデバイスをブラックリスト指定するため使用する正規表現を指定します。
hardware_handler	パスグループの切り替えや I/O エラーの処理時にハードウェア固有の動作を行わせるため使用するモジュールを指定します。使用できる値を以下に示します。 1 emc: EMC ストレージアレイ用のハードウェアハンドラーです。 1 alua: SCSI-3 ALUA アレイ用のハードウェアハンドラーです。 1 hp_sw: Compaq/HP コントローラー用のハードウェアハンドラーです。 1 rdac: LSI/Engenio RDAC コントローラー用のハードウェアハンドラーです。

属性	説明
path_grouping_policy	<p>未指定のマルチパスに適用するデフォルトのパスグループポリシーを指定します。使用できる値を以下に示します。</p> <p>failover = 優先グループ毎に1つのパス</p> <p>multibus = 1つの優先グループ内の全パスが有効</p> <p>group_by_serial = 検出されたシリアル番号毎に1つの優先グループ</p> <p>group_by_prio = パス優先値毎に1つの優先グループ</p> <p>group_by_node_name = ターゲットノード名毎に1つの優先グループ</p>
path_selector	<p>次の I/O 動作に使用するパスの決定に使うデフォルトのアルゴリズムを指定します。使用できる値を以下に示します。</p> <p>round-robin 0: パスグループ内の全てのパスをループスルーしそれぞれに同負荷の I/O を送ります。</p> <p>queue-length 0: 未処理の I/O 要求数が最も少ないパスに次の I/O 群を送ります。</p> <p>service-time 0: 推定処理時間が最も短いパスに次の I/O 群を送ります。推定処理時間は各パスに対する未処理の I/O の合計サイズをその相対スループットで割ることにより求められます。</p>
path_checker	<p>パスの状態を判断するのに使用されるデフォルトのメソッドを指定します。使用できる値を以下に示します。</p> <p>readsector0: デバイスの最初のセクターを読み取ります。</p> <p>tur: デバイスに対して TEST UNIT READY を発行します。</p> <p>emc_clariion: EMC Clariion 固有のEVPDページ 0xC0 の問い合わせを行ってパスを特定します。</p> <p>hp_sw: Active/Standby のファームウェアを搭載した HP ストレージアレイのパスの状態をチェックします。</p> <p>rdac: LSI/Engenio RDAC ストレージコントローラーのパスの状態をチェックします。</p> <p>directio: 直接 I/O を使用する最初のセクターを読み取ります。</p>

属性	説明
features	<p>マルチパスデバイスの追加デフォルト features です。次の形式を取ります。"<i>number_of_features_plus_arguments feature1 ...</i>".</p> <p>features に使用できる値:</p> <p>queue_if_no_path: no_path_retry を queue に設定するのと同じです。この機能の使用時に発生する可能性がある問題については「queue_if_no_path 機能に関する問題」を参照してください。</p> <p>retain_attached_hw_handler: このパラメーターを yes に設定したとき既に SCSI 層によりハードウェアハンドラーがパスデバイスに接続されていた場合は、multipath.conf ファイルで指定している hardware_handler の使用をデバイスに強制しません。SCSI 層によるハードウェアハンドラーの接続が行われていない場合は、通常通り、マルチパスで設定されているハードウェアハンドラーの使用が継続されます。デフォルト値は no です。</p> <p>pg_init_retries <i>n</i>: 失敗するまでパスグループの初期化を最高 <i>n</i> 回再試行します。ここで、$1 \leq n \leq 50$ です。</p> <p>pg_init_delay_msecs <i>n</i>: パスグループの初期化の再試行の間隔として <i>n</i> ミリ秒待機します。ここで、$0 \leq n \leq 60000$ です。</p>
prio	<p>パスの優先値を得るため呼び出すデフォルトの関数を指定します。例えば、SPC-3 の ALUA ビットは悪用可能な prio 値を提供します。使用できる値を以下に示します。</p> <p>const: 全てのパスに優先度 1 をセットします。</p> <p>emc: EMC アレイのパス優先度を生成します。</p> <p>alua: SCSI-3 ALUA セットアップに基づいて、パスの優先度を生成します。</p> <p>ontap: NetApp アレイのパスの優先度を生成します。</p> <p>rdac: LSI/Engenio RDAC コントローラーのパスの優先度を生成します。</p> <p>hp_sw: active/standby モードの Compaq/HP コントローラー用のパスの優先度を生成します。</p> <p>hds: Hitachi HDS Modular ストレージアレイのパスの優先度を生成します。</p>

属性	説明
failback	<p>パスグループのフェイルバックを管理します。</p> <p>immediate の値を設定すると、active パスを含んでいる最も優先度の高いパスグループに直ちにフェイルバックします。</p> <p>manual の値を指定すると、直ちにフェイルバックはさせずオペレーター介入がある場合のみにフェイルバックします。</p> <p>followover の値を指定すると、パスグループの最初のパスが active になった時点で自動フェイルバックを実行させます。これによって、別のノードがフェイルオーバーを要求している場合にノードが自動的にフェイルバックしないようにします。</p> <p>「0」以上の数値を使ってフェイルバックの秒単位の遅延を指定します。</p>
rr_weight	<p>priorities に設定すると、次のパス選択のため path_selector を呼び出すまでに送信する要求数に rr_min_io を使用せず、prio 関数で決定されるパスの優先度に rr_min_io をかけることで送信する要求数を求めます。uniform に設定すると全てのパスの重みは等しくなります。</p>
no_path_retry	<p>この属性の数値でキュー待ちを無効にするまでに障害の発生したパスの使用を試行させる回数を指定します。</p> <p>fail の値を指定するとキュー待ちせず直ちに失敗します。</p> <p>queue の値を指定するとパスが修復されるまでキュー待ちを停止させません。</p>
rr_min_io	<p>現在のパスグループ内の次のパスに切り替えるまでにルーティングする I/O 要求数を指定します。この設定は 2.6.31 より旧式のカーネルを実行しているシステム専用です。これより新しいシステムには rr_min_io_rq を使用してください。デフォルト値は 1000 です。</p>
rr_min_io_rq	<p>現在のパスグループ内の次のパスに切り替えるまでにルーティングする I/O 要求数を指定します。要求ベースの device-mapper-multipath を使用しています。最近のカーネルを実行しているシステムで使用してください。2.6.31 より旧式のカーネルを実行しているシステムの場合は rr_min_io を使用してください。デフォルト値は 1 です。</p>
fast_io_fail_tmo	<p>FC リモートポート上における問題検出後に、そのリモートポート上でデバイスへの I/O に失敗するまでの SCSI レイヤーの待機時間の秒数です。この値は、dev_loss_tmo よりも小さくすることをお勧めします。これを off にセットすると、タイムアウトが無効になります。</p>
dev_loss_tmo	<p>FC リモートポート上での問題検出後に、システムから削除されるまでの SCSI レイヤーの待機時間 (秒) です。この値を制限なしに設定するには 2147483647 秒または 68 年にセットします。</p>
flush_on_last_del	<p>yes に設定すると multipathd デモンによりデバイスへの最後のパスが削除された時点でキュー待ちが無効にされます。</p>
user_friendly_names	<p>yes に設定すると /etc/multipath/bindings ファイルを使った永続的で固有となるエイリアスが mpathn の形式でマルチパスに割り当てられます。no に設定すると WWID がマルチパスのエイリアスに使用されます。いずれの設定であっても設定ファイルの multipaths セクションで指定されるデバイス固有のエイリアスの方が優先されることとなります。デフォルト値は no です。</p>

属性	説明
retain_attached_hw_handler	このパラメーターを yes に設定したとき既に SCSI 層によりハードウェアハンドラーがパスデバイスに接続されていた場合は、 multipath.conf ファイルで指定している hardware_handler の使用をデバイスに強制しません。SCSI 層によるハードウェアハンドラーの接続が行われていない場合は、通常通り、マルチパスで設定されているハードウェアハンドラーの使用が継続されます。デフォルト値は no です。
detect_prio	yes に設定すると最初にデバイスで ALUA に対応しているかをチェックします。対応している場合は alua prioritizer をデバイスに割り当てます。デバイスが ALUA に対応していない場合は通常通り prioritizer の特定を行います。

マルチパス設定ファイルの **device** エントリーの例を以下に示します。

```
# }
# device {
#   vendor    "COMPAQ  "
#   product   "MSA1000      "
#   path_grouping_policy multibus
#   path_checker tur
#   rr_weight priorities
# }
#}
```

第5章 DM Multipath の管理とトラブルシューティング

この章では稼働中のシステムで DM-Multipath を管理する方法について説明しています。以下のようなトピックで構成しています。

- ※ オンラインのマルチパスデバイスのサイズ変更
- ※ root デバイスをシングルパスデバイスからマルチパスデバイスへ移行
- ※ swap デバイスをシングルパスデバイスからマルチパスデバイスへ移行
- ※ マルチパスデーモン
- ※ 多数の LUN に伴う問題
- ※ `queue_if_no_path` 機能に関する問題
- ※ `multipath` コマンドの出力
- ※ `multipath` コマンドを使ったマルチパスクエリー
- ※ `multipath` コマンドのオプション
- ※ `dmsetup` コマンドを使ったマルチパスクエリー
- ※ `multipathd` インタラクティブコンソールに関するトラブルシューティング

5.1. オンラインのマルチパスデバイスのサイズ変更

オンラインのマルチパスデバイスのサイズを変更する必要がある場合は以下の手順に従ってください。

1. 物理デバイスのサイズを変更します。
2. 以下のコマンドを実行して LUN へのパスを検索します。

```
# multipath -l
```

3. パスのサイズを変更します。SCSI デバイスの場合、デバイスの `rescan` ファイルに 1 と書き込むと SCSI ドライバーによる再スキャンが行われます。以下にコマンドの例を示します。

```
# echo 1 > /sys/block/device_name/device/rescan
```

4. `multipathd resize` コマンドを実行してマルチパスデバイスのサイズを変更します。

```
# multipathd -k'resize map mpatha'
```

5. ファイルシステムのサイズを変更します (LVM または DOS のパーティションが使用されていないことを前提とします)。

```
# resize2fs /dev/mapper/mpatha
```

5.2. root ファイルシステムをシングルパスデバイスからマルチパスデバイスへ移行

シングルパスのデバイスにシステムをインストールした後、後日、別のパスを root ファイルシステムに追加する場合、root ファイルシステムをマルチパスのデバイスに移行する必要があります。本セクションではシングルパスからマルチパスのデバイスへの移行手順を説明します。

device-mapper-multipath パッケージがインストールされていることを確認した上で以下の手順を実行します。

1. 以下のコマンドを実行して **/etc/multipath.conf** 設定ファイルを作成、マルチパスモジュールをロードして **multipathd** の **chkconfig** を **on** に設定します。

```
# mpathconf --enable
```

mpathconf コマンドを使ったマルチパスの設定に関する詳細は「[DM Multipath の設定](#)」を参照してください。

2. **find_multipaths** 設定パラメーターが **yes** に設定されていない場合は、「[設定ファイルの blacklist セクション](#)」の説明に従い **/etc/multipath.conf** ファイルの **blacklist** セクションと **blacklist_exceptions** セクションを編集してください。
3. 検出され次第 root デバイスの上にマルチパスデバイスを構築させるため次のコマンドを実行します。また、このコマンドを実行することでパスが一つしかない場合でも必ずマルチパスデバイスが構築されるようになります。

```
# multipath -a root_devname
```

例えば root デバイスが **/dev/sdb** の場合は次のコマンドを実行します。

```
# multipath -a /dev/sdb  
wwid '3600d02300069c9ce09d41c4ac9c53200' added
```

4. 設定ファイルの設定が正しく行われたか確認するため **multipath** コマンドを実行して次のような行の出力を探します。この行はコマンドがマルチパスデバイスの作成に失敗したことを示しています。

```
date wwid: ignoring map
```

例えばデバイスの WWID が **3600d02300069c9ce09d41c4ac9c53200** なら次のような行が出力に表示されます。

```
# multipath  
Oct 21 09:37:19 | 3600d02300069c9ce09d41c4ac9c53200: ignoring map
```

5. **dracut** コマンドに以下のオプションを付けて実行し、**multipath** 付きの **initramfs** ファイルシステムを再ビルドします。

```
# dracut --force -H --add multipath
```

6. マシンをシャットダウンします。
7. マシンから他のパスが見えるよう FC スイッチを設定します。
8. マシンを起動します。
9. root ファイルシステム (**/**) がマルチパス設定したデバイス上にあることを確認します。

5.3. swap ファイルシステムをシングルパスデバイスからマルチパスデバイスへ移行

デフォルトでは swap デバイスは論理ボリュームとして設定されます。論理ボリュームグループを構成する物理ボリューム上でマルチパスを設定している限り、これらのデバイスをマルチパスデバイスとして設定するのに特別な手順は必要ありません。ただし、swap デバイスが LVM ボリュームではなく、またデバイス名でマウントされる場合には、**fstab** ファイルを編集して適切なマルチパスデバイス名に変更する必要があります。

1. **-v3** オプションを付けて **/sbin/multipath** コマンドを実行し、swap デバイスの WWID 番号を特定します。コマンドの出力のパス一覧に swap デバイスが表示されます。

swap デバイスを示す次のような行のコマンド出力を探します。

```
WWID H:B:T:L devname MAJOR:MINOR
```

例えば、swap ファイルシステムを **sda** かそのパーティションの一つに設定している場合は以下のような行が表示されます。

```
==== paths list ====
...
1ATA      WDC WD800JD-75MSA3          WD-WMAM9F
1:0:0:0 sda 8:0
...
```

2. **/etc/multipath.conf** ファイルに swap デバイスのエイリアスを設定します。

```
multipaths {
    multipath {
        wwid WWID_of_swap_device
        alias swapdev
    }
}
```

3. **/etc/fstab** を編集して root デバイスへの古いデバイスパスをマルチパスデバイスに置き換えます。

例えば **/etc/fstab** ファイル内に以下のようなエントリーがあった場合、

```
/dev/sda2 swap                                ext4    defaults    0 0
```

そのエントリーを以下のように変更します。

```
/dev/mapper/swapdev swap                                ext4    defaults    0 0
```

5.4. マルチパスデーモン

マルチパス設定の実装に問題があることが判明した場合は、[3章DM Multipath の設定](#)に説明しているようにマルチパスデーモンが実行しているか確認してください。マルチパス設定されたデバイスを使用するには **multipathd** デーモンを実行しておく必要があります。

5.5. 多数の LUN に伴う問題

5.5. udev ルールを削除する

多数の LUN が 1 ノードに追加されると、マルチパス設定したデバイスを使用することで **udev** デバイスマネージャーによるデバイスノードの作成時間が大幅に増大する場合があります。この問題が発生した場合は `/etc/udev/rules.d/40-multipath.rules` の以下の行を削除すると修正することができます。

```
KERNEL!="dm-[0-9]*", ACTION=="add", PROGRAM==" /bin/bash -c '/sbin/lsmode | /bin/grep ^dm_multipath'", RUN+=" /sbin/multipath -v0 %M:%m"
```

ブロックデバイスがノードに追加される度に **udev** デバイスマネージャーが **multipath** を実行する原因となるのがこの行です。この行を削除しても、引き続き **multipathd** デーモンによりマルチパス設定したデバイスが自動的に作成され、**multipath** がマルチパス設定した root ファイルシステムでノードの起動中に呼び出されます。唯一の変化は、**multipathd** デーモンが実行されていない時はマルチパス設定したデバイスが自動的に作成されないという点です。これで大多数のマルチパスユーザーには問題となくなるとはなりません。

5.6. queue_if_no_path 機能に関する問題

features "1 queue_if_no_path" を `/etc/multipath.conf` ファイルに指定すると 1パスまたはそれ以上のパスが復帰するまで I/O を発行するすべてのプロセスがハングします。これを回避するため `/etc/multipath.conf` ファイルに **no_path_retry N** パラメーターを設定します (**N** はシステムにパスを再試行させる回数)。

features "1 queue_if_no_path" オプションを使用しなければならない状況で上記の問題が発生した場合は、**dmsetup** コマンドを使って特定の LUN (全てのパスが利用できないもの) 用のランタイムポリシーを編集します。例えば、マルチパスデバイス **mpathc** のポリシーを **"queue_if_no_path"** から **"fail_if_no_path"** に変更する場合は以下のコマンドを実行します。

```
dmsetup message mpathc 0 "fail_if_no_path"
```

指定する必要があるのは、パスではなく **mpathn** のエイリアスであることに注意してください。

5.7. Multipath コマンドの出力

マルチパスデバイスの作成、修正、一覧表示を行う際、現在のデバイス設定が表示されます。形式を以下に示します。

各マルチパスデバイス

```
action_if_any: alias (wwid_if_different_from_alias)
dm_device_name_if_known vendor,product size=size features='features'
hwhandler='hardware_handler' wp=write_permission_if_known
```

各パスグループ

```
-+- policy='scheduling_policy' prio=prio_if_known
status=path_group_status_if_known
```

各パス

```
`- host:channel:id:lun devnode major:minor dm_status_if_known
path_status online_status
```

マルチパスコマンドの出力例を以下に示します。

```
3600d0230000000000e13955cc3757800 dm-1 WINSYS,SF2372
size=269G features='0' hwhandler='0' wp=rw
|+- policy='round-robin 0' prio=1 status=active
| ` 6:0:0:0 sdb 8:16 active ready running
`-+- policy='round-robin 0' prio=1 status=enabled
   ` 7:0:0:0 sdf 8:80 active ready running
```

パスの準備が整うとパスの状態が **ready** または **ghost** になります。パスに障害が発生すると状態は **faulty** または **shaky** になります。パスの状態は `/etc/multipath.conf` ファイルに定義されているポーリング間隔に応じ `multipathd` デーモンによって定期的に更新されます。

dm の状態はパスの状態と似ていますが、カーネルの観点からは異なります。dm の状態は **failed** (**faulty** に似ている) とそれ以外の状態を示す **active** の 2 種類があります。任意のデバイスのパス状態と dm 状態が一時的に一致しない場合があります。

online_status に使用できる値は **running** と **offline** です。**offline** の状態とはこの SCSI デバイスが無効になっているということです。



注記

マルチパスデバイスの作成中または修正中の場合は、パスグループの状態、dm のデバイス名、書き込みパーミッション、dm の状態などは不明です。また、機能も常に正しいとは限りません。

5.8. multipath コマンドを使ったマルチパスクエリー

`multipath` コマンドで **-l** と **-ll** のオプションを使用すると現在のマルチパス設定を表示することができます。**-l** オプションは `sysfs` 内の情報から収集したマルチパストポロジーとデバイスマップを表示します。**-ll** オプションは **-l** オプションが表示する情報の他、使用できる他のすべてのシステムコンポーネントを表示します。

マルチパス設定を表示する際に `multipath` コマンドの **-v** オプションで指定できる詳細レベルが 3 レベルあります。**-v0** は出力なしです。**-v1** は作成または更新されたマルチパス名のみを出力します。マルチパス名は `kpartx` など他のツールに与えることができます。**-v2** は検知されたすべてのパス、マルチパス、デバイスマップを表示します。

`multipath -l` コマンドの出力を以下に示します。

```
# multipath -l
3600d0230000000000e13955cc3757800 dm-1 WINSYS,SF2372
size=269G features='0' hwhandler='0' wp=rw
|+- policy='round-robin 0' prio=1 status=active
| ` 6:0:0:0 sdb 8:16 active ready running
`-+- policy='round-robin 0' prio=1 status=enabled
   ` 7:0:0:0 sdf 8:80 active ready running
```

`multipath -ll` コマンドの出力を以下に示します。

```
# multipath -ll
3600d0230000000000e13955cc3757801 dm-10 WINSYS,SF2372
size=269G features='0' hwhandler='0' wp=rw
|+- policy='round-robin 0' prio=1 status=enabled
```

```
| `- 19:0:0:1 sdc 8:32 active ready running
`-+- policy='round-robin 0' prio=1 status=enabled
  `- 18:0:0:1 sdh 8:112 active ready running
3600d0230000000000e13955cc3757803 dm-2 WINSYS,SF2372
size=125G features='0' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
  |- 19:0:0:3 sde 8:64 active ready running
  `- 18:0:0:3 sdj 8:144 active ready running
```

5.9. multipath コマンドのオプション

役立つと思われる `multipath` コマンドのオプションをいくつか[表5.1「役立つ multipath コマンドのオプション」](#)に示します。

表5.1 役立つ `multipath` コマンドのオプション

オプション	説明
<code>-l</code>	<code>sysfs</code> から収集した現在のマルチパス設定とデバイスマッパーを表示します。
<code>-ll</code>	<code>sysfs</code> から収集した現在のマルチパス設定、デバイスマッパー、使用可能な全システムコンポーネントを表示します。
<code>-f device</code>	指定されたマルチパスデバイスを削除します。
<code>-F</code>	不要な全てのマルチパスデバイスを削除します。
<code>-w device</code>	<code>wwids</code> ファイルから指定されたデバイスの <code>wwid</code> を削除します。
<code>-W</code>	現在のマルチパスデバイスのみを組み込むように <code>wwids</code> ファイルをリセットします。

一部の `multipathd` コマンドには後にワイルドカードを付けた `format` オプションを含むものがあります。次のコマンドで使用できるワイルドカード一覧を表示させることができます。

```
# multipathd show wildcards
```

5.10. dmsetup コマンドを使ってデバイスマッパーのエントリーを特定する

`dmsetup` コマンドを使ってマルチパス設定したデバイスに一致するデバイスマッパーのエントリーを検索することができます。

以下のコマンドでは全てのデバイスマッパーデバイスとそのメジャー番号、マイナー番号を表示します。dm デバイス名はマイナー番号で特定することができます。例えば、マイナー番号 3 はマルチパス設定したデバイス `/dev/dm-3` になります。

```
# dmsetup ls
mpathd (253:4)
mpathep1 (253:12)
mpathfp1 (253:11)
mpathb (253:3)
mpathgp1 (253:14)
mpathhp1 (253:13)
mpatha (253:2)
mpathh (253:9)
mpathg (253:8)
VolGroup00-LogVol01 (253:1)
```

```
mpathf (253:7)
VolGroup00-LogVol100 (253:0)
mpathe (253:6)
mpathbp1 (253:10)
mpathd (253:5)
```

5.11. 対話式コンソール `multipathd` を使用してトラブルシューティングを行う

`multipathd -k` コマンドは `multipathd` デーモンに対する対話式のインターフェースです。このコマンドを実行すると対話式のマルチパスコンソールが立ち上がります。このコマンドの実行後に `help` を入力すると使用可能なコマンドの一覧が表示され、対話式コマンドを入力することができます。または `CTRL-D` を入力すると終了します。

`multipathd` 対話式コンソールを使用するとシステムで発生している問題のトラブルシューティングを行うことができます。例えば、以下のようにコマンドを順番に入力するとデフォルト値などのマルチパス設定を表示して、その後コンソールを終了します。

```
# multipathd -k
> > show config
> > CTRL-D
```

次の順序でコマンドを実行し `multipath.conf` に対する変更がすべて反映されるようになります。

```
# multipathd -k
> > reconfigure
> > CTRL-D
```

次の順序でコマンドを実行しパスチェッカーが正しく動作していることを確認します。

```
# multipathd -k
> > show paths
> > CTRL-D
```

付録A 改訂履歴

改訂 0.2-7.3	Thu Aug 6 2015	Noriko Mizumoto
「マルチパスデバイス」と「DM Multipath の設定」の章での一部翻訳の明確化を行うため更新		
改訂 0.2-7.2	Wed Jul 29 2015	Noriko Mizumoto
翻訳および査読完了		
改訂 0.2-7.1	Wed Jul 29 2015	Noriko Mizumoto
翻訳ファイルを XML ソースバージョン 0.2-7 と同期		
改訂 0.2-7	Mon Feb 16 2015	Steven Levine
7.1 GA リリースバージョン		
改訂 0.2-6	Thu Dec 11 2014	Steven Levine
7.1 Beta リリースバージョン		
改訂 0.2-2	Tue Dec 2 2014	Steven Levine
修正: #1162514、#1085979、#1129897 技術的な修正と明確化のため更新		
修正: #1117037 <code>multipath</code> コマンドの <code>-w</code> と <code>-W</code> のオプションに関して記載		
修正: #1117043 features 設定パラメーターの新しいオプションに関して記載		
修正: #1117499 force_sync 設定パラメーターに関して記載		
改訂 0.2-1	Thu Oct 30 2014	Steven Levine
7.0 向け root ファイルシステムの移行方法に関する手順を更新するためビルド		
改訂 0.2-0	Wed Oct 22 2014	Steven Levine
root ファイルシステムの移行方法に関する手順を更新		
改訂 0.1-22	Mon Jun 2 2014	Steven Levine
7.0 GA リリース向けバージョン		
改訂 0.1-20	Tue May 20 2014	Steven Levine
スタイル変更のため再ビルド		
修正: #1069212, #1074858 新しいパラメーターおよび更新されたパラメーターに関して記載		
修正: #1040328 技術的な修正 (マイナー)		
改訂 0.1-10	Wed Apr 9 2014	Steven Levine
7.0 Beta 更新		
改訂 0.1-3	Tue Nov 26 2013	Steven Levine

7.0 pre-Beta バージョン

改訂 0.1-2 Tue Nov 12 2013 Steven Levine
 RHEL 6 の言及を削除

改訂 0.1-1 Wed Jan 16 2013 Steven Levine
 Red Hat Enterprise Linux 6 バージョンのドキュメントからブランチ作成

索引

シンボル

`/etc/multipath.conf` パッケージ, [DM Multipath の設定](#)

ストレージアレイ

- 追加する, [ストレージデバイスを設定する](#) [設定ファイルの devices セクション](#)

ストレージアレイの対応, [ストレージアレイの対応](#)

デバイスの

- 追加する, [ストレージデバイスを設定する](#)

デバイス名, [マルチパスデバイス識別子](#)

フェイルオーバー, [DM Multipath の概要](#)

ブラックリスト

- WWID, [WWID でブラックリスト指定する](#)
 - デバイス名, [デバイス名でブラックリスト指定する](#)

マルチパスデバイス, [マルチパスデバイス](#)

- LVM 論理ボリューム, [論理ボリューム内のマルチパスデバイス](#)
 - 論理ボリューム, [論理ボリューム内のマルチパスデバイス](#)

マルチパスデバイスのサイズ変更, [オンラインのマルチパスデバイスのサイズ変更](#)

マルチパスデーモン (multipathd), [マルチパスデーモン](#)

マルチパス設定した root ファイルシステム, [root ファイルシステムをシングルパスデバイスからマルチパスデバイスへ移行](#)

マルチパス設定した swap ファイルシステム, [swap ファイルシステムをシングルパスデバイスからマルチパスデバイスへ移行](#)

ローカルのディスク、無視する, [マルチパスデバイスの作成時にローカルのディスクを無視する](#)

設定

- DM Multipath, [DM Multipath の設定](#)

設定する

- DM Multipath, [DM Multipath の設定](#)

設定ファイル

- alias パラメーター, [設定ファイルの multipaths セクション](#)
 - blacklist, [設定ファイルの blacklist セクション](#)
 - checker_timeout パラメーター, [設定ファイルの defaults セクション](#)

- [dev_loss_tmo](#) パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの devices セクション](#)
- [failback](#) パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの multipaths セクション](#), [設定ファイルの devices セクション](#)
- [fast_io_fail_tmo](#) パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの devices セクション](#)
- [features](#) パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの devices セクション](#)
- [flush_on_last_del](#) パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの multipaths セクション](#), [設定ファイルの devices セクション](#)
- [force_sync](#) パラメーター, [設定ファイルの defaults セクション](#)
- [hardware_handler](#) パラメーター, [設定ファイルの devices セクション](#)
- [hwtable_string_match](#) パラメーター, [設定ファイルの defaults セクション](#)
- [max_fds](#) パラメーター, [設定ファイルの defaults セクション](#)
- [no_path_retry](#) パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの multipaths セクション](#), [設定ファイルの devices セクション](#)
- [path_checker](#) パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの devices セクション](#)
- [path_grouping_policy](#) パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの multipaths セクション](#), [設定ファイルの devices セクション](#)
- [path_selector](#) パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの multipaths セクション](#), [設定ファイルの devices セクション](#)
- [polling-interval](#) パラメーター, [設定ファイルの defaults セクション](#)
- [prio](#) パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの devices セクション](#)
- [product](#) パラメーター, [設定ファイルの devices セクション](#)
- [product_blacklist](#) パラメーター, [設定ファイルの devices セクション](#)
- [queue_without_daemon](#) パラメーター, [設定ファイルの defaults セクション](#)
- [retain_attached_hw_handler](#) パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの multipaths セクション](#)
- [revision](#) パラメーター, [設定ファイルの devices セクション](#)
- [rr_min_io](#) パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの multipaths セクション](#)
- [rr_weight](#) パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの multipaths セクション](#), [設定ファイルの devices セクション](#)
- [udev_dir](#) パラメーター, [設定ファイルの defaults セクション](#)
- [uid_attribute](#) パラメーター, [設定ファイルの defaults セクション](#)
- [user_friendly_names](#) パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの multipaths セクション](#), [設定ファイルの devices セクション](#)
- [vendor](#) パラメーター, [設定ファイルの devices セクション](#)
- [verbosity](#) パラメーター, [設定ファイルの defaults セクション](#)
- [wwid](#) パラメーター, [設定ファイルの multipaths セクション](#)
- 概要, [設定ファイルの概要](#)

A

active/active 設定

- 図, [DM Multipath の概要](#)
- 定義, [DM Multipath の概要](#)

active/passive 設定

- 図, [DM Multipath の概要](#)
- 定義, [DM Multipath の概要](#)

alias パラメーター, [設定ファイルの multipaths セクション](#)

- 設定ファイル, [マルチパスデバイス識別子](#)

B

blacklist

- デバイスタイプ, [デバイスタイプでブラックリスト指定する](#)
- デフォルトのデバイス, [デバイス名でブラックリスト指定する](#)
- 設定ファイル, [設定ファイルの blacklist セクション](#)

blacklist_exceptions セクション

- multipath.conf ファイル, [ブラックリストから除外する](#)

C

checker_timeout パラメーター, [設定ファイルの defaults セクション](#)**configuration file**

- detect_prio パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの multipaths セクション](#)

D

defaults セクション

- multipath.conf ファイル, [設定ファイルの defaults セクション](#)

detect_prio パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの multipaths セクション](#)**dev/mapper** ディレクトリー, [マルチパスデバイス識別子](#)**device-mapper-multipath** パッケージ, [DM Multipath の設定](#)**devices**

- 追加する, [設定ファイルの devices セクション](#)

devices セクション

- multipath.conf ファイル, [設定ファイルの devices セクション](#)

dev_loss_tmo パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの devices セクション](#)**DM Multipath**

- および LVM, [論理ボリューム内のマルチパスデバイス](#)
- コンポーネント, [DM Multipath のコンポーネント](#)
- デバイス, [マルチパスデバイス](#)
- デバイス名, [マルチパスデバイス識別子](#)
- フェイルオーバー, [DM Multipath の概要](#)
- 冗長性, [DM Multipath の概要](#)
- 定義, [Device Mapper を使ったマルチパス設定](#)
- 概要, [DM Multipath の概要](#)
- 設定, [DM Multipath の設定](#)
- 設定、概要, [DM Multipath 設定の概要](#)
- 設定する, [DM Multipath の設定](#)
- 設定ファイル, [DM Multipath 設定ファイル](#)

dm-n デバイス, [マルチパスデバイス識別子](#)**dmsetup** コマンド、デバイスマッパーのエントリーを特定する, [dmsetup コマンドを使ってデバイスマッパーのエントリーを特定する](#)**dm_multipath** カーネルモジュール, [DM Multipath のコンポーネント](#)

F

failback パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの multipaths セクション](#), [設定ファイルの devices セクション](#)

fast_io_fail_tmo パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの devices セクション](#)

features パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの devices セクション](#)

flush_on_last_del パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの multipaths セクション](#), [設定ファイルの devices セクション](#)

H

hardware_handler パラメーター, [設定ファイルの devices セクション](#)

hwtable_string_match parameter, [設定ファイルの defaults セクション](#)

I

initramfs

- マルチパスを起動する, [initramfs ファイルシステムでマルチパスを設定する](#)

K

kpartx コマンド, [DM Multipath のコンポーネント](#)

L

LVM 論理ボリューム

- マルチパスデバイス, [論理ボリューム内のマルチパスデバイス](#)

lvm.conf ファイル, [論理ボリューム内のマルチパスデバイス](#)

M

max_fds パラメーター, [設定ファイルの defaults セクション](#)

mpathconf コマンド, [DM Multipath のコンポーネント](#)

multipath コマンド, [DM Multipath のコンポーネント](#)

multipath コマンドの

- オプション, [multipath コマンドのオプション](#)
- クエリー, [multipath コマンドを使ったマルチパスクエリー](#)
- 出力, [Multipath コマンドの出力](#)

multipath.conf のファイル

- blacklist_exceptions セクション, [ブラックリストから除外する](#)

multipath.conf ファイル, [ストレージレイの対応](#), [DM Multipath 設定ファイル](#)

- defaults セクション, [設定ファイルの defaults セクション](#)
- devices セクション, [設定ファイルの devices セクション](#)
- multipaths セクション, [設定ファイルの multipaths セクション](#)

multipathd

- コマンド, [対話式コンソール multipathd を使用してトラブルシューティングを行う](#)
- 対話式コンソール, [対話式コンソール multipathd を使用してトラブルシューティングを行う](#)

multipathd start コマンド, [DM Multipath の設定](#)

multipathd デーモン, [DM Multipath のコンポーネント](#)

multipaths セクション

- multipath.conf ファイル, [設定ファイルの multipaths セクション](#)

N

[no_path_retry](#) パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの multipaths セクション](#), [設定ファイルの devices セクション](#)

P

[path_checker](#) パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの devices セクション](#)

[path_grouping_policy](#) パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの multipaths セクション](#), [設定ファイルの devices セクション](#)

[path_selector](#) パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの multipaths セクション](#), [設定ファイルの devices セクション](#)

[polling_interval](#) パラメーター, [設定ファイルの defaults セクション](#)

[prio](#) パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの devices セクション](#)

[product](#) パラメーター, [設定ファイルの devices セクション](#)

[product_blacklist](#) パラメーター, [設定ファイルの devices セクション](#)

Q

[queue_without_daemon](#) パラメーター, [設定ファイルの defaults セクション](#)

R

[retain_attached_hw_handler](#) パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの multipaths セクション](#)

[revision](#) パラメーター, [設定ファイルの devices セクション](#)

[root](#) ファイルシステム, [root ファイルシステムをシングルパスデバイスからマルチパスデバイスへ移行](#)

[rr_min_io](#) パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの multipaths セクション](#)

[rr_weight](#) パラメーター, [設定ファイルの defaults セクション](#), [設定ファイルの multipaths セクション](#), [設定ファイルの devices セクション](#)

S

[swap](#) ファイルシステム, [swap ファイルシステムをシングルパスデバイスからマルチパスデバイスへ移行](#)

U

[udev_dir](#) パラメーター, [設定ファイルの defaults セクション](#)

[uid_attribute](#) パラメーター, [設定ファイルの defaults セクション](#)

[user_friendly_names](#) パラメーター, [マルチパスデバイス識別子](#), [設定ファイルの defaults セクション](#), [設定ファイルの multipaths セクション](#), [設定ファイルの devices セクション](#)

V

[vendor](#) パラメーター, [設定ファイルの devices セクション](#)

[verbosity](#) パラメーター, [設定ファイルの defaults セクション](#)

W

World Wide Identifier (WWID), [マルチパスデバイス識別子](#)

[wwid](#) パラメーター, [設定ファイルの multipaths セクション](#)

