

drbdsetup

名前

drbdsetup — DRBD のセットアップツール

指定方法

drbdsetup { *device* } { *disk* } { *lower_dev* } { *meta_data_dev* } { *meta_data_index* } [-d { *size* }] [-e { *err_handler* }] [-f { *fencing_policy* }] [-b]

drbdsetup { *device* } { *net* } { *local_addr* } [:*port*] { *remote_addr* } [:*port*] { *protocol* } [-c { *time* }] [-i { *time* }] [-t { *val* }] [-S { *size* }] [-k { *count* }] [-e { *max_epoch_size* }] [-b { *max_buffers* }] [-m] [-a { *hash_alg* }] [-x { *shared_secret* }] [-A { *asb-0p-policy* }] [-B { *asb-1p-policy* }] [-C { *asb-2p-policy* }] [-D] [-R { *role-resync-conflict-policy* }] [-p { *ping_timeout* }] [-u { *val* }] [-d { *hash_alg* }]

drbdsetup { *device* } { *syncer* } [-a { *dev_minor* }] [-r { *rate* }] [-e { *extents* }] [-v { *hash_alg* }] [-c { *cpu-mask* }]

drbdsetup { *device* } { *disconnect* }

drbdsetup { *device* } { *detach* }

drbdsetup { *device* } { *down* }

drbdsetup { *device* } { *primary* } [-o]

drbdsetup { *device* } { *secondary* }

drbdsetup { *device* } { *verify* }

drbdsetup { *device* } { *invalidate* }

drbdsetup { *device* } { *invalidate-remote* }

drbdsetup { *device* } { *wait-connect* } [-t { *wfc_timeout* }] [-d { *degr_wfc_timeout* }] [-w]

drbdsetup { *device* } { *wait-sync* } [-t { *wfc_timeout* }] [-d { *degr_wfc_timeout* }] [-w]

drbdsetup { *device* } {state}

drbdsetup { *device* } {cstate}

drbdsetup { *device* } {dstate}

drbdsetup { *device* } {resize} [-d { *size*}]

drbdsetup { *device* } {pause-sync}

drbdsetup { *device* } {resume-sync}

drbdsetup { *device* } {outdate}

drbdsetup { *device* } {show-gi}

drbdsetup { *device* } {get-gi}

drbdsetup { *device* } {show}

drbdsetup { *device* } {suspend-io}

drbdsetup { *device* } {resume-io}

drbdsetup { *device* } {events} [-u] [-a]

説明

drbdsetup は、DRBD デバイスと下位レベルブロックデバイスを結びつける、DRBD デバイス間で下位レベルデバイス同士ミラーリングを設定する、現在実行中の DRBD デバイスの設定を検査する、などの目的で使用する。

注意

drbdsetup は DRBD プログラム群の中での低レベルのツールである。デバイスドライバを操作するために、drbddisk や drbd スクリプトなどの中で使用されている。

コマンド

サブコマンドの中には、独自の引数やオプションを持つものがある。すべての値にはデフォルトの単位があるが、**K**、**M**、または**G**を明示的に指定することによって単位を変えられる。これらの単位はコンピュータでおなじみの方法で定義される ($K = 2^{10} = 1024$ 、 $M = 1024 K$ 、 $G = 1024 M$)。

全般オプション

すべての `drbdsetup` サブコマンドに次のオプションを指定することができる。

`--create-device`

指定した **DRBD** デバイスに対応するデバイスファイルがない場合、自動的に作成するよう指定する。

`--set-defaults`

`set-defaults` を指定すると、そのサブコマンドに関する全オプションのうち指定しなかったものを、デフォルト値に設定する。

ディスク

DRBD リソースに指定したデバイス (`device`) と その下位レベルブロックデバイス (`lower_device`) を結びつけ、下位レベルデバイスにデータを書き込めるようにする。下位レベルデバイスの利用可能な全領域を使わなくてもいい場合に限り、`-d` (または `disk-size`) オプションを指定する。このオプションを指定しなかった場合、対向ノードに接続できた後にデバイスが利用可能になる (`net` コマンドも参照)。

`-d, --disk-size size`

DRBD は利用できるデータ領域を自動的に決定する機能を持つ。対向ノードに接続することなくただちに下位デバイスを利用したい場合、このオプションを指定して、利用したいディスクサイズを `size` に指定する。デフォルトの単位はキロバイト ($1 \text{ KB} = 1024 \text{ バイト}$) である。

`-e, --on-io-error err_handler`

下位レベルのドライバがエラーを報告したときの **DRBD** の挙動を指定する。さらに上位のレイヤにそのエラーを伝える (`pass_on`)。ヘルププログラムを実行する (`call-local-io-error`)。デバイスから下位レベルデバイスを切り離して以後の I/O を対向ペアに委ねる (`detach`)。

`-f, --fencing fencing_policy`

2 ノードの接続が途絶えた状態で両ノードがともにプライマリになること (スプリットブレイク状態) を抑止するためのポリシーを指定する。

次のポリシーを指定できる。

dont-care

デフォルトの設定値で、フェンシングのためのアクションを実行しない。

resource-only

ノードが切り離されたプライマリ状態になると、DRBD は、DRBD は `outdate-peer` ハンドラを実行して他ノードを期限切れ状態に変えようとする。 `outdate` ハンドラは、他ノード上にネットワーク経由で接続し、 `'drbdadm outdate res'` を実行しようとする。

resource-and-stonith

ノードが切り離されたプライマリ状態になると、DRBD はすべてのディスク I/O を停止して `outdate-peer` ハンドラを呼び出す。 `outdate` ハンドラは、他ノード上にネットワーク経由で接続し、 `'drbdadm outdate res'` を実行しようとする。 ハンドラが他ノードに到達できない場合、DRBD は STONITH 機能を使って他ノードを強制排除する。これらが完了したら、ディスク I/O を再開する。ハンドラの実行が失敗した場合、 `drbdsetup` の `resume-io` コマンドを使ってディスク I/O を復旧させることができる。

`-b, --use-bmbv`

下位レベルのストレージドライバが `merge_bvec_fn()` 関数を備えている場合、DRBD は 4 キロバイトを越えない単位でのディスク I/O リクエストだけを処理する。本マニュアル執筆時点では、この機能を備えているドライバは、 `md` (ソフトウェア RAID)、 `dm` (デバイスマッピング LVM)、および DRBD 自身だけである。

ソフトウェア RAID やその他の `merge_bvec_fn()` 関数を持つドライバの上位で DRBD を使う場合であって、全部のクラスタ構成ノードで同関数が同様に振る舞うことがわかっている場合 (すなわちソフトウェア RAID などを構成する物理的なディスクが同タイプである場合)、最高のパフォーマンスを得るにはこのパラメータを設定すべきである。このオプションは、何をしているのかを理解した上でのみ使うように。

ネットワーク

対向ノードからの接続を `local_addr:port` で待ち受け、対向ノードの `remote_addr:port` に接続するよう、DRBD デバイスをセットアップする。 `port` を省略すると、デフォルトの 7788 が使われる。

TCP/IP リンク上の通信では、独自のプロトコルが使われる。3種類のプロトコル (A、B または C) を `protocol` に指定する。

プロトコル A: ローカルディスクとローカル TCP 送信バッファにデータを書き込んだらディスクへの書き込みが完了したと判断する。

プロトコル B: ローカルディスクとリモートバッファキャッシュにデータを書き込んだらディスクへの書き込みが完了したと判断する。

プロトコル C: ローカルディスクとリモートディスクの両方にデータを書き込んだらディスクへの書き込みが完了したと判断する。

`-c, --connect-int time`

対向ノードにただちに接続できない場合、DRBD は接続を繰り返し試行する。このパラメータは試行間隔を指定する。デフォルト値は 10 で、このパラメータの単位は秒である。

`-i, --ping-int time`

2つの DRBD ドライバ間の接続が確立していて、`time` 秒の間に何も通信が行われなかった場合、DRBD は対向ノードが生きているか確認するためのパケットを送出する。デフォルト値は 10 で、このパラメータの単位は秒である。

`-t, --timeout val`

対向ノードからの応答パケットが $1/10$ 秒の `time` 倍の時間以内に返ってこない場合、対向ノードが死んだと判断して、TCP/IP コネクションを切断する。この値は `connect-int` 値および `ping-int` よりも小さくなければならない。デフォルト値は 60 で、これは 6 秒に相当する。すなわちこのパラメータの単位は 0.1 秒である。

`-S, --sndbuf-size size`

ソケット送信バッファは、セカンダリノードに送信するパケットを格納するために使われる。この中のパケットは、(ネットワーク的には)セカンダリ側から受信確認を受け取っていない。プロトコル A を使う場合は、両ノード間の同期を高めるために、このバッファサイズを増やす必要が生じる可能性がある。しかし、プライマリノードがフェールしたときに失うデータが増えることも考慮しておく必要がある。`size` のデフォルト値は 128 キロバイトで、デフォルトの単位はキロバイトである。

`-k, --ko-count count`

セカンダリノードが書き込みリクエストを `count` 回以上失敗した場合、そのセカンダリノードはクラスタから排除され、プライマリノードは **StandAlone** モードに遷移する。デフォルト値は 0 で、これは本機能が無効になることを意味する。

`-e, --max-epoch-size val`

このオプションを指定すると、バリアとバリアの間の書き込みリクエスト数の最大値を制限できる。`max-buffers` と同じ値を指定するのが望ましい。100 より小さい値は、パフォーマンス低下をもたらす。デフォルト値は 2048 である。

`-b, --max-buffers val`

DRBD 受信スレッドに割り当てるバッファページの最大値を指定する。`max-epoch-size` と同じ値を指定するのが望ましい。小さい値はパフォーマンス低下をもたらす (最小値は 32)。デフォルト値は 2048 である。

`-u, --unplug-watermark val`

セカンダリノード上に書き込まれていない書き込みリクエスト数がこの値を上回ると、下位レベルのデバイスに対して書き込みリクエストを送る。ストレージによっては小さい値でも良好な結果が得られるが、多くのデバイスでは `max-buffers` と同じ値を指定するとき最良の結果が得られる。デフォルト値は 128 で、指定できる最小値は 16、最大値は 131072 である。

`-m, --allow-two-primaries`

このオプションを指定すると、両ノードにプライマリを割り当てられる。このオプションは分散共有ファイルシステムを使うときのみ指定する。現在 DRBD がサポートするファイルシステムは OCFS2 と GFS である。これら以外のファイルシステムを使うときに個のオプションを指定すると、データの破損とノードのダウンを引き起こす。

`-a, --cram-hmac-alg alg`

対向ノードの認証を行いたい場合、HMAC アルゴリズムを指定する。対向ノードの認証は行うべきである。チャレンジ-レスポンス方式で対向ノードを認証するのに、HMAC アルゴリズムが使われる。/proc/crypto に記録されている任意のダイジェストアルゴリズムを指定できる。

`-x, --shared-secret secret`

64 文字までの共有秘密鍵を指定する。

`-A, --after-sb-0pri asb-0p-policy`

スプリットブレイン状態が生じて両ノードが同時にプライマリになった後で、両ノードの接続が回復した場合に行われる修復方法を指定する。次のポリシーを指定できる。

`disconnect`

自動再同期を行わず接続を切断する。

`discard-younger-primary`

スプリットブレイン発生前にプライマリであったノードからの再同期を自動的に実行する。

`discard-older-primary`

スプリットブレイン発生時にプライマリになったノードからの再同期を自動的に実行する。

`discard-zero-changes`

スプリットブレイン発生後 どちらか一方のノードに書き込みがまったく行われなかったことが明白な場合、書き込みが行われたノードから行われなかったノードに対する再同期が実行される。どちらも書き込まれなかった場合は、DRBD はランダムな判断によって 0 ブロックの再同期を実行する。両ノードに書き込みが行われた場合、このポリシーはノードの接続を切断する。

`discard-least-changes`

スプリットブレイン発生後、より多くのブロックを書き込んだノードから他方に対する再同期を実行する。

`discard-node-NODENAME`

指定した名前のノードに対する再同期を実行する。

-B, --after-sb-1pri *asb-1p-policy*

スプリットブレイン発生後どちらか一方のノードがプライマリになっている場合、このパラメータのポリシーにしたがって修復が行われる。次のポリシーを指定できる。

disconnect

自動再同期を行わず接続を切断する。

consensus

after-sb-0pri アルゴリズムの結果が現在のセカンダリノードのデータを壊すことになる場合、セカンダリノードのデータを捨てる。そうではない場合は接続を切断する。

discard-secondary

セカンダリ側のデータを捨てる。

call-pri-lost-after-sb

after-sb-0pri アルゴリズムの判断をつねに採用する。セカンダリ側のデータが正しいと判断された場合には、現在のプライマリ側で *pri-lost-after-sb* ハンドラが呼び出される。

violently-as0p

after-sb-0pri アルゴリズムの判断をつねに採用する。現在のセカンダリ側が正しいデータを保持しているという結論になった場合でも、プライマリ側のデータの変更箇所を受け入れる。

-C, --after-sb-2pri *asb-2p-policy*

スプリットブレイン発生後に両ノードがプライマリになってしまった場合の処理を指定する。次のポリシーを指定できる。

disconnect

自動再同期を行わず接続を切断する。

call-pri-lost-after-sb

after-sb-0pri アルゴリズムの判断をつねに採用する。現在のセカンダリ側が正しいデータを保持しているという結論になった場合には、現在のプライマリ側で *pri-lost-after-sb* ハンドラを実行する。

violently-as0p

after-sb-0pri アルゴリズムの判断をつねに採用する。現在のセカンダリ側が正しいデータを保持しているという結論になった場合でも、プライマリ側のデータの変更箇所を受け入れる。

-P, --always-asbp

3番目のノードが存在しないことが現在の UUID 値から明らかな場合、通常のスプリットブレイン発生後の修復ポリシーだけが適用される。

このオプションを指定すると、両ノードのデータに関連性があると認められる場合のみ通常のスプリットブレイン発生後のポリシーが適用される。UUID の分析により3番目のノードの存在が疑われる場合や、なんらかの別の原因によって間違った UUID セットで判断してしまった場合には、フル同期が行われるかもしれない。

-R, --rr-conflict *role-resync-conflict-policy*

同期先ノードがプライマリ状態のときに、いつ再同期を実行すべきかの判断方法を制御する。設定できる値は、disconnect、call-pri-lost または violently である。disconnect は接続を切断する。call-pri-lost は pri-lost ハンドラを呼び出す。このハンドラは、ノードの状態をセカンダリに切り替えるか、あるいはノードをクラスタから切り離す処理を実行するべきである。デフォルト値は disconnect である。

violently を指定すると、プライマリノードを強制的に同期先 (SyncTarget) にできる。このオプションを指定すると、即座に同期元 (SyncSource) のデータに書き換えられる。このオプションは、意味および効果を明確に意識した上でのみ利用すべきである。

-d, --data-integrity-*alg hash_alg*

ネットワーク経由で受け渡されるデータの整合性を担保するために、DRBD はハッシュ値を比較する機能を備えている。通常は、TCP/IP パケット自体のヘッダに含まれる 16 ビットチェックサムで保証されるが、ギガビット NIC を備えた TCP/IP オフロードエンジンの中にはチェックサムを壊すものがあることがわかっている。したがってこのオプションをテスト中に指定し、十分にテストが終わった後は CPU 負荷を軽減するためにこのオプションを無効にするとよい。オプション値には、カーネルがサポートする任意のダイジェストアルゴリズムを指定できる。一般的なカーネルの場合、少なくとも md5、sha1 または crc32c のどれかが利用できる。デフォルトでは、この機能は無効である。

-p, --ping-timeout *ping_timeout*

ping-int パケットに対して対向ノードはこのパラメータに指定した時間以内に応答しなければならない。応答パケットが返ってこない場合、その対向ノードは死んだと判断される。デフォルト値は 500ms で、100ms 単位で指定する。

-D, --discard-my-data

スプリットブレイン状態から復旧するときに、このオプションを手作業で指定する。自動的な復旧ポリシーを設定していない場合には、DRBD は接続を拒否する。このオプションを実行すると、そのノードは接続後ただちに同期先になる。

同期

DRBD デバイスの同期デーモンに対する設定パラメータを実行中に変更する。

-r, --rate *rate*

DRBD の上位レイヤの円滑な実行のために、バックグラウンドの同期作業が利用するバンド幅を制限できる。デフォルト値は 250KB/秒。デフォルトの単位は KB/秒だが、K、M、G の

接尾語を補って単位を変更できる。

-a, --after *minor*

minor に指定したマイナー番号を持つデバイスが接続された後、他のデバイスの再同期を開始する。接続されるまでの間は **SyncPause** 状態になる。

-e, --al-extents *extents*

DRBD はホットエリア (アクティブセット) を自動的に検出できる。このパラメータを指定すると、ホットエリアの大きさを制御できる。各エクステントは、低レベルデバイスの 4 メガバイトの領域になる。予定外の事情によってプライマリノードがクラスタから切り離されると、そのときのホットエリアのデータは、次回接続したときの再同期の対象になる。このデータ構造は、メタデータ領域に書き込まれる。したがって、アクティブセットの状態更新は、メタデータデバイスへの書き込みを引き起こす。エクステント値を大きくすると、再同期所要時間が長くなるが、メタデータの更新頻度を減らすことができる。*extents* のデフォルト値は 127 で、指定できる最小値は 7、最大値は 3843 である。

-v, --verify-alg *hash-alg*

drbdsetup コマンドの **verify** サブコマンドでディスク内容をオンライン検証できる。ビット単位の比較の代わりに、ブロックごとのハッシュ値による検証が行われる。検証に利用するハッシュアルゴリズムは、このパラメータで指定する。オプション値には、カーネルがサポートする任意のダイジェストアルゴリズムを指定できる。一般的なカーネルの場合、少なくとも md5、sha1 または crc32c のどれかが利用できる。デフォルトでは、この機能は無効である。オンライン検証を有効にするには、このパラメータを明示的に設定する必要がある。

-c, --cpu-mask *cpu-mask*

DRBD カーネルスレッドに対する CPU アフィニティマスクを指定する。*cpu-mask* のデフォルト値は 0 で、DRBD カーネルスレッドは利用可能なすべての CPU にまたがって動作することを表す。

プライマリ

デバイス (*device*) をプライマリ状態に切り替える。これにより、アプリケーション (この場合はファイルシステムなど) はデバイスを読み書きモードでオープンできるようになる。プライマリ状態のデバイスに書き込んだデータは、セカンダリ状態のディスクにミラーされる。

通常の実行状態では、DRBD デバイスパアの両方を同時にプライマリ状態に切り替えることはできない。*allow-two-primaries* を指定すると、デフォルトの実行方法を変更して両ノードをともにプライマリ状態にできる。

-o, --overwrite-data-of-peer

ローカルディスクの複製物に不整合がある場合は、プライマリ状態になれない。このオプションを指定すると、この状況に関係なくプライマリ状態に切り替える。このオプションは、何が起きるか明確に理解している場合のみ指定すること。

セカンダリ

デバイス (*device*) をセカンダリ状態に切り替える。何らかのアプリケーションが書き込みモードでデバイスをオープンしている間は、この操作は失敗する。

DRBD デバイスペアの両方がともにセカンダリ状態になることは問題なく可能である。

verify

オンライン状態でのデバイスの検証を実施する。オンライン検証は、ローカルノードの全ブロックを対向ノードの対応するブロックと比較することである。検証の進行状況は、`/proc/drbd` を通じてモニタできる。ローカルディスクと対向ペアで内容が異なるブロックは、DRBD のディスク上ビットマップに不整合としてマークされる。検証で見つかった不整合の自動的な再同期は行われぬ。再同期を行うには、いったんリソースの接続を切って再接続すればよい。

このコマンドは、デバイスペアが接続されていない場合には失敗する。

invalidate

接続された DRBD デバイスペアのローカル側を `SyncTarget` 状態に切り替える。これにより、対向ペア側のすべてのデータブロックがローカルディスクにコピーされる。

このコマンドは、デバイスペアが接続されていない場合には失敗する。

invalidate-remote

接続された DRBD デバイスペアの対向ペア側を `SyncTarget` 状態に切り替える。これにより、ローカルディスクのすべてのデータブロックが対向ペアにコピーされる。

wait-connect

デバイスが対向ペアと通信可能になるまで待機する。

```
-t, --wfc-timeout wfc_timeout
-d, --degr-wfc-timeout degr_wfc_timeout
-w, --wait-after-sb
```

timeout 秒以内に対向ペアと通信できなければ、このコマンドは失敗する。リブート以前にペアがともに動作していた場合には、*wfc_timeout* パラメータの値が、リブート以前にペアの接続が切れていた場合には、*degr_wfc_timeout* パラメータの値が使われる。*wfc_timeout* のデフォルト値は 0 で、接続されるまで永久に待機することを表す。*degr_wfc_timeout* のデフォルト値は 120 秒である。スプリットブレイン状態が起きたためにデバイスが接続された後 `StandaAlone` 状態になってしまった場合、このコマンドは失敗する。`wait-after-sb` を指定すると、デフォルトの動作を変更できる。

wait-sync

デバイスの同期が終わるまで待機する。このオプションの動作は `wait-connect` コマンドと同じである。

disconnect

`net` コマンドで確立したデバイスに対する情報を全部削除する。その結果、デバイス間の接続は切れ、その後のネットワーク経由の情報は受け取らない。

detach

`disk` コマンドでデバイスに与えた情報を消去する。その結果、デバイスは下位レベルデバイスから切り離される。

down

デバイスに与えたすべての情報を消去し、未設定状態に戻す。

state

デバイスとその対向デバイスの現在の状態を表示する。表示形式は"ローカル/他ノード"である(例: Primary/Secondary)。

cstate

デバイスの現在の接続状態を表示する。

dstate

現在のデバイスと下位デバイスの関係を表示する。

resize

下位レベルデバイスのサイズを再評価する。実際には、両方の下位レベルデバイスのサイズを大きく変更した後に、各ノードでそれぞれ `resize` コマンドを実行する。

pause-sync

自ノードの一時停止フラグをセットして再同期を一時停止する。両ノードの一時停止フラグがともにセットされていない場合にのみ再同期が行われる。下位レベルデバイスの RAID を再構成しているときなど、DRBD の再同期を一時停止するのが望ましい場合がある。

resume-sync

自ノードの一時停止フラグをクリアする。

outdate

自ノードの下位レベルデバイスの内容が「時代遅れ」であるとマークする。時代遅れのノードはプライマリになれない。このコマンドは通常、fencing や 対向ペアの outdate-peer ハンドラと組み合わせて利用する。

show-gi

デバイスのデータ世代識別パート (data generation identifiers) の内容を説明テキストとともに表示する。

get-gi

デバイスのデータ世代識別パート (data generation identifiers) の内容を表示する。

show

デバイスに関するすべての設定情報を表示する。

suspend-io

このコマンドの明確な使いみちはないが、コマンドセットの完全性のために用意されている。

resume-io

fencing ポリシーが resource-and-stonith で、かつ outdate-peer ハンドラの実行が失敗した場合、このコマンドを実行することによって凍結されていたディスク I/O を再開できる。

events

DRBD のすべての状態変化とヘルパープログラムの呼び出し経緯を表示する。このコマンドは、DRBD の状態変化を他のプログラムにパイプで渡したいときに利用できる。

-a, --all-devices

すべての DRBD デバイスの状態を表示する。

-u, --unfiltered

このオプションはデバッグ用である。すべてのネットリンクレイやの受信メッセージの内容を表示する。

例題

このコマンドの使用例は *DRBD Quick Start Guide* (<http://www.linux-ha.org/DRBD/QuickStart07>) を参照のこと。

バージョン

このドキュメントは DRBD バージョン 8.2.2 向けに書かれている。

著者

Philipp Reisner <philipp.reisner@linbit.com>, Lars Ellenberg <lars.ellenberg@linbit.com>

バグ報告方法

バグについては、<drbd-user@lists.linbit.com>宛のメールで報告してほしい。

著作権

Copyright 2001-2008 LINBIT Information Technologies, Philipp Reisner, Lars Ellenberg. This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

本マニュアルの日本語版の翻訳著作権は、株式会社サードウェアが保有しています。

参照

drbd.conf(5), drbd(8), drbddisk(8) drbdadm(8) *DRBD* ホームページ (英語) (<http://www.drbd.org/>) *DRBD* ホームページ (日本語) (<http://www.drbd.jp/>)