Red Hat Ceph Storage 2

トラブルシューティングガイド

Red Hat Ceph Storage のトラブルシューティング

Last Updated: 2017-10-10

Red Hat Ceph Storage 2 トラブルシューティングガイド

Red Hat Ceph Storage のトラブルシューティング

法律上の通知

概要

本ガイドでは、Red Hat Ceph Storage でのよくある問題の解決方法について説明します。

目次

第1章 初期のトラブルシューティング

本章では以下について説明します。

- Ceph エラーのトラブルシュートの開始 (「問題の特定」)
- 最も一般的な **ceph health** エラーメッセージ (「**ceph health** コマンド出力について」)
- 最も一般的な Ceph ログエラーメッセージ (「Ceph ログについて」)

1.1. 問題の特定

Red Hat Ceph Storage で直面するエラーの原因を判定するために、構成を確認してから以下の質問に答えてください。

1. 特定の問題は、サポートされていない構成によって発生する場合があります。お使いの構成がサポートされていることを確認してください。詳細は、 Red Hat Ceph Storage でサポートされる構成 を参照してください。
2. Ceph のどのコンポーネントで問題が発生しているかご存知ですか?
   a. いいえ。この場合は 「Ceph Storage クラスターの健全性の診断」 に進んでください。
   b. モニター。この場合は 4章 *モニターのトラブルシューティング* に進んでください。
   c. OSD。この場合は 5章 *OSD のトラブルシューティング* に進んでください。
   d. プレイスメントグループ。この場合は 6章 *プレイスメントグループのトラブルシューティング* に進んでください。

1.1.1. Ceph Storage クラスターの健全性の診断

以下の手順では、Ceph Storage クラスターの健全性を診断する基本的なステップを説明します。

1. クラスターの全体的な状態を確認します。

   ```
   # ceph health detail
   ```

   このコマンドで **HEALTH_WARN** または **HEALTH_ERR** が返される場合は、「**ceph health** コマンド出力について」 を参照してください。

2. 「Ceph ログについて」 にあるエラーメッセージの Ceph ログを確認します。ログはデフォルトで **/var/log/ceph/** ディレクトリーに保存されます。

3. ログで十分な情報が見つからない場合は、デバッグレベルを上げてから失敗するアクションを再度実行します。2章 *ロギングの設定* を参照してください。

1.2. CEPH HEALTH コマンド出力について

ceph health コマンドは Ceph Storage クラスターの状態についての情報を返します。

- **HEALTH_OK** は、クラスターが健全であることを示します。
- **HEALTH_WARN** は警告です。Ceph が再バランスプロセスを完了した場合などは、**HEALTH_OK** が自動的に返される場合もあります。ただし、クラスターが長く **HEALTH_WARN** の状態にある

場合は、さらにトラブルシュートを行うことを検討してください。

- **HEALTH_ERR** は問題が重大であり、直ちに対応が必要であることを示します。

ceph health detail および **ceph -s** コマンドを使うとより詳細な出力が返されます。

以下のテーブルでは、モニター、OSD、およびプレイスメントグループに関するよくある **HEALTH_ERR** と **HEALTH_WARN** のエラーメッセージを示しています。各エラーの内容を説明し、その解決方法が記載された対応セクションも表示しています。

表1.1 モニターに関するエラーメッセージ

エラーメッセージ	参照先
HEALTH_WARN	
mon.X is down (out of quorum)	「モニターが Quorum 不足 (Out of Quorum)」
clock skew	「Clock Skew」
store is getting too big!	「Monitor ストアが大きくなりすぎている」

表1.2 OSD に関するエラーメッセージ

エラーメッセージ	参照先
HEALTH_ERR	
full osds	「Full OSDs」
HEALTH_WARN	
nearfull osds	「Nearfull OSDs」
osds are down	「(1 つ以上の) OSDs Are Down」 「OSD のフラッピング」
requests are blocked	「遅延リクエスト、およびリクエストがブロックされる」
slow requests	「遅延リクエスト、およびリクエストがブロックされる」

表1.3 プレイスメントグループに関するエラーメッセージ

エラーメッセージ	参照先
HEALTH_ERR	

エラーメッセージ	参照先
pgs down	「プレイスメントグループが **down**」
pgs inconsistent	「Inconsistent プレイスメントグループ」
scrub errors	「Inconsistent プレイスメントグループ」
HEALTH_WARN	
pgs stale	「Stale プレイスメントグループ」
unfound	「Unfound オブジェクト」

1.3. CEPH ログについて

デフォルトでは、Ceph はログを **/var/log/ceph/** ディレクトリーに保存します。

<cluster-name>.log は、グローバルのクラスターイベントを含むクラスターログファイルです。このログは、デフォルトで **ceph.log** と命名されます。メインクラスターログは、モニターホストにのみ格納されます。

OSD と Monitor にはそれぞれのログファイルがあり、**<cluster-name>-osd.<number>.log** と **<cluster-name>-mon.<hostname>.log** という名前になります。

Ceph サブシステムのデバッグレベルを上げると、Ceph はそのサブシステム向けの新規ログファイルを生成します。ロギングについての詳細は、*2章ロギングの設定* を参照してください。

以下のテーブルでは、モニターと OSD に関するよくある Ceph エラーメッセージを示しています。各エラーの内容を説明し、その解決方法が記載された対応セクションも表示しています。

表1.4 モニターに関する Ceph ログのよくあるエラーメッセージ

エラーメッセージ	ログファイル	参照先
clock skew	メインクラスターログ	「Clock Skew」
clocks not synchronized	メインクラスターログ	「Clock Skew」
Corruption: error in middle of record	モニターログ	「モニターが Quorum 不足 (Out of Quorum)」 「モニターストアの復旧」
Corruption: 1 missing files	モニターログ	「モニターが Quorum 不足 (Out of Quorum)」 「モニターストアの復旧」

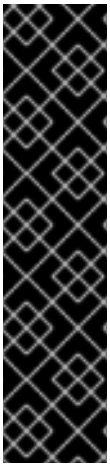
エラーメッセージ	ログファイル	参照先
`Caught signal (Bus error)`	モニターログ	「モニターが Quorum 不足 (Out of Quorum)」

表1.5 OSD に関する Ceph ログのよくあるエラーメッセージ

エラーメッセージ	ログファイル	参照先
`heartbeat_check: no reply from osd.X`	メインクラスターログ	「OSD のフラッピング」
`wrongly marked me down`	メインクラスターログ	「OSD のフラッピング」
`osds have slow requests`	メインクラスターログ	「遅延リクエスト、およびリクエストがブロックされる」
`FAILED assert(!m_filestore_fail_eio)`	OSD ログ	「(1 つ以上の) OSDs Are Down」
`FAILED assert(0 == "hit suicide timeout")`	OSD ログ	「(1 つ以上の) OSDs Are Down」

第2章 ロギングの設定

本章では、Ceph の各種サブシステム向けにロギングを設定する方法について説明します。



重要

ロギングはリソース集約的作業です。また、詳細なロギングは比較的短時間に膨大な量のデータを生成しかねません。クラスターの特定のサブシステム内で問題が発生している場合は、該当するサブシステムのロギングのみを有効にしてください。詳細は、「Ceph サブシステム」 を参照してください。

また、ログファイルのローテーション設定も検討してください。詳細は 「ログローテーションの迅速化」 を参照してください。

問題が解決したら、サブシステムのログとメモリーのレベルをデフォルト値に戻します。Ceph の全サブシステムおよびそれらのデフォルト値については、付録A *サブシステムのデフォルトのロギングレベル* を参照してください。

Ceph のロギングは以下の方法で設定できます。

- ランタイムに **ceph** コマンドを使用する。これが一般的なアプローチです。詳細は、「ランタイムのロギング設定」 を参照してください。
- Ceph 設定ファイルを更新する。クラスターの起動時に問題が発生している場合は、このアプローチを使用してください。詳細は、「Ceph 設定ファイルでのロギング設定」 を参照してください。

2.1. CEPH サブシステム

本セクションでは、Ceph サブシステムとそれらのロギングレベルについて説明します。

Ceph サブシステムとロギングレベル

Ceph はいくつかのサブシステムで構成されており、各サブシステムには以下のロギングレベルがあります。

- 出力ログ。これはデフォルトで **/var/log/ceph/** ディレクトリーに保存されます (ログレベル)。
- メモリーキャッシュに保存されるログ (メモリーレベル)。

一般的に、Ceph は以下の場合を除いてメモリーにあるログを出力ログに送信しません。

- 致命的なシグナルが発生した場合。
- リソースコードのアサートが発動された場合。
- ユーザーがリクエストした場合。

Ceph のロギングレベルは、**1** から **20** までの値を設定することができます。**1** が最も簡潔で、**20** が最も詳細になります。

ログレベルとメモリーレベルに単一の値を使用すると、それらに同じ値が設定されます。例えば、**debug_osd = 5** とすると、**ceph-osd** デーモンのデバッグレベルは **5** に設定されます。

ログレベルとメモリーレベルに異なる値を設定するには、それらの値をスラッシュ (/) でわけます。例えば、**debug_mon = 1/5** とすると、**ceph-mon** デーモンのデバッグログレベルは **1** に、メモリーログレベルは **5** に設定されます。

よく使用される Ceph サブシステムとそのデフォルト値

サブシステム	ログレベル	メモリーレベル	説明
asok	1	5	管理ソケット
auth	1	5	認証
client	0	5	**librados** を使ってクラスターに接続するアプリケーションまたはライブラリー
filestore	1	5	FileStore OSD バックエンド
journal	1	5	OSD ジャーナル
mds	1	5	メタデータサーバー
monc	0	5	モニタークライアントが、Ceph デーモンとモニター間のほとんどの通信を処理します。
mon	1	5	モニター
ms	0	5	Ceph コンポーネント間のメッセージングシステム
osd	0	5	OSD デーモン
paxos	0	5	モニターが合意を確立するために使用するアルゴリズム
rados	0	5	Ceph の核となるコンポーネントの Reliable Autonomic Distributed Object Store
rbd	0	5	Ceph ブロックデバイス
rgw	1	5	Ceph オブジェクトゲートウェイ

ログ出力の例

以下は、モニターおよび OSD 向けのログの詳細度を高めた場合に出力されるログメッセージの例です。

モニターのデバッグ設定

```
debug_ms = 5
```

```
debug_mon = 20
debug_paxos = 20
debug_auth = 20
```

モニターのデバッグ設定のログ出力例

```
2016-02-12 12:37:04.278761 7f45a9afc700 10 mon.cephn2@0(leader).osd e322
e322: 2 osds: 2 up, 2 in
2016-02-12 12:37:04.278792 7f45a9afc700 10 mon.cephn2@0(leader).osd e322
min_last_epoch_clean 322
2016-02-12 12:37:04.278795 7f45a9afc700 10 mon.cephn2@0(leader).log
v1010106 log
2016-02-12 12:37:04.278799 7f45a9afc700 10 mon.cephn2@0(leader).auth v2877
auth
2016-02-12 12:37:04.278811 7f45a9afc700 20 mon.cephn2@0(leader) e1
sync_trim_providers
2016-02-12 12:37:09.278914 7f45a9afc700 11 mon.cephn2@0(leader) e1 tick
2016-02-12 12:37:09.278949 7f45a9afc700 10 mon.cephn2@0(leader).pg v8126
v8126: 64 pgs: 64 active+clean; 60168 kB data, 172 MB used, 20285 MB /
20457 MB avail
2016-02-12 12:37:09.278975 7f45a9afc700 10
mon.cephn2@0(leader).paxosservice(pgmap 7511..8126) maybe_trim trim_to
7626 would only trim 115 < paxos_service_trim_min 250
2016-02-12 12:37:09.278982 7f45a9afc700 10 mon.cephn2@0(leader).osd e322
e322: 2 osds: 2 up, 2 in
2016-02-12 12:37:09.278989 7f45a9afc700  5
mon.cephn2@0(leader).paxos(paxos active c 1028850..1029466) is_readable =
1 - now=2016-02-12 12:37:09.278990 lease_expire=0.000000 has v0 lc 1029466
....
2016-02-12 12:59:18.769963 7f45a92fb700  1 -- 192.168.0.112:6789/0 <==
osd.1 192.168.0.114:6800/2801 5724 ==== pg_stats(0 pgs tid 3045 v 0) v1
==== 124+0+0 (2380105412 0 0) 0x5d96300 con 0x4d5bf40
2016-02-12 12:59:18.770053 7f45a92fb700  1 -- 192.168.0.112:6789/0 -->
192.168.0.114:6800/2801 -- pg_stats_ack(0 pgs tid 3045) v1 -- ?+0
0x550ae00 con 0x4d5bf40
2016-02-12 12:59:32.916397 7f45a9afc700  0
mon.cephn2@0(leader).data_health(1) update_stats avail 53% total 1951 MB,
used 780 MB, avail 1053 MB
....
2016-02-12 13:01:05.256263 7f45a92fb700  1 -- 192.168.0.112:6789/0 -->
192.168.0.113:6800/2410 -- mon_subscribe_ack(300s) v1 -- ?+0 0x4f283c0 con
0x4d5b440
```

OSD のデバッグ設定

```
debug_ms = 5
debug_osd = 20
debug_filestore = 20
debug_journal = 20
```

OSD のデバッグ設定のログ出力例

```
2016-02-12 11:27:53.869151 7f5d55d84700  1 -- 192.168.17.3:0/2410 -->
192.168.17.4:6801/2801 -- osd_ping(ping e322 stamp 2016-02-12
```

```
11:27:53.869147) v2 -- ?+0 0x63baa00 con 0x578dee0
2016-02-12 11:27:53.869214 7f5d55d84700  1 -- 192.168.17.3:0/2410 -->
192.168.0.114:6801/2801 -- osd_ping(ping e322 stamp 2016-02-12
11:27:53.869147) v2 -- ?+0 0x638f200 con 0x578e040
2016-02-12 11:27:53.870215 7f5d6359f700  1 -- 192.168.17.3:0/2410 <==
osd.1 192.168.0.114:6801/2801 109210 ==== osd_ping(ping_reply e322 stamp
2016-02-12 11:27:53.869147) v2 ==== 47+0+0 (261193640 0 0) 0x63c1a00 con
0x578e040
2016-02-12 11:27:53.870698 7f5d6359f700  1 -- 192.168.17.3:0/2410 <==
osd.1 192.168.17.4:6801/2801 109210 ==== osd_ping(ping_reply e322 stamp
2016-02-12 11:27:53.869147) v2 ==== 47+0+0 (261193640 0 0) 0x6313200 con
0x578dee0
....
2016-02-12 11:28:10.432313 7f5d6e71f700  5 osd.0 322 tick
2016-02-12 11:28:10.432375 7f5d6e71f700 20 osd.0 322 scrub_random_backoff
lost coin flip, randomly backing off
2016-02-12 11:28:10.432381 7f5d6e71f700 10 osd.0 322 do_waiters -- start
2016-02-12 11:28:10.432383 7f5d6e71f700 10 osd.0 322 do_waiters -- finish
```

その他の参照先

- 「ランタイムのロギング設定」
- 「Ceph 設定ファイルでのロギング設定」

2.2. ランタイムのロギング設定

ランタイムに Ceph デバッグ出力である **dout()** をアクティベートするには、以下を実行します。

```
ceph tell <type>.<id> injectargs --debug-<subsystem> <value> [--<name>
<value>]
```

上記コマンドで

- **<type>** を Ceph デーモンのタイプ (**osd**、**mon**、または **mds**) で置き換えます。
- **<id>** を Ceph デーモンの特定 ID で置き換えます。別の方法では、***** を使ってランタイム設定を特定のタイプのデーモンすべてに適用することもできます。
- **<subsystem>** を特定のサブシステムで置き換えます。詳細は 「Ceph サブシステム」 を参照してください。
- **<value>** を **1** から **20** までの数字で置き換えます。**1** が最も簡潔で、**20** が最も詳細になります。

例えば、**osd.0** という名前の OSD にある OSD サブシステムのログレベルを 0 に、メモリーレベルを 5 に設定するには、以下を実行します。

```
# ceph tell osd.0 injectargs --debug-osd 0/5
```

ランタイム設定を確認するには、以下を実行します。

1. **ceph-osd** や **ceph-mon** などの実行中の Ceph デーモンのあるホストにログインします。
2. 設定を表示します。

```
ceph daemon <name> config show | less
```

以下のように Ceph デーモンの名前を指定します。

```
# ceph daemon osd.0 config show | less
```

その他の参照先

- 「Ceph 設定ファイルでのロギング設定」
- Red Hat Ceph Storage 2 の Configuration Guide に記載の Logging Configuration Reference の章。

2.3. CEPH 設定ファイルでのロギング設定

起動時に Ceph デバッグ出力である **dout()** をアクティベートするには、Ceph 設定ファイルにデバッグ設定を追加します。

- 各デーモンで共通のサブシステムの場合は、**[global]** セクションに設定を追加します。
- 特定のデーモンのサブシステムの場合は、**[mon]**、**[osd]**、または **[mds]** などのデーモンのセクションに設定を追加します。

例を以下に示します。

```
[global]
        debug_ms = 1/5

[mon]
        debug_mon = 20
        debug_paxos = 1/5
        debug_auth = 2

[osd]
        debug_osd = 1/5
        debug_filestore = 1/5
        debug_journal = 1
        debug_monc = 5/20

[mds]
        debug_mds = 1
```

その他の参照先

- 「Ceph サブシステム」
- 「ランタイムのロギング設定」
- Red Hat Ceph Storage 2 の Configuration Guide に記載の Logging Configuration Reference の章。

2.4. ログローテーションの迅速化

Ceph コンポーネントのデバッグレベルを上げると、大量のデータが生成される可能性があります。

ディスクが満杯に近い場合は、**/etc/logrotate.d/ceph** にある Ceph ログローテーションファイルを修正することでログのローテーションを早めることができます。Cron ジョブスケジューラーはこのファイルを使用してログのローテーションをスケジュールします。

手順: ログローテーションの迅速化

1. ログのローテーションファイルでローテーション頻度の後に size 設定を追加します。

```
rotate 7
weekly
size <size>
compress
sharedscripts
```

例えば、ログファイルが 500 MB に達したらローテーションするようにします。

```
rotate 7
weekly
size 500 MB
compress
sharedscripts
size 500M
```

2. **crontab** エディターを開きます。

```
$ crontab -e
```

3. **/etc/logrotate.d/ceph** ファイルをチェックするようにするエントリーを追加します。Cron が 30 分ごとに **/etc/logrotate.d/ceph** をチェックするようにするには、以下のようにします。

```
30 * * * * /usr/sbin/logrotate /etc/logrotate.d/ceph >/dev/null 2>&1
```

その他の参照先

- Red Hat Enterprise Linux 7 システム管理者のガイドの システムタスクの自動化 のセクション。

第3章 ネットワーク問題のトラブルシューティング

本章では、ネットワークとネットワークタイムプロトコル (NTP) に関する基本的なトラブルシューティングの手順を説明します。

3.1. 基本的なネットワーク問題のトラブルシューティング

Red Hat Ceph Storage は、信頼性のあるネットワーク接続に大きく依存しています。Ceph ノードは、ネットワークを使用して相互に通信します。ネットワークに問題があると、OSD フラッピングや OSD が間違って **down** とレポートされるなどの OSD に関する多くの問題が発生しかねません。ネットワーク問題は、モニターの **clock skew** エラーも引き起こします。また、パケット損失や高レイテンシー、帯域幅が制限される場合には、クラスターのパフォーマンスと安定性に影響が出ます。

手順: 基本的なネットワーク問題のトラブルシューティング

1. Ceph 設定ファイルの **cluster_network** と **public_network** のパラメーターに適切な値が含まれているか確認します。

2. ネットワークインターフェイスが起動していることを確認します。詳細については、カスタマーポータルの Basic Network troubleshooting を参照してください。

3. Ceph のノードが短いホスト名を使って相互に接続できることを確認します。Red Hat Ceph Storage 2 の Installation Guide for Red Hat Enterprise Linux または Installation Guide for Ubuntu の **Setting DNS Name Resolution** セクションを参照してください。

4. ファイアウォールを使用する場合は、Ceph のノードが適切なポートで相互に接続できることを確認します。Red Hat Ceph Storage 2 の Installation Guide for Red Hat Enterprise Linux または Installation Guide for Ubuntu の **Configuring Firewall** セクションを参照してください。

5. インターフェイスカウンターにエラーがないこと、ホスト間の接続のレイテンシーが期待値内であること、パケット損失が無いことを確認します。詳細はカスタマーポータルの "ethtool" コマンドは何ですか? このコマンドを使用して、ネットワークデバイスおよびインターフェイスの情報を取得する方法は? と rx_fw_discards が原因でシステムがパケットを落とします のソリューションを参照してください。

6. パフォーマンスに問題がある場合は、レイテンシーの確認のほかに、**iperf** ユーティリティーを使用してクラスターの全ノード間のネットワーク帯域幅を確認します。詳細はカスタマーポータルの What are the performance benchmarking tools available for Red Hat Ceph Storage? のソリューションを参照してください。

7. 全ホストのネットワーク相互接続スピードが同じであることを確認してください。これが違うと、遅いノードが速いノードを遅らせる可能性があります。また、スイッチ間リンクが接続されているノードの集計帯域幅を処理できることを確認します。

その他の参照先

- Red Hat Enterprise Linux 7 ネットワークガイド
- カスタマーポータルにあるネットワーク問題のトラブルシュートに関連する 記事およびソリューション

3.2. 基本的な NTP 問題のトラブルシューティング

本セクションでは、基本的な NTP 問題のトラブルシュートの手順を説明します。

手順: 基本的な NTP 問題のトラブルシューティング

1. **ntpd** デーモンがモニターホストで稼働していることを確認します。

```
# systemctl status ntpd
```

2. **ntpd** が稼働していない場合は、これを有効にして開始します。

```
# systemctl enable ntpd
# systemctl start ntpd
```

3. **ntpd** がクロックを正常に同期していることを確認します。

```
$ ntpq -p
```

4. 高度な NTP トラブルシュートの手順については、カスタマーポータルにある NTP 問題のトラブルシューティング のソリューションを参照してください。

その他の参照先

- 「Clock Skew」

第4章 モニターのトラブルシューティング

本章では、Ceph モニターに関連する一般的な問題の解決方法を説明します。

はじめに

- ネットワーク接続を確認してください。詳細は *3章 ネットワーク問題のトラブルシューティング* を参照してください。

4.1. モニターに関連するよくあるエラーメッセージ

以下のテーブルでは、**ceph health detail** コマンドで返される、または Ceph ログに含まれる最も一般的なエラーメッセージを示しています。各エラーの内容を説明し、その解決方法が記載された対応セクションも表示しています。

表4.1 モニターに関するエラーメッセージ

エラーメッセージ	参照先
HEALTH_WARN	
`mon.X is down (out of quorum)`	「モニターが Quorum 不足 (Out of Quorum)」
`clock skew`	「Clock Skew」
`store is getting too big!`	「Monitor ストアが大きくなりすぎている」

表4.2 モニターに関する Ceph ログのよくあるエラーメッセージ

エラーメッセージ	ログファイル	参照先
`clock skew`	メインクラスターログ	「Clock Skew」
`clocks not synchronized`	メインクラスターログ	「Clock Skew」
`Corruption: error in middle of record`	モニターログ	「モニターが Quorum 不足 (Out of Quorum)」 「モニターストアの復旧」
`Corruption: 1 missing files`	モニターログ	「モニターが Quorum 不足 (Out of Quorum)」 「モニターストアの復旧」
`Caught signal (Bus error)`	モニターログ	「モニターが Quorum 不足 (Out of Quorum)」

4.1.1. モニターが Quorum 不足 (Out of Quorum)

1 つ以上のモニターで **down** とマークされますが、他のモニターでは引き続き quorum を形成することができます。また、**ceph health detail** コマンドも以下のようなエラーメッセージを返します。

```
HEALTH_WARN 1 mons down, quorum 1,2 mon.b,mon.c
mon.a (rank 0) addr 127.0.0.1:6789/0 is down (out of quorum)
```

エラー内容
Ceph がモニターを **down** とマークするには、いくつかの理由があります。

ceph-mon デーモンが稼働していなければ、ストレージが破損しているか、他のエラーが原因でこのデーモンが起動できない可能性があります。また、**/var/** パーティションが満杯という可能性もあります。このため、**ceph-mon** がデフォルトの **/var/lib/ceph/mon-<short-host-name>/store.db** の場所にあるストアに対する操作がなにもできず、終了することになります。

ceph-mon デーモンが稼働していて、かつモニターが quorum 不足になり **down** とマークされる場合は、モニターの状態によって問題の原因は異なります。

- モニターが予想時間よりも長く **probing** 状態にあると、他のモニターを見つけられなくなります。これはネットワーク問題が原因で発生しているか、モニターマップ (**monmap**) が古くなっていて、間違った IP アドレスにある他のモニターに接続を試みている可能性があります。**monmap** が最新の場合は、モニターのクロックが同期されていないこともあります。

- モニターが予想時間よりも長く **electing** 状態にある場合は、モニターのクロックが同期されていない可能性があります。

- モニターの状態が **synchronizing** から **electing** に変わり、また元に戻る場合は、クラスターの状態が進んでいます。つまり、同期プロセスが処理可能なスピードよりも速く新しいマップが生成されていることを示しています。

- モニターが **leader** または **peon** とマークしている場合は、そのモニターは quorum 状態にあり、クラスターの残りはその状態にないと考えられます。クロックの同期が失敗していることでこの問題が発生している可能性があります。

解決方法

1. **ceph-mon** デーモンが実行中であることを確認します。実行中でない場合は、これを起動します。

   ```
   systemctl status ceph-mon@<host-name>
   systemctl start ceph-mon@<host-name>
   ```

   <host-name> をデーモンが実行中のホストの短い名前で置き換えます。分からない場合は、**hostname -s** を使用します。

2. **ceph-mon** を起動できない場合は、**ceph-mon デーモンを起動できない** にある手順に従ってください。

3. **ceph-mon** デーモンを起動できるものの、**down** とマークされてしまう場合は、**ceph-mon デーモンは起動するが、down とマークされる** にある手順に従ってください。

ceph-mon デーモンを起動できない

1. 対応するモニターログをチェックします。これはデフォルトで **/var/log/ceph/ceph-mon.<host-name>.log** にあります。

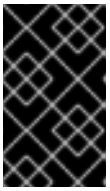
2. ログに以下のようなエラーメッセージがある場合は、モニターのストアが破損している可能性があります。

```
Corruption: error in middle of record
Corruption: 1 missing files; e.g.:
/var/lib/ceph/mon/mon.0/store.db/1234567.ldb
```

この問題を解決するには、モニターを置換します。「失敗したモニターの置き換え」 を参照してください。

3. ログに以下のようなエラーメッセージがある場合は、**/var/** パーティションが満杯になっている可能性があります。**/var/** から不要なデータを削除してください。

```
Caught signal (Bus error)
```



重要

モニターから直接手動でデータを削除しないでください。**ceph-monstore-tool** を使って圧縮するようにしてください。詳細は 「モニターストアの圧縮」 を参照してください。

4. これら以外のエラーメッセージがある場合は、サポートチケットを開いてください。詳細は 7章*Red Hat サポートへの連絡* を参照してください。

ceph-mon デーモンは起動するが、down とマークされる

1. Quorum 不足となっているモニターホストから、**mon_status** コマンドを使って状態をチェックします。

```
ceph daemon <id> mon_status
```

<id> をモニターの ID で置き換えます。例を示します。

```
# ceph daemon mon.a mon_status
```

2. ステータスが **probing** の場合は、**mon_status** 出力にある他のモニターの場所を確認します。

   a. アドレスが正しくない場合は、モニターに間違ったモニターマップ (**monmap**) があります。この問題の解決方法については、「モニターマップの挿入」 を参照してください。

   b. アドレスが正しい場合は、モニタークロックが同期されているか確認します。詳細は 「Clock Skew」 を参照してください。また、3章*ネットワーク問題のトラブルシューティング* を参照してネットワーク問題を解決してください。

3. ステータスが **electing** の場合は、モニタークロックが同期されているか確認します。「Clock Skew」 を参照してください。

4. ステータスが **electing** から **synchronizing** に変わる場合は、サポートチケットを開いてください。詳細は 7章*Red Hat サポートへの連絡* を参照してください。

5. モニターが **leader** または **peon** の場合は、モニタークロックが同期されているか確認します。「Clock Skew」 を参照してください。クロックを同期しても問題が解決しない場合は、サポートチケットを開いてください。詳細は 7章*Red Hat サポートへの連絡* を参照してください。

その他の参照先

- 「モニターのステータスについて」
- Red Hat Ceph Storage 2 の Administration Guide に記載の Starting, Stopping, Restarting a Daemon by Instances のセクション。
- Red Hat Ceph Storage 2 の Administration Guide に記載の Using the Administration Socket のセクション。

4.1.2. Clock Skew

Ceph モニターが quorum 不足で、**ceph health detail** コマンドの出力に以下のようなエラーメッセージが含まれます。

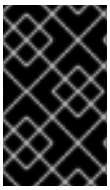
```
mon.a (rank 0) addr 127.0.0.1:6789/0 is down (out of quorum)
mon.a addr 127.0.0.1:6789/0 clock skew 0.08235s > max 0.05s (latency
0.0045s)
```

また、Ceph ログに以下のようなエラーメッセージが含まれます。

```
2015-06-04 07:28:32.035795 7f806062e700 0 log [WRN] : mon.a
127.0.0.1:6789/0 clock skew 0.14s > max 0.05s
2015-06-04 04:31:25.773235 7f4997663700 0 log [WRN] : message from mon.1
was stamped 0.186257s in the future, clocks not synchronized
```

エラー内容
clock skew エラーメッセージは、モニターのクロックが同期されていないことを示します。モニターは正確な時間に依存し、クロックが同期されたいないと予測できない動作をするので、クロックの同期は重要になります。

クロック間で許容される差異は **mon_clock_drift_allowed** パラメーターで指定し、デフォルトでは 0.05 秒に設定されます。



重要

テストをせずに **mon_clock_drift_allowed** のデフォルト値を変更しないでください。この値を変更すると、モニターと Ceph ストレージクラスター全般の安定性に影響を与える場合があります。

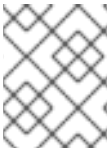
clock skew エラーの原因として考えられるものには、ネットワーク問題やネットワークタイムプロトコル (NTP) の同期 (設定されている場合) があります。なお、仮想マシンにデプロイされているモニターでは、時間の同期が正常に機能しません。

解決方法

1. ネットワークが正常に機能していることを確認します。詳細は *3章 ネットワーク問題のトラブルシューティング* を参照してください。特に NTP を使用している場合は、NTP クライアントに関する問題を解決してください。詳細情報は、「基本的な NTP 問題のトラブルシューティング」 を参照してください。
2. リモートの NTP サーバーを使用している場合は、ご自分のネットワークに独自の NTP サーバーを導入することを検討してください。詳細は、Red Hat Enterprise Linux 7 システム管理者のガイドの ntpd を使用した NTP 設定 の章を参照してください。

3. NTP クライアントを使用していない場合は、これを設定します。詳細は、Red Hat Ceph Storage 2 Installation Guide for Red Hat Enterprise Linux もしくは Ubuntu の **Configuring Network Time Protocol** のセクションを参照してください。

4. モニターを仮想マシンでホストしている場合は、これをベアメタルのホストに移してください。仮想マシンでのモニターのホストはサポートされていません。詳細は、Red Hat カスタマーポータルの Red Hat Ceph Storage でサポートされる構成 を参照してください。



注記

Ceph が時間の同期を評価するのは 5 分ごとなので、問題を修正してから **`clock skew`** メッセージが消えるまでに時間のずれがあります。

その他の参照先

- 「モニターのステータスについて」
- 「モニターが Quorum 不足 (Out of Quorum)」

4.1.3. Monitor ストアが大きくなりすぎている

`ceph health` コマンドで以下のようなエラーメッセージが返されます。

```
mon.ceph1 store is getting too big! 48031 MB >= 15360 MB -- 62% avail
```

エラー内容
Ceph モニターは、エントリーをキーと値のペアとして保存する LevelDB データベースです。このデータベースにはクラスターマップが含まれ、デフォルトでは **`/var/lib/ceph/mon/<cluster-name>-<short-host-name>/store.db`** に格納されます。

大規模なモニターストアへのクエリーには時間がかかります。このため、クライアントのクエリーへの反応が遅くなる可能性があります。

また、**`/var/`** パーティションが満杯の場合は、モニターはストアへの書き込み操作ができず、終了します。この問題の解決方法については、「モニターが Quorum 不足 (Out of Quorum)」 を参照してください。

解決方法

1. データベースのサイズを確認します。

   ```
   du -sch /var/lib/ceph/mon/<cluster-name>-<short-host-name>/store.db
   ```

   クラスター名と **`ceph-mon`** を実行しているホストの短い名前を指定します。例を示します。

   ```
   # du -sch /var/lib/ceph/mon/ceph-host1/store.db
   47G     /var/lib/ceph/mon/ceph-ceph1/store.db/
   47G     total
   ```

2. モニターストアを圧縮します。詳細は 「モニターストアの圧縮」 を参照してください。

その他の参照先

- 「モニターが Quorum 不足 (Out of Quorum)」

4.1.4. モニターのステータスについて

mon_status コマンドは以下のようなモニターについての情報を返します。

- State
- Rank
- Elections epoch
- Monitor map (**monmap**)

モニターで quorum の形成が可能な場合は、**ceph** ユーティリティーに **mon_status** を使用します。

モニターで quorum を形成できないものの、**ceph-mon** デーモンが実行中の場合は、管理ソケットを使用して **mon_status** を実行します。詳細は、Red Hat Ceph Storage 2 の Administration Guide に記載の Using the Administration Socket のセクションを参照してください。

mon_status の出力例

```
{
    "name": "mon.3",
    "rank": 2,
    "state": "peon",
    "election_epoch": 96,
    "quorum": [
        1,
        2
    ],
    "outside_quorum": [],
    "extra_probe_peers": [],
    "sync_provider": [],
    "monmap": {
        "epoch": 1,
        "fsid": "d5552d32-9d1d-436c-8db1-ab5fc2c63cd0",
        "modified": "0.000000",
        "created": "0.000000",
        "mons": [
            {
                "rank": 0,
                "name": "mon.1",
                "addr": "172.25.1.10:6789\/0"
            },
            {
                "rank": 1,
                "name": "mon.2",
                "addr": "172.25.1.12:6789\/0"
            },
            {
                "rank": 2,
                "name": "mon.3",
                "addr": "172.25.1.13:6789\/0"
            }
        ]
    }
}
```

モニターの状態

Leader

Electing 段階では、モニターは leader を選出しています。Leader とは最高ランクのモニターのことで、rank の値は一番低いものになります。上記の例の leader は、**mon.1** になります。

Peon

Peon は、leader ではない quorum 内のモニターのことです。leader が失敗した場合に、ランクの一番高い peon が新たな leader になります。

Probing

モニターが他のモニターを探している状態が probing です。例えば、モニターは起動した後、quorum を形成するためにモニターマップ (**monmap**) で指定された数のモニターを見つけるまで **probing** 状態にあります。

Electing

leader を選出するプロセス中であれば、モニターは electing 状態にあります。通常このステータスはすぐに変更されます。

Synchronizing

モニターが quorum に参加するために他のモニターと同期していると、synchronizing 状態になります。モニターのストア容量が小さければ小さいほど、同期プロセスは速くなります。このため、大規模ストアだと、同期に時間がかかります。

4.2. モニターマップの挿入

モニターに古いまたは破損したモニターマップ (**monmap**) があると、間違った IP アドレスで他のモニターに接続しようとすることになるので、quorum に参加できなくなります。

この問題を解決する最も安全な方法は、別のモニターから実際のモニターマップを挿入することです。このアクションでは、挿入先のモニターにある既存のモニターマップが上書きされることに注意してください。

以下の手順では、他のモニターが quorum を形成できる、または少なくとも 1 つのモニターに正常なモニターマップある場合に、モニターマップを挿入する方法を説明します。すべてのモニターのストアが破損していて、モニターマップも破損している場合は、「モニターストアの復旧」 を参照してください。

手順: モニターマップの挿入

1. 該当モニター以外のモニターで quorum を形成できる場合、**ceph mon getmap** コマンドでモニターマップを取得します。

```
# ceph mon getmap -o /tmp/monmap
```

2. 他のモニターでは quorum を形成できないものの、少なくとも 1 つのモニターに正常なモニターマップがある場合は、これをコピーします。

   a. モニターマップのコピー元となるモニターを停止します。

```
systemctl stop ceph-mon@<host-name>
```

   例えば、短いホスト名が **host1** のホスト上で実行中のモニターを停止するには、以下のコマンドを使用します。

```
# systemctl stop ceph-mon@host1
```

b. モニターマップをコピーします。

```
ceph-mon -i <id> --extract-monmap /tmp/monmap
```

<id> をモニターマップのコピー元となるモニターの ID で置き換えます。

```
# ceph-mon -i mon.a  --extract-monmap /tmp/monmap
```

3. モニターマップが破損している、または古くなっているモニターを停止します。

```
systemctl stop ceph-mon@<host-name>
```

例えば、短いホスト名が **host2** のホスト上で実行中のモニターを停止するには、以下のコマンドを使用します。

```
# systemctl stop ceph-mon@host2
```

4. モニターマップを挿入します。

```
ceph-mon -i <id> --inject-monmap /tmp/monmap
```

<id> をモニターマップが破損または古くなっているモニターの ID で置き換えます。例を示します。

```
# ceph-mon -i mon.c --inject-monmap /tmp/monmap
```

5. モニターを起動します。

```
# systemctl start ceph-mon@host2
```

モニターマップのコピー元となったモニターも起動します。

```
# systemctl start ceph-mon@host1
```

その他の参照先

- 「モニターが Quorum 不足 (Out of Quorum)」
- 「モニターストアの復旧」

4.3. モニターストアの復旧

Ceph モニターは、LevelDB のようなキーと値のストアにクラスターマップを保存します。モニター上でストアが破損していると、モニターが突然、終了し、再度起動できなくなります。Ceph ログには以下のエラーが含まれている場合があります。

```
Corruption: error in middle of record
Corruption: 1 missing files; e.g.:
/var/lib/ceph/mon/mon.0/store.db/1234567.ldb
```

実稼働のクラスターでは、あるモニターが失敗しても別のクラスターが相互に置換できるように、少なくとも 3 つのモニターを使用する必要があります。ただし、特定の条件では、すべてのモニターでスト

アが破損することもあります。例えば、モニターノードでディスクやファイルシステムが間違って設定されていると、停電が発生した場合に基礎となるファイルシステムが破損する可能性があります。

すべてのモニターでストアが破損している場合は、**ceph-monstore-tool** および **ceph-objectstore-tool** というユーティリティーで OSD ノードに保存されている情報を使って復旧することができます。

重要

この手順では以下の情報は復旧できません。

- メタデータデーモンサーバー (MDS) のキーリングおよびマップ
- プレイスメントグループ設定:
  - **ceph pg set_full_ratio** コマンドを使用した **full ratio** セット
  - **ceph pg set_nearfull_ratio** コマンドを使用した **nearfull ratio** セット

はじめに

- **rsync** ユーティリティーと **ceph-test** パッケージがインストールされていることを確認してください。

手順: モニターストアの復旧
ストアが破損しているモニターノードから以下のコマンドを実行します。

1. 全 OSD ノードからクラスターマップを収集します。

```
ms=<directory>
mkdir $ms

for host in $host_list; do
  rsync -avz "$ms" root@$host:"$ms"; rm -rf "$ms"
  ssh root@$host <<EOF
  for osd in  /var/lib/ceph/osd/ceph-*; do
    ceph-objectstore-tool --data-path \$osd --op update-mon-db --
mon-store-path $ms
  done
EOF
rsync -avz root@$host:$ms $ms; done
```

<directory> を収集するクラスターマップを保存する一時ディレクトリーに置き換えます。例を示します。

```
$ ms=/tmp/monstore/
$ mkdir $ms
$ for host in $host_list; do
  rsync -avz "$ms" root@$host:"$ms"; rm -rf "$ms"
  ssh root@$host <<EOF
  for osd in  /var/lib/ceph/osd/ceph-*; do
    ceph-objectstore-tool --data-path \$osd --op update-mon-db --
mon-store-path $ms
```

```
  done
EOF
rsync -avz root@$host:$ms $ms; done
```

2. 適切なケイパビリティーを設定します。

```
ceph-authtool <keyring>  -n mon. --cap mon 'allow *'
ceph-authtool <keyring>  -n client.admin --cap mon 'allow *' --cap
osd 'allow *' --cap mds 'allow *'
```

<keyring> をクライアントの管理キーリングへのパスで置き換えます。例を示します。

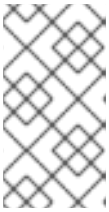
```
$ ceph-authtool /etc/ceph/ceph.client.admin.keyring  -n mon. --cap
mon 'allow *'
$ ceph-authtool /etc/ceph/ceph.client.admin.keyring  -n client.admin
--cap mon 'allow *' --cap osd 'allow *' --cap mds 'allow *'
```

3. 収集したマップからモニターストアを再ビルドします。

```
ceph-monstore-tool <directory> rebuild -- --keyring <keyring>
```

<directory> を最初のステップの一時ディレクトリーで置き換え、 **<keyring>** をクライアントの管理キーリングへのパスで置き換えます。例を示します。

```
$ ceph-monstore-tool /tmp/mon-store rebuild -- --keyring
/etc/ceph/ceph.client.admin.keyring
```



注記

cephfx 認証を使用しない場合は、 **--keyring** オプションは省略します。

```
$ ceph-monstore-tool /tmp/mon-store rebuild
```

4. 破損したストアのバックアップを作成します。

```
mv /var/lib/ceph/mon/<mon-ID>/store.db \
   /var/lib/ceph/mon/<mon-ID>/store.db.corrupted
```

<mon-ID> を **<mon.0>** といったモニター ID で置き換えます。

```
# mv /var/lib/ceph/mon/mon.0/store.db \
     /var/lib/ceph/mon/mon.0/store.db.corrupted
```

5. 破損したストアを置き換えます。

```
mv /tmp/mon-store/store.db /var/lib/ceph/mon/<mon-ID>/store.db
```

<mon-ID> を **<mon.0>** といったモニター ID で置き換えます。

```
# mv /tmp/mon-store/store.db /var/lib/ceph/mon/mon.0/store.db
```

ストアが破損している全モニターでこれらのステップを繰り返します。

6. 新規ストアの所有者を変更します。

```
chown -R ceph:ceph /var/lib/ceph/mon/<mon-ID>/store.db
```

<mon-ID> を **<mon.0>** といったモニター ID で置き換えます。

```
# chown -R ceph:ceph /var/lib/ceph/mon/mon.0/store.db
```

ストアが破損している全モニターでこれらのステップを繰り返します。

その他の参照先

- 「失敗したモニターの置き換え」

4.4. 失敗したモニターの置き換え

モニターのストアが破損した場合の解決方法には、Ansible 自動化アプリケーションを使用してモニターを置換することが推奨されます。

はじめに

- あるモニターを削除する前に、ほかのモニターが実行中でそれらが quorum を形成できることを確認してください。

手順: 失敗したモニターの置き換え

1. モニターホストから、デフォルトで **/var/lib/ceph/mon/<cluster-name>-<short-host-name>** にあるモニターストアを削除します。

```
rm -rf /var/lib/ceph/mon/<cluster-name>-<short-host-name>
```

モニターホストの短いホスト名とクラスター名を指定します。例えば、 **remote** という名前のクラスターから **host1** で実行中のモニターのモニターストアを削除するには、以下を実行します。

```
# rm -rf /var/lib/ceph/mon/remote-host1
```

2. モニターマップ (**monmap**) からモニターを削除します。

```
ceph mon remove <short-host-name> --cluster <cluster-name>
```

モニターホストの短いホスト名とクラスター名を指定します。例えば、 **remote** という名前のクラスターから **host1** で実行中のモニターを削除するには、以下を実行します。

```
# ceph mon remove host1 --cluster remote
```

3. モニターホストのハードウェアもしくは基礎となっているファイルシステムに関連する問題を解決します。

4. Ansible の管理ノードから **ceph-ansible** プレイブックを実行してモニターを再デプロイします。

```
# /usr/share/ceph-ansible/ansible-playbook site.yml
```

その他の参照先

- 「モニターが Quorum 不足 (Out of Quorum)」
- Red Hat Ceph Storage 2 の Administration Guide に記載の Managing Cluster Size の章。
- Red Hat Ceph Storage 2 の Installation Guide for Red Hat Enterprise Linux に記載の Deploying a Ceph Cluster の章。

4.5. モニターストアの圧縮

モニターストアのサイズが大きくなったら、以下の方法でこれを圧縮することができます。

- **ceph tell** コマンドを使って動的に圧縮します。具体的な手順については、モニターストアを動的に圧縮する を参照してください。
- **ceph-mon** デーモンの起動時に圧縮します。詳細は 起動時にモニターストアを圧縮する の手順を参照してください。
- **ceph-mon** を実行していない時に **ceph-monstore-tool** を使って圧縮します。上記の方法でモニターストアを圧縮できない場合、もしくはモニターが quorum の外にあり、そのログに **Caught signal (Bus error)** エラーメッセージが含まれている場合はこの方法を使用してください。詳細は、**ceph-monstore-tool** を使ったモニターストアの圧縮 の手順を参照してください。

重要

モニターストアのサイズは、クラスターが **active+clean** 状態でない場合、もしくは再バランスプロセス中に変化します。このため、モニターストアの圧縮は再バランスが完了してから行なってください。また、プレイスメントグループが **active+clean** の状態にあることを確認してください。

手順: モニターストアを動的に圧縮する

ceph-mon デーモンの実行中にモニターストアを圧縮するには、以下を実行します。

```
ceph tell mon.<host-name> compact
```

<host-name> を **ceph-mon** が実行中のホストの短い名前で置き換えます。分からない場合は、**hostname -s** を使用します。

```
# ceph tell mon.host1 compact
```

手順: 起動時にモニターストアを圧縮する

1. Ceph 設定の **[mon]** セクション下に以下のパラメーターを追加します。

```
[mon]
mon_compact_on_start = true
```

2. **ceph-mon** デーモンを再起動します。

```
systemctl restart ceph-mon@<host-name>
```

<host-name> をデーモンが実行中のホストの短い名前で置き換えます。分からない場合は、**hostname -s** を使用します。

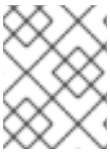
```
# systemctl restart ceph-mon@host1
```

3. モニターが quorum を形成していることを確認します。

```
# ceph mon stat
```

4. 必要に応じて他のモニターでこのステップを繰り返します。

手順: ceph-monstore-tool を使ってモニターストアを圧縮する



注記

まず最初に **ceph-test** パッケージがインストールされていることを確認してください。

1. **ceph-mon** デーモンが大きいストアで稼働していないことを確認します。必要に応じてデーモンを停止します。

```
systemctl status ceph-mon@<host-name>
systemctl stop ceph-mon@<host-name>
```

<host-name> をデーモンが実行中のホストの短い名前で置き換えます。分からない場合は、**hostname -s** を使用します。

```
# systemctl status ceph-mon@host1
# systemctl stop ceph-mon@host1
```

2. モニターストアを圧縮します。

```
ceph-monstore-tool /var/lib/ceph/mon/mon.<host-name> compact
```

<host-name> をモニターホストの短いホスト名で置き換えます。

```
# ceph-monstore-tool /var/lib/ceph/mon/mon.node1 compact
```

3. **ceph-mon** を再起動します。

```
systemctl start ceph-mon@<host-name>
```

例を以下に示します。

```
# systemctl start ceph-mon@host1
```

その他の参照先

- 「Monitor ストアが大きくなりすぎている」

- 「モニターが Quorum 不足 (Out of Quorum)」

第5章 OSD のトラブルシューティング

本章では、Ceph OSD に関連する一般的な問題の解決方法を説明します。

はじめに

- ネットワーク接続を確認してください。詳細は *3章 ネットワーク問題のトラブルシューティング* を参照してください。
- **ceph health** コマンドを実行してモニターに quorum があることを確認します。正常なステータス (**HEALTH_OK**、**HEALTH_WARN**、または **HEALTH_ERR**) が返されると、モニターで quorum が形成できることを示しています。これらが返されない場合は、まずモニターの問題を解決してください。詳細は *4章 モニターのトラブルシューティング* を参照してください。**ceph health** ついての詳細は、「**ceph health** コマンド出力について」 を参照してください。
- オプションで、再バランスプロセスを停止して時間とリソースを節約することもできます。詳細は 「再バランスの停止と開始」 を参照してください。

5.1. OSD に関連するよくあるエラーメッセージ

以下のテーブルでは、**ceph health detail** コマンドで返される、または Ceph ログに含まれる最も一般的なエラーメッセージを示しています。各エラーの内容を説明し、その解決方法が記載された対応セクションも表示しています。

表5.1 OSD に関するエラーメッセージ

エラーメッセージ	参照先
HEALTH_ERR	
full osds	「Full OSDs」
HEALTH_WARN	
nearfull osds	「Nearfull OSDs」
osds are down	「(1 つ以上の) OSDs Are Down」 「OSD のフラッピング」
requests are blocked	「遅延リクエスト、およびリクエストがブロックされる」
slow requests	「遅延リクエスト、およびリクエストがブロックされる」

表5.2 OSD に関する Ceph ログのよくあるエラーメッセージ

エラーメッセージ	ログファイル	参照先
heartbeat_check: no reply from osd.X	メインクラスターログ	「OSD のフラッピング」
wrongly marked me down	メインクラスターログ	「OSD のフラッピング」
osds have slow requests	メインクラスターログ	「遅延リクエスト、およびリクエストがブロックされる」
FAILED assert(!m_filestore_fail_eio)	OSD ログ	「(1 つ以上の) OSDs Are Down」
FAILED assert(0 == "hit suicide timeout")	OSD ログ	「(1 つ以上の) OSDs Are Down」

5.1.1. Full OSDs

ceph health detail コマンドで以下のようなエラーメッセージが返されます。

```
HEALTH_ERR 1 full osds
osd.3 is full at 95%
```

エラー内容
Ceph によって、クライアントは完全な OSD ノードでデータ損失を回避するための I/O 操作が実行できません。**mon_osd_full_ratio** パラメーターで設定されている容量にクラスターが到達すると、**HEALTH_ERR full osds** メッセージが返されます。デフォルトではこのパラメーターは **0.95** に設定されており、これはクラスター容量の 95% を意味します。

解決方法
raw ストレージの使用されている割合 (**%RAW USED**) を判定します。

```
# ceph df
```

%RAW USED が 70-75% を超えている場合は、以下が実行可能です。

- 不要なデータを削除します。実稼働のダウンタイムを回避するには、これが短期的な解決策になります。詳細は、「満杯のクラスターからデータを削除する」 を参照してください。
- 新規 OSD ノードを追加してクラスターを拡大します。Red Hat が推奨する長期的な解決策はこちらになります。詳細は、Red Hat Ceph Storage 2 の Administration Guide に記載の Adding and Removing OSD Nodes の章を参照してください。

その他の参照先

- 「Nearfull OSDs」

5.1.2. Nearfull OSDs

ceph health detail コマンドで以下のようなエラーメッセージが返されます。

-

```
HEALTH_WARN 1 nearfull osds
osd.2 is near full at 85%
```

エラー内容

`mon osd nearfull ratio defaults` パラメーターで設定されている容量にクラスターが到達すると、**`nearfull osds`** メッセージが返されます。デフォルトではこのパラメーターは **`0.85`** に設定されており、これはクラスター容量の 85% を意味します。

Ceph は CRUSH 階層に基づいて最善の方法でデータを分配しますが、等しく分配されることは保証されません。データが不平等に分配され、**`nearfull osds`** メッセージが返される主な原因は以下のとおりです。

- OSD がクラスター内の OSD ノード間でバランス化されていない。つまり、ある OSD ノードの方がほかのノードよりも非常に多くの OSD をホストしています。または、CRUSH マップ内のいくつかの OSD のウェイトがそれらの容量に十分ではありません。
- プレイスメントグループ (PG) の数が、OSD 数、ユースケース、OSD あたりの PG、および OSD 利用率に対して適切なものではない。
- クラスターが不適切な CRUSH の調整可能なパラメーターを使用している。
- OSD のバックエンドストレージが満杯に近い。

解決方法

1. PG カウントが十分であることを確認し、必要であればこれを増やします。詳細は、「PG カウントの増加」 を参照してください。
2. CRUSH の調整可能なパラメーターでクラスターのバージョンに合ったものを使用していることを確認します。必要に応じてこれを調整してください。詳細は、Red Hat Ceph Storage 2 の Storage Strategies guide 記載の CRUSH Tunables セクションと、Red Hat カスタマーポータルの How can I test the impact CRUSH map tunable modifications will have on my PG distribution across OSDs in Red Hat Ceph Storage? を参照してください。
3. 利用率によって OSD のウェイトを変更します。詳細は、Red Hat Ceph Storage 2 の Storage Strategies guide 記載の Set an OSD's Weight by Utilization セクションを参照してください。
4. OSD が使用するディスクの空き容量を判定します。
   a. OSD が使用しているスペース全体を確認するには、以下を実行します。

   ```
   # ceph osd df
   ```

   b. OSD が特定のノード上で使用しているスペースを確認します。**`nearful`** OSD があるノードから以下のコマンドを実行します。

   ```
   $ df
   ```

   c. 必要な場合は新規 OSD ノードを追加します。詳細は、Red Hat Ceph Storage 2 の Administration Guide に記載の Adding and Removing OSD Nodes の章を参照してください。

その他の参照先

- 「Full OSDs」

5.1.3. (1 つ以上の) OSDs Are Down

ceph health コマンドで以下のようなエラーメッセージが返されます。

```
HEALTH_WARN 1/3 in osds are down
```

エラー内容
サービス障害または他の OSD との通信問題のために、**ceph-osd** プロセスの 1 つが利用できません。このため、残った **ceph-osd** デーモンがこの失敗をモニターに報告しました。

ceph-osd デーモンが稼働していない場合は、基礎となる OSD ドライブまたはファイルシステムが破損しているか、キーリングがないなどの他のエラーによってデーモンが起動できなくなっています。

ceph-osd デーモンが稼働している、または **down** とマークされている場合はほとんどのケースで、ネットワーク問題によってこの状況が発生しています。

解決方法

1. どの OSD が **down** になっているか判定します。

```
# ceph health detail
HEALTH_WARN 1/3 in osds are down
osd.0 is down since epoch 23, last address
192.168.106.220:6800/11080
```

2. **ceph-osd** デーモンを再起動します。

```
systemctl restart ceph-osd@<OSD-number>
```

<OSD-number> を **down** になっている OSD の ID で置き換えます。例を示します。

```
# systemctl restart ceph-osd@0
```

   a. **ceph-osd** を起動できない場合は、**ceph-osd** デーモンを起動できない にある手順に従ってください。

   b. **ceph-osd** デーモンを起動できるものの、**down** とマークされてしまう場合は、**ceph-osd** デーモンは起動するが、**down** とマークされる にある手順に従ってください。

ceph-osd デーモンを起動できない

1. 多くの OSD (通常、12 以上) を含むノードの場合は、デフォルトのスレッド最大数 (PID カウント) が十分かどうかを確認してください。詳細は、「PID カウントの増加」 を参照してください。

2. OSD データおよびジャーナルパーティションが正常にマウントされていることを確認します。

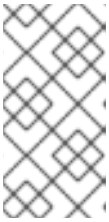
```
# ceph-disk list
...
/dev/vdb :
 /dev/vdb1 ceph data, prepared
 /dev/vdb2 ceph journal
/dev/vdc :
 /dev/vdc1 ceph data, active, cluster ceph, osd.1, journal /dev/vdc2
 /dev/vdc2 ceph journal, for /dev/vdc1
```

```
/dev/sdd1 :
 /dev/sdd1 ceph data, unprepared
 /dev/sdd2 ceph journal
```

ceph-disk で **active** とマークされているものは、パーティションがマウントされています。**prepared** となっているパーティションは、マウントします。詳細は、「OSD データパーティションのマウント」 を参照してください。**unprepared** となっているパーティションについては、マウントする前に準備する必要があります。詳細は、Red Hat Ceph Storage 2 の Administration Guide に記載の Preparing the OSD Data and Journal Drives セクションを参照してください。

3. **ERROR: missing keyring, cannot use cephx for authentication** というエラーメッセージが返されたら、OSD にキーリングがないことを意味します。詳細は、Red Hat Ceph Storage 2 の Administration Guide に記載の Keyring Management セクションを参照してください。

4. **ERROR: unable to open OSD superblock on /var/lib/ceph/osd/ceph-1** というエラーメッセージが返されたら、**ceph-osd** デーモンが基礎となるファイルシステムの読み込みをできないことを意味します。このエラーの解決方法については、以下の手順を参考にしてください。



注記

OSD ホストの起動中にこのエラーメッセージが返される場合は、サポートチケットを開いてください。Red Hat Bugzilla 1439210 で追跡中の既知の問題である可能性があります。詳細は、*7章Red Hat サポートへの連絡* を参照してください。

5. エラーの原因を判定するために、対応するログファイルをチェックします。デフォルトでは、Ceph はログファイルを **/var/log/ceph/** ディレクトリーに保存します。

   a. 以下のような **EIO** エラーメッセージは、基礎となるディスクに障害があることを示しています。

   ```
   FAILED assert(!m_filestore_fail_eio || r != -5)
   ```

   この問題を解決するには、基礎となる OSD ディスクを交換します。詳細は、「OSD ドライブの交換」 を参照してください。

   b. ログに以下のような別の **FAILED assert** エラーがある場合は、サポートチケットを開いてください。詳細は、*7章Red Hat サポートへの連絡* を参照してください。

   ```
   FAILED assert(0 == "hit suicide timeout")
   ```

6. **dmesg** で、基礎となるファイルシステムまたはディスクのエラー出力をチェックします。

```
$ dmesg
```

   a. 以下のような **error -5** エラーメッセージは、基礎となる XFS ファイルシステムが破損していることを示します。この問題の解決方法については、Red Hat カスタマーポータルの xfs_log_force: error 5 returned は何を示していますか? を参照してください。

   ```
   xfs_log_force: error -5 returned
   ```

b. **dmesg** の出力に **SCSI error** エラーメッセージがある場合は、Red Hat カスタマーポータルの SCSI Error Codes Solution Finder ソリューションを参考にして問題解決方法を判断してください。

c. 別の方法では、基礎となるファイルシステムを修正できない場合は、OSD ドライブを交換します。詳細は、「OSD ドライブの交換」 を参照してください。

7. 以下のようなセグメンテーション違反で OSD が失敗する場合は、情報を収集してサポートチケットを開いてください。詳細は、*7章Red Hat サポートへの連絡* を参照してください。

```
Caught signal (Segmentation fault)
```

ceph-osd は稼働しているが down とマークされる

1. エラーの原因を判定するために、対応するログファイルをチェックします。デフォルトでは、Ceph はログファイルを **/var/log/ceph/** ディレクトリーに保存します。

a. ログに以下のようなエラーメッセージがある場合は、「OSD のフラッピング」 を参照してください。

```
wrongly marked me down
heartbeat_check: no reply from osd.2 since back
```

b. これら以外のエラーメッセージがある場合は、サポートチケットを開いてください。詳細は *7章Red Hat サポートへの連絡* を参照してください。

その他の参照先

- 「OSD のフラッピング」
- 「Stale プレイスメントグループ」
- Red Hat Ceph Storage 2 の Administration Guide に記載の Starting, Stopping, Restarting a Daemon by Instances のセクション。

5.1.4. OSD のフラッピング

ceph -w | grep osds コマンドで、短時間に OSD が何度も **down** と表示された後に **up** と示されます。

```
# ceph -w | grep osds
2017-04-05 06:27:20.810535 mon.0 [INF] osdmap e609: 9 osds: 8 up, 9 in
2017-04-05 06:27:24.120611 mon.0 [INF] osdmap e611: 9 osds: 7 up, 9 in
2017-04-05 06:27:25.975622 mon.0 [INF] HEALTH_WARN; 118 pgs stale; 2/9 in
osds are down
2017-04-05 06:27:27.489790 mon.0 [INF] osdmap e614: 9 osds: 6 up, 9 in
2017-04-05 06:27:36.540000 mon.0 [INF] osdmap e616: 9 osds: 7 up, 9 in
2017-04-05 06:27:39.681913 mon.0 [INF] osdmap e618: 9 osds: 8 up, 9 in
2017-04-05 06:27:43.269401 mon.0 [INF] osdmap e620: 9 osds: 9 up, 9 in
2017-04-05 06:27:54.884426 mon.0 [INF] osdmap e622: 9 osds: 8 up, 9 in
2017-04-05 06:27:57.398706 mon.0 [INF] osdmap e624: 9 osds: 7 up, 9 in
2017-04-05 06:27:59.669841 mon.0 [INF] osdmap e625: 9 osds: 6 up, 9 in
2017-04-05 06:28:07.043677 mon.0 [INF] osdmap e628: 9 osds: 7 up, 9 in
2017-04-05 06:28:10.512331 mon.0 [INF] osdmap e630: 9 osds: 8 up, 9 in
2017-04-05 06:28:12.670923 mon.0 [INF] osdmap e631: 9 osds: 9 up, 9 in
```

また、Ceph ログに以下のようなエラーメッセージが含まれます。

```
2016-07-25 03:44:06.510583 osd.50 127.0.0.1:6801/149046 18992 : cluster
[WRN] map e600547 wrongly marked me down
```

```
2016-07-25 19:00:08.906864 7fa2a0033700 -1 osd.254 609110 heartbeat_check:
no reply from osd.2 since back 2016-07-25 19:00:07.444113 front 2016-07-25
18:59:48.311935 (cutoff 2016-07-25 18:59:48.906862)
```

エラー内容
OSD のフラッピングの主な原因は以下のとおりです。

- スクラビングやリカバリーなどの特定のクラスター操作には、非常に長い時間がかかります。例えば、大規模なインデックスのあるオブジェクトや大型のプレイスメントグループでこれらの操作を実行する場合などです。通常は、これらの操作が完了すると、OSD のフラッピング問題は解消します。
- 基礎となる物理ハードウェアの問題。この場合は、**ceph health detail** コマンドが **slow requests** エラーメッセージを返します。詳細は、「遅延リクエスト、およびリクエストがブロックされる」 を参照してください。
- ネットワークの問題。

パブリック (フロントエンド) ネットワークが正常に機能している間にクラスター (バックエンド) ネットワークが失敗したり、大幅なレイテンシーが発生すると、OSD はこのような状況をうまく処理出来ません。

OSD は、それらが **up** や **in** であることを示すために相互にハートビートパケットを送信する際にクラスターネットワークを使用します。クラスターネットワークが正常に機能しないと、OSD はハートビートパケットの送受信ができません。この場合、それぞれを **up** とマークする一方で、**down** になっていることをモニターに報告します。

Ceph 設定ファイルの以下のパラメーターでこの動作が決まります。

パラメーター	説明	デフォルト値
osd_heartbeat_grace_time	OSD を **down** とモニターにレポートするまでに OSD がハートビートパケットの返信を待機する時間。	20 秒
mon_osd_min_down_reporters	モニターが OSD を **down** とマークするまでに OSD が他の OSD を **down** とレポートする数。	1
mon_osd_min_down_reports	モニターが OSD を **down** とマークするまでに OSD が **down** とレポートされる回数。	3

上記のテーブルでは、デフォルト設定の場合、1 つの OSD が最初の OSD を 3 回 **down** と報告するだけで、モニターが OSD を **down** とマークします。場合によっては、1 つのホストがネットワーク問題に遭遇すると、クラスター全体で OSD フラッピングが発生することがあります。これは、そのホストにある OSD がクラスター内の他の OSD を **down** とレポートするためです。

注記

OSD フラッピングのシナリオには、OSD プロセスの開始直後にこれを強制終了するという状況は含まれていません。

解決方法

1. **ceph health detail** コマンドの出力を再度チェックします。これに **slow requests** エラーメッセージが含まれている場合は、「遅延リクエスト、およびリクエストがブロックされる」 の解決方法を参照してください。

```
# ceph health detail
HEALTH_WARN 30 requests are blocked > 32 sec; 3 osds have slow
requests
30 ops are blocked > 268435 sec
1 ops are blocked > 268435 sec on osd.11
1 ops are blocked > 268435 sec on osd.18
28 ops are blocked > 268435 sec on osd.39
3 osds have slow requests
```

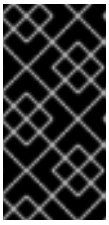
2. **down** とマークされている OSD と、これがどのノードにあるか判別します。

```
# ceph osd tree | grep down
```

3. フラップしている OSD のあるノードで、ネットワークの問題を解決します。詳細は、*3章ネットワーク問題のトラブルシューティング* を参照してください。

4. 別の方法では、**noup** および **nodown** フラグを設定することで、モニターが OSD を一時的に **down** または **up** とマークできないようにすることもできます。

```
# ceph osd set noup
# ceph osd set nodown
```



重要

noup および **nodown** のフラグの使用は問題を根本的に解決するわけではなく、OSD のフラッピングを回避するだけです。ご自分でこのエラーを解決できない場合は、サポートチケットを開いてください。詳細は、*7章Red Hat サポートへの連絡* を参照してください。

その他の参照先

- Red Hat Ceph Storage 2 Installation Guide for Red Hat Enterprise Linux もしくは Ubuntu の **Configuring Network** セクション。
- Red Hat Ceph Storage 2 の Administration Guide に記載の Heartbeating のセクション。

5.1.5. 遅延リクエスト、およびリクエストがブロックされる

リクエストに対する **ceph-osd** デーモンの応答が遅く、**ceph health detail** コマンドで以下のようなエラーメッセージが返されます。

```
HEALTH_WARN 30 requests are blocked > 32 sec; 3 osds have slow requests
```

```
30 ops are blocked > 268435 sec
1 ops are blocked > 268435 sec on osd.11
1 ops are blocked > 268435 sec on osd.18
28 ops are blocked > 268435 sec on osd.39
3 osds have slow requests
```

また、Ceph ログに以下のようなエラーメッセージが含まれます。

```
2015-08-24 13:18:10.024659 osd.1 127.0.0.1:6812/3032 9 : cluster [WRN] 6
slow requests, 6 included below; oldest blocked for > 61.758455 secs
```

```
2016-07-25 03:44:06.510583 osd.50 [WRN] slow request 30.005692 seconds
old, received at {date-time}: osd_op(client.4240.0:8 benchmark_data_ceph-
1_39426_object7 [write 0~4194304] 0.69848840) v4 currently waiting for
subops from [610]
```

エラー内容
遅延リクエストの OSD とは、**`osd_op_complaint_time`** パラメーターで定義した時間内でキューにおける IOPS (I/O operations per second) を実行できない OSD のことです。デフォルトでは、このパラメーターは 30 秒に設定されています。

OSD のリクエスト応答が遅くなる主な原因は以下のとおりです。

- ディスクドライブやホスト、ラック、ネットワークスイッチなどの基礎となるハードウェアに問題がある場合。
- ネットワークに問題がある場合。この問題は通常、OSD フラッピングに関連しています。詳細は、「OSD のフラッピング」 を参照してください。
- システム負荷

以下のテーブルでは、遅延リクエストのタイプを示しています。**`dump_historic_ops`** 管理ソケットコマンドを使用してタイプを判別します。管理ソケットについての詳細は、Red Hat Ceph Storage 2 の Administration Guide に記載の Using the Administration Socket のセクションを参照してください。

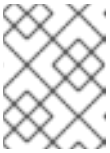
遅延リクエストのタイプ	説明
`waiting for rw locks`	OSD は、操作のためにプレイスメントグループのロックの取得を待機しています。
`waiting for subops`	OSD は、レプリカ OSD が操作をジャーナルに適用することを待っています。
`no flag points reached`	OSD は、操作の主要な節目に到達しませんでした。
`waiting for degraded object`	OSD によるオブジェクトの複製回数が、まだ指定された数に到達していません。

解決方法

1. 遅延リクエストの OSD またはリクエストブロックの OSD が、ディスクドライブやホスト、ラック、ネットワークスイッチなどのハードウェアを共有しているか判別します。

2. OSD がディスクを共有している場合は、以下の手順を実行します。

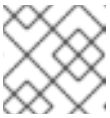
   a. **smartmontools** ユーティリティーを使ってディスクまたはログの健全性を確認し、ディスクにエラーがあるかどうか判断します。

   注記

   smartmontools ユーティリティーは **smartmontools** パッケージに含まれています。

   b. **iostat** ユーティリティーを使って OSD ディスク上の I/O wait レポート (**%iowai**) を取得し、ディスクの負荷が高くなっているかどうかを確認します。

   注記

   iostat ユーティリティーは **sysstat** パッケージに含まれています。

3. OSD がホストを共有している場合は、以下の手順を実行します。

   a. RAM と CPU の使用率をチェックします。

   b. **netstat** ユーティリティーを使用して NIC (ネットワークインターフェイスコントローラー) のネットワーク統計値を確認し、ネットワーク問題を解決します。詳細情報は、*3章 ネットワーク問題のトラブルシューティング* を参照してください。

4. OSD がラックを共有している場合は、ラックのネットワークスイッチをチェックします。例えば、ジャンボフレームを使用している場合は、パスにある NIC にジャンボフレームが設定されていることを確認します。

5. 遅延リクエストの OSD で共有されているハードウェアを特定できない場合、またはハードウェアやネットワークの問題を解決できない場合は、サポートチケットを開いてください。詳細は、*7章Red Hat サポートへの連絡* を参照してください。

その他の参照先

- Red Hat Ceph Storage 2 の Administration Guide に記載の Using the Administration Socket のセクション。

5.2. 再バランスの停止と開始

OSD が失敗するか、これを停止すると、CRUSH アルゴリズムが自動的に再バランスプロセスを開始して、データを残りの OSD に再分配します。

再バランスは時間がかかり、リソースも多く使用するので、トラブルシュートの間や OSD のメンテナンス時には再バランスを停止することを検討してください。これを停止するには、OSD の停止前に **noout** フラグを設定します。

```
# ceph osd set noout
```

トラブルシュートやメンテナンスが終了したら、**noout** フラグの設定を解除して再バランスを開始します。

```
# ceph osd unset noout
```

注記

停止した OSD 内のプレイスメントグループは、トラブルシュートおよびメンテナンス中に **degraded** となります。

その他の参照先

- Red Hat Ceph Storage 2 の Architecture Guide に記載の Rebalancing and Recovery のセクション。

5.3. OSD データパーティションのマウント

OSD データパーティションが正常にマウントされていないと、**ceph-osd** デーモンを起動することができなくなります。パーティションが想定通りにマウントされていないことが判明したら、本セクションの手順にしたがってマウントしてください。

手順: OSD データパーティションのマウント

1. パーティションをマウントします。

```
# mount -o noatime <partition> /var/lib/ceph/osd/<cluster-name>-
<osd-number>
```

<partition> を、OSD データ専用の OSD ドライブにあるパーティションへのパスで置き換えます。以下のように、クラスター名と OSD 番号を指定します。

```
# mount -o noatime /dev/sdd1 /var/lib/ceph/osd/ceph-0
```

2. 失敗した **ceph-osd** デーモンを起動します。

```
# systemctl start ceph-osd@<OSD-number>
```

<OSD-number> を OSD の ID で置き換えます。例を示します。

```
# systemctl start ceph-osd@0
```

その他の参照先

- 「(1 つ以上の) OSDs Are Down」

5.4. OSD ドライブの交換

Ceph は フォールトトラレンス設計となっているので、**degraded** 状態でもデータを損失することなく動作できます。このため、Ceph はデータストレージドライブが失敗した場合でも、作動します。ドライブが失敗した場合での **degraded** 状態とは、他の OSD に保存されているデータの余分なコピーがクラスター内の他の OSD に自動的にバックフィルが実行されます。ただし、これが発生した場合は、失敗した OSD ドライブを取り外し、手動で OSD を再生成してください。

ドライブが失敗すると、Ceph は OSD が **down** とレポートします。

```
HEALTH_WARN 1/3 in osds are down
osd.0 is down since epoch 23, last address 192.168.106.220:6800/11080
```

注記

ネットワーク問題またはパーミッション問題のために、Ceph が OSD を **down** とマークすることもあります。詳細は、「(1 つ以上の) OSDs Are Down」 を参照してください。

最近のサーバーは通常、ホットスワップ対応のドライブが装備されているので、失敗したドライブは、ノードを停止せずに新しいドライブと交換することができます。全体的な手順は以下のようになります。

1. Ceph クラスターから OSD を削除します。詳細は、Ceph クラスターから OSD を削除する を参照してください。
2. ドライブを交換します。詳細は、物理ドライブの交換 を参照してください。
3. クラスターに OSD を追加します。Ceph クラスターに OSD を追加する を参照してください。

はじめに

1. どの OSD が **down** になっているか判定します。

```
# ceph osd tree | grep -i down
ID WEIGHT  TYPE NAME      UP/DOWN REWEIGHT PRIMARY-AFFINITY
 0 0.00999         osd.0     down  1.00000          1.00000
```

2. OSD プロセスが停止していることを確認します。OSD ノードから以下のコマンドを実行します。

```
# systemctl status ceph-osd@<OSD-number>
```

<OSD-number> を **down** になっている OSD の ID で置き換えます。例を示します。

```
# systemctl status ceph-osd@osd.0
...
   Active: inactive (dead)
```

ceph-osd デーモンが実行中の場合。OSD は **down** とマークされているものの、対応する **ceph-osd** デーモンが実行中の場合についてのトラブルシュートは、「(1 つ以上の) OSDs Are Down」 を参照してください。

手順: Ceph クラスターから OSD を削除する

1. OSD を **out** とマークします。

```
# ceph osd out osd.<OSD-number>
```

<OSD-number> を **down** になっている OSD の ID で置き換えます。例を示します。

```
# ceph osd out osd.0
marked out osd.0.
```

注記

OSD が **down** となっている場合、Ceph がその OSD からハートビートパケットを 900 秒間受け取らなければ、自動的にこれを **out** とマークします。これが発生すると、失敗した OSD データのコピーがある他の OSD がバックフィルを開始して、クラスター内に必要な数のコピーが存在するようにします。クラスターはバックフィルを行う間、**degraded** 状態になります。

2. 失敗した OSD がバックフィルを行なっていることを確認します。出力は以下のようになります。

```
# ceph -w | grep backfill
2017-06-02 04:48:03.403872 mon.0 [INF] pgmap v10293282: 431 pgs: 1
active+undersized+degraded+remapped+backfilling, 28
active+undersized+degraded, 49
active+undersized+degraded+remapped+wait_backfill, 59
stale+active+clean, 294 active+clean; 72347 MB data, 101302 MB used,
1624 GB / 1722 GB avail; 227 kB/s rd, 1358 B/s wr, 12 op/s;
10626/35917 objects degraded (29.585%); 6757/35917 objects misplaced
(18.813%); 63500 kB/s, 15 objects/s recovering
2017-06-02 04:48:04.414397 mon.0 [INF] pgmap v10293283: 431 pgs: 2
active+undersized+degraded+remapped+backfilling, 75
active+undersized+degraded+remapped+wait_backfill, 59
stale+active+clean, 295 active+clean; 72347 MB data, 101398 MB used,
1623 GB / 1722 GB avail; 969 kB/s rd, 6778 B/s wr, 32 op/s;
10626/35917 objects degraded (29.585%); 10580/35917 objects
misplaced (29.457%); 125 MB/s, 31 objects/s recovering
2017-06-02 04:48:00.380063 osd.1 [INF] 0.6f starting backfill to
osd.0 from (0'0,0'0] MAX to 2521'166639
2017-06-02 04:48:00.380139 osd.1 [INF] 0.48 starting backfill to
osd.0 from (0'0,0'0] MAX to 2513'43079
2017-06-02 04:48:00.380260 osd.1 [INF] 0.d starting backfill to
osd.0 from (0'0,0'0] MAX to 2513'136847
2017-06-02 04:48:00.380849 osd.1 [INF] 0.71 starting backfill to
osd.0 from (0'0,0'0] MAX to 2331'28496
2017-06-02 04:48:00.381027 osd.1 [INF] 0.51 starting backfill to
osd.0 from (0'0,0'0] MAX to 2513'87544
```

3. CRUSH マップから OSD を削除します。

```
# ceph osd crush remove osd.<OSD-number>
```

<OSD-number> を **down** になっている OSD の ID で置き換えます。例を示します。

```
# ceph osd crush remove osd.0
removed item id 0 name 'osd.0' from crush map
```

4. OSD に関連付けられた認証キーを削除します。

```
# ceph auth del osd.<OSD-number>
```

<OSD-number> を **down** になっている OSD の ID で置き換えます。例を示します。

```
# ceph auth del osd.0
updated
```

5. Ceph ストレージクラスターから OSD を削除します。

```
# ceph osd rm osd.<OSD-number>
```

<OSD-number> を **down** になっている OSD の ID で置き換えます。例を示します。

```
# ceph osd rm osd.0
removed osd.0
```

以下のコマンド出力に OSD が含まれていなければ、正常に削除されています。

```
# ceph osd tree
```

6. 失敗したドライブをアンマウントします。

```
# umount /var/lib/ceph/osd/<cluster-name>-<OSD-number>
```

クラスター名と OSD の ID を指定します。例を示します。

```
# umount /var/lib/ceph/osd/ceph-0/
```

以下のコマンド出力にドライブが含まれていなければ、アンマウントが成功しています。

```
# df -h
```

手順: 物理ドライブの交換

1. ハードウェアノードのドキュメントで、物理ドライブ交換についての詳細を参照します。
   a. ドライブがホットスワップ対応の場合は、失敗したドライブを新規のもので交換します。
   b. ドライブがホットスワップ対応ではなく、ノードに複数の OSD がある場合は、ノード全体をシャットダウンして物理ドライブを交換する必要がある可能性があります。クラスターがバックフィルを行わないことを検討してください。詳細は 「再バランスの停止と開始」を参照してください。
2. ドライブが **/dev/** ディレクトリー下に表示される際のパスを書き留めます。
3. OSD を手動で追加する場合は、OSD ドライブを見つけてディスクをフォーマットします。

手順: Ceph クラスターに OSD を追加する

1. OSD を再度追加します。
   a. クラスターのデプロイに Ansible を使っている場合は、Ceph 管理サーバーから **ceph-ansible** プレイブックを再度実行します。

```
# ansible-playbook /usr/share/ceph-ansible site.yml
```

b. OSD を手動で追加している場合は、Red Hat Ceph Storage 2 の Administration Guide に記載の Adding an OSD with the Command-line Interface セクションを参照してください。

2. CRUSH 階層が正確であることを確認します。

```
# ceph osd tree
```

3. CRUSH 階層内での OSD の場所が想定したものではない場合は、OSD を所定の場所に移動します。

```
ceph osd crush move <bucket-to-move> <bucket-type>=<parent-bucket>
```

例えば、**sdd:row1** にある bucket を root bucket に移動するには、以下を実行します。

```
# ceph osd crush move ssd:row1 root=ssd:root
```

その他の参照先

- 「(1 つ以上の) OSDs Are Down」
- Red Hat Ceph Storage 2 の Administration Guide に記載の Managing Cluster Size の章。
- Red Hat Ceph Storage 2 Installation Guide for Red Hat Enterprise Linux もしくは Installation Guide for Ubuntu。

5.5. PID カウントの増加

12 を超える数の Ceph OSD があるノードでは、特にリカバリー中にはスレッド (PID カウント) のデフォルトの最大数が十分でない場合があります。このため、**ceph-osd** デーモンが終了して再起動に失敗することがあります。このような事態になったら、スレッドを可能な範囲の最大数に増やします。

カウントを一時的に増やすには、以下を実行します。

```
# sysctl -w kernel.pid.max=4194303
```

カウントを永続的に増やすには、**/etc/sysctl.conf** ファイルを以下のように更新します。

```
kernel.pid.max = 4194303
```

5.6. 満杯のクラスターからデータを削除する

mon_osd_full_ratio パラメーターで指定した容量に OSD が到達すると、Ceph は自動的にこの OSDでの I/O 操作ができなくなり、**full osds** というエラーメッセージを返します。

以下の手順では不要なデータを削除して、この問題を解決する方法を説明します。

注記

mon_osd_full_ratio パラメーターは、クラスター作成時に **full_ratio** パラメーターの値を設定します。**mon_osd_full_ratio** の値を後で変えることはできません。**full_ratio** の値を一時的に増やすには、代わりに **pg_full_ratio** を増やします。

手順: 満杯のクラスターからデータを削除する

1. **full_ratio** の現在の値を確認します。デフォルトでは **0.95** に設定されています。

```
# ceph pg dump | grep -i full
full_ratio 0.95
```

2. **pg_full_ratio** の値を一時的に **0.97** に増やします。

```
# ceph pg set_full_ratio 0.97
```

重要

Red Hat では、**pg_full_ratio** を 0.97 より大きい値に設定しないことを強く推奨しています。これより大きい値を設定すると、リカバリープロセスが難しくなり、完全な OSD を復旧できなくなる可能性があります。

3. パラメーターが **0.97** に設定されたことを確認します。

```
# ceph pg dump | grep -i full
full_ratio 0.97
```

4. クラスターの状態をモニターします。

```
# ceph -w
```

クラスターの状態が **full** から **nearfull** に変わったらすぐに不要なデータを削除します。

5. **full ratio** の値を **0.95** に戻します。

```
# ceph pg set_full_ratio 0.95
```

6. パラメーターが **0.95** に設定されたことを確認します。

```
# ceph pg dump | grep -i full
full_ratio 0.95
```

その他の参照先

- 「Full OSDs」
- 「Nearfull OSDs」

第6章 プレイスメントグループのトラブルシューティング

本章では、Ceph プレイスメントグループ (PG) に関する一般的な問題の解決方法を説明します。

はじめに

- ネットワーク接続を確認してください。詳細は *3章 ネットワーク問題のトラブルシューティング* を参照してください。
- モニターが quorum を形成可能であることを確認します。モニター関連の問題のトラブルシューティングについては、*4章 モニターのトラブルシューティング* を参照してください。
- 正常な OSD がすべて **up** かつ **in** であること、またバックフィルと復旧プロセスが完了していることを確認します。OSD 関連の一般的な問題のトラブルシューティングについては、 *5章 OSD のトラブルシューティング* を参照してください。

6.1. プレイスメントグループに関する一般的なエラーメッセージ

以下のテーブルでは、**ceph health detail** コマンドで返される最も一般的なエラーメッセージを示しています。各エラーの内容を説明し、その解決方法が記載された対応セクションも表示しています。

また、最適ではない状態に陥ったプレイスメントグループを一覧表示することもできます。詳細は 「**stale**、**inactive**、**unclean** 状態のプレイスメントグループ」 を参照してください。

表6.1 プレイスメントグループに関するエラーメッセージ

エラーメッセージ	参照先
HEALTH_ERR	
pgs down	「プレイスメントグループが **down**」
pgs inconsistent	「Inconsistent プレイスメントグループ」
scrub errors	「Inconsistent プレイスメントグループ」
HEALTH_WARN	
pgs stale	「Stale プレイスメントグループ」
unfound	「Unfound オブジェクト」

6.1.1. Stale プレイスメントグループ

ceph health コマンドでは、**stale** の プレイスメントグループ (PG) を一覧表示する場合があります。

```
HEALTH_WARN 24 pgs stale; 3/300 in osds are down
```

エラー内容

プレイスメントグループがアクティブなプライマリー OSD からステータス更新を受け取らない、またはプライマリー OSD が **down** であると他の OSD がレポートした場合、モニターはこのプレイスメントグループを **stale** とマークします。

通常は、ユーザーがストレージクラスターを起動してからピアリングプロセスが完了するまで、PG は **stale** 状態に入ります。ただし、PG が想定期間よりも長く **stale** に留まっている場合は、それら PG のプライマリー OSD が **down** となっているか、モニターに PG 統計情報を報告していない可能性があります。**stale** 状態の PG を保存しているプライマリー OSD が **up** に戻ると、Ceph は PG の復旧を開始します。

mon_osd_report_timeout では、OSD が PG 統計情報をモニターに報告する頻度を設定します。デフォルトではこのパラメーターは **0.5** に設定され、OSD は 0.5 秒ごとに統計情報を報告します。

解決方法

1. どの PG が **stale** にあり、それらがどの OSD に保存されているかを特定します。エラーメッセージには、以下のような情報が含まれます。

```
# ceph health detail
HEALTH_WARN 24 pgs stale; 3/300 in osds are down
...
pg 2.5 is stuck stale+active+remapped, last acting [2,0]
...
osd.10 is down since epoch 23, last address
192.168.106.220:6800/11080
osd.11 is down since epoch 13, last address
192.168.106.220:6803/11539
osd.12 is down since epoch 24, last address
192.168.106.220:6806/11861
```

2. **down** とマークされている OSD の問題を解決します。詳細は、「(1 つ以上の) OSDs Are Down」 を参照してください。

その他の参照先

- Red Hat Ceph Storage 2 の Administration Guide に記載の Monitoring Placement Group States のセクション。

6.1.2. Inconsistent プレイスメントグループ

プレイスメントグループの中には **active + clean + inconsistent** とマークされるものがあり、**ceph health detail** は以下のようなエラーメッセージを返します。

```
HEALTH_ERR 1 pgs inconsistent; 2 scrub errors
pg 0.6 is active+clean+inconsistent, acting [0,1,2]
2 scrub errors
```

エラー内容
Ceph がプレイスメントグループ内の 1 つ以上のオブジェクトのレプリカで不一致を検出すると、そのプレイスメントグループを **inconsistent** をマークします。一般的な不一致は以下のものです。

- オブジェクトのサイズが正しくない。
- 復旧後にオブジェクトがいずれかのレプリカにない。

ほとんどの場合、スクラビング中のエラーがプレイスメントグループ内での不一致を引き起こします。

解決方法

1. どのプレイスメントグループが **inconsistent** 状態にあるか判定します。

```
# ceph health detail
HEALTH_ERR 1 pgs inconsistent; 2 scrub errors
pg 0.6 is active+clean+inconsistent, acting [0,1,2]
2 scrub errors
```

2. プレイスメントグループが **inconsistent** になっている原因を判定します。

   a. プレイスメントグループで詳細なスクラビングプロセスを開始します。

```
ceph pg deep-scrub <id>
```

   <id> を **inconsistent** プレイスメントグループの ID で置き換えます。例を示します。

```
# ceph pg deep-scrub 0.6
instructing pg 0.6 on osd.0 to deep-scrub
```

   b. **ceph -w** の出力で該当するプレイスメントグループに関連するメッセージを検索します。

```
ceph -w | grep <id>
```

   <id> を **inconsistent** プレイスメントグループの ID で置き換えます。例を示します。

```
# ceph -w | grep 0.6
2015-02-26 01:35:36.778215 osd.106 [ERR] 0.6 deep-scrub stat
mismatch, got 636/635 objects, 0/0 clones, 0/0 dirty, 0/0 omap,
0/0 hit_set_archive, 0/0 whiteouts, 1855455/1854371 bytes.
2015-02-26 01:35:36.788334 osd.106 [ERR] 0.6 deep-scrub 1 errors
```

3. 出力に以下のようなエラーメッセージが含まれる場合は、**inconsistent** プレイスメントグループの修復が可能です。詳細は、「Inconsistent プレイスメントグループの修復」 を参照してください。

```
<pg.id> shard <osd>: soid <object> missing attr _, missing attr
<attr type>
<pg.id> shard <osd>: soid <object> digest 0 != known digest
<digest>, size 0 != known size <size>
<pg.id> shard <osd>: soid <object> size 0 != known size <size>
<pg.id> deep-scrub stat mismatch, got <mismatch>
<pg.id> shard <osd>: soid <object> candidate had a read error,
digest 0 != known digest <digest>
```

4. 出力に以下のようなエラーメッセージが含まれる場合は、**inconsistent** プレイスメントグループの修復を行うとデータが失われる可能性があるので、安全ではありません。この場合は、サポートチケットを開いてください。詳細は *7章Red Hat サポートへの連絡* を参照してください。

```
<pg.id> shard <osd>: soid <object> digest <digest> != known digest
<digest>
```

```
<pg.id> shard <osd>: soid <object> omap_digest <digest> != known
omap_digest <digest>
```

その他の参照先

- 「Inconsistent プレイスメントグループの修復」
- 「不一致の一覧表示」
- Red Hat Ceph Storage 2 の Architecture Guide に記載の Scrubbing のセクション。
- Red Hat Ceph Storage 2 の Configuration Guide に記載の Scrubbing のセクション。

6.1.3. Unclean プレイスメントグループ

ceph health コマンドで以下のようなエラーメッセージが返されます。

```
HEALTH_WARN 197 pgs stuck unclean
```

エラー内容
Ceph 設定ファイルの **mon_pg_stuck_threshold** パラメーターで指定されている秒数間、プレイスメントグループが **active+clean** 状態を達成しないと、Ceph はそのプレイスメントグループを **unclean** とマークします。**mon_pg_stuck_threshold** のデフォルト値は、**300** 秒です

プレイスメントグループが **unclean** の場合、**osd_pool_default_size** パラメーターで指定された回数、複製されていないオブジェクトがそのプレイスメントグループに含まれています。**osd_pool_default_size** のデフォルト値は **3** で、Ceph はレプリカを 3 つ作成することになります。

通常、**unclean** プレイスメントグループが示すのは、**down** となっている OSD があるということです。

解決方法

1. どの OSD が **down** になっているか判定します。

```
# ceph osd tree
```

2. OSD の問題を解決します。詳細は 「(1 つ以上の) OSDs Are Down」 を参照してください。

その他の参照先

- 「**stale**、**inactive**、**unclean** 状態のプレイスメントグループ」

6.1.4. Inactive プレイスメントグループ

ceph health コマンドで以下のようなエラーメッセージが返されます。

```
HEALTH_WARN 197 pgs stuck inactive
```

エラー内容
Ceph 設定ファイルの **mon_pg_stuck_threshold** パラメーターで指定されている秒数間、プレイスメントグループがアクティブでなかった場合、Ceph はそのプレイスメントグループを **inactive** とマークします。**mon_pg_stuck_threshold** のデフォルト値は、**300** 秒です

通常、**inactive** プレイスメントグループが示すのは、**down** となっている OSD があるということです。

解決方法

1. どの OSD が **down** になっているか判定します。

```
# ceph osd tree
```

2. OSD の問題を解決します。詳細は 「(1 つ以上の) OSDs Are Down」 を参照してください。

その他の参照先

- 「**stale**、**inactive**、**unclean** 状態のプレイスメントグループ」

6.1.5. プレイスメントグループが down

ceph health detail コマンドで、プレイスメントグループが **down** になっていると報告されます。

```
HEALTH_ERR 7 pgs degraded; 12 pgs down; 12 pgs peering; 1 pgs recovering; 6
pgs stuck unclean; 114/3300 degraded (3.455%); 1/3 in osds are down
...
pg 0.5 is down+peering
pg 1.4 is down+peering
...
osd.1 is down since epoch 69, last address 192.168.106.220:6801/8651
```

エラー内容
場合によっては、ピアリングプロセスがブロックされ、プレイスメントグループがアクティブかつ使用可能になりません。通常、OSD の失敗がピアリングの失敗を引き起こしています。

解決方法
ピアリングプロセスをブロックしているものを判定します。

```
ceph pg <id> query
```

<id> を **down** となっているプレイスメントグループの ID で置き換えます。例を示します。

```
# ceph pg 0.5 query

{ "state": "down+peering",
  ...
  "recovery_state": [
        { "name": "Started\/Primary\/Peering\/GetInfo",
          "enter_time": "2012-03-06 14:40:16.169679",
          "requested_info_from": []},
        { "name": "Started\/Primary\/Peering",
          "enter_time": "2012-03-06 14:40:16.169659",
          "probing_osds": [
                0,
                1],
          "blocked": "peering is blocked due to down osds",
          "down_osds_we_would_probe": [
                1],
          "peering_blocked_by": [
```

```
                { "osd": 1,
                  "current_lost_at": 0,
                  "comment": "starting or marking this osd lost may let us
proceed"}]},
        { "name": "Started",
          "enter_time": "2012-03-06 14:40:16.169513"}
    ]
}
```

recovery_state セクションには、ピアリングプロセスがブロックされている原因が表示されています。

- 出力に **peering is blocked due to down osds** というエラーメッセージがある場合は、「(1 つ以上の) OSDs Are Down」 を参照してください。
- これら以外のエラーメッセージがある場合は、サポートチケットを開いてください。詳細は 7 章*Red Hat サポートへの連絡* を参照してください。

その他の参照先

- Red Hat Ceph Storage 2 の Administration Guide に記載の Peering のセクション。

6.1.6. Unfound オブジェクト

ceph health コマンドで、**unfound**というキーワードを含む以下のようなエラーメッセージが返されます。

```
HEALTH_WARN 1 pgs degraded; 78/3778 unfound (2.065%)
```

エラー内容
Ceph は、オブジェクトもしくはそれらの新規コピーが存在することが分かっているものの、見つけられない場合、それらのオブジェクトを **unfound** とマークします。その結果、Ceph はそれらのオブジェクトを復旧できず、復旧プロセスを進めることができません。

例
プレイスメントグループがデータを **osd.1** と **osd.2** に保存します。

1. **osd.1** が **down** となります。
2. **osd.2** が書き込み操作を処理します。
3. **osd.1** が **up** となります。
4. **osd.1** と **osd.2** のピアリングプロセスが開始されます。**osd.1** にないオブジェクトが復旧のキューに登録されます。
5. Ceph が新規オブジェクトをコピーする前に、**osd.2** が **down** となります。

この結果、**osd.1** はオブジェクトが存在することは分かっているものの、そのオブジェクトを保存している OSD がない状態になります。

このシナリオでは、Ceph は失敗しているノードが再度アクセス可能になること待機し、**unfound** オブジェクトが復旧プロセスをブロックします。

解決方法

1. **unfound** オブジェクトが含まれているプレイスメントグループを判定します。

```
# ceph health detail
HEALTH_WARN 1 pgs recovering; 1 pgs stuck unclean; recovery 5/937611
objects degraded (0.001%); 1/312537 unfound (0.000%)
pg 3.8a5 is stuck unclean for 803946.712780, current state
active+recovering, last acting [320,248,0]
pg 3.8a5 is active+recovering, acting [320,248,0], 1 unfound
recovery 5/937611 objects degraded (0.001%); **1/312537 unfound
(0.000%)**
```

2. そのプレイスメントグループについての詳細情報を表示します。

```
# ceph pg <id> query
```

<id> を **unfound** オブジェクトが含まれているプレイスメントグループの ID で置き換えます。例を示します。

```
# ceph pg 3.8a5 query
{ "state": "active+recovering",
  "epoch": 10741,
  "up": [
        320,
        248,
        0],
  "acting": [
        320,
        248,
        0],
<snip>
  "recovery_state": [
        { "name": "Started\/Primary\/Active",
          "enter_time": "2015-01-28 19:30:12.058136",
          "might_have_unfound": [
                { "osd": "0",
                  "status": "already probed"},
                { "osd": "248",
                  "status": "already probed"},
                { "osd": "301",
                  "status": "already probed"},
                { "osd": "362",
                  "status": "already probed"},
                { "osd": "395",
                  "status": "already probed"},
                { "osd": "429",
                  "status": "osd is down"}],
          "recovery_progress": { "backfill_targets": [],
              "waiting_on_backfill": [],
              "last_backfill_started": "0\/\/0\/\/-1",
              "backfill_info": { "begin": "0\/\/0\/\/-1",
                  "end": "0\/\/0\/\/-1",
                  "objects": []},
              "peer_backfill_info": [],
              "backfills_in_flight": [],
              "recovering": [],
```

```
                "pg_backend": { "pull_from_peer": [],
                    "pushing": []}},
              "scrub": { "scrubber.epoch_start": "0",
                  "scrubber.active": 0,
                  "scrubber.block_writes": 0,
                  "scrubber.finalizing": 0,
                  "scrubber.waiting_on": 0,
                  "scrubber.waiting_on_whom": []}},
            { "name": "Started",
              "enter_time": "2015-01-28 19:30:11.044020"}],
```

might_have_unfound セクションには、Ceph が **unfound** オブジェクトを見つけようとした OSD が含まれます。

- **already probed** ステータスは、Ceph が その **unfound** オブジェクトを OSD で見つけられないことを示します。
- **osd is down** ステータスは、Ceph が OSD に連絡できないことを示します。

3. **down** とマークされている OSD の問題を解決します。詳細は、「(1 つ以上の) OSDs Are Down」 を参照してください。
4. OSD を **down** としている問題を解決できない場合は、サポートチケットを開いてください。詳細は *7章Red Hat サポートへの連絡* を参照してください。

6.2. STALE、 INACTIVE、 UNCLEAN 状態のプレイスメントグループ

プレイスメントグループは失敗後、**degraded** または **peering** といった状態になります。この状態は、失敗空の復旧プロセスによる正常な進行状況を示しています。

ただし、これらの状態にプレイスメントグループが想定よりも長い時間留まっている場合は、より大きな問題である可能性があります。プレイスメントグループが最適でない状態に留まっている場合は、モニターが報告します。

以下の表では、これらの状態とその説明を示しています。

State	説明	一般的な原因	参照先
inactive	PG が読み取り/書き込みリクエストを実行できない。	• ピアリングの問題	「Inactive プレイスメントグループ」
unclean	望ましい回数複製されていないオブジェクトが PG に含まれている。なんらかの原因で PG の復旧ができない。	• **unfound** オブジェクト • OSDs が **down** • 設定が間違っている	「Unclean プレイスメントグループ」

State	説明	一般的な原因	参照先
stale	**ceph-osd** デーモンが PG のステータスを更新していない。	• OSDs が **down**	「Stale プレイスメントグループ」

Ceph 設定ファイル内の **mon_pg_stuck_threshold** パラメーターで設定された秒数が経過すると、プレイスメントグループ は**inactive**、**unclean**、または **stale** であるとみなされます。

stuck した PG を一覧表示します。

```
# ceph pg dump_stuck inactive
# ceph pg dump_stuck unclean
# ceph pg dump_stuck stale
```

その他の参照先

- Red Hat Ceph Storage 2 の Administration Guide に記載の Monitoring Placement Group States のセクション。

6.3. 不一致の一覧表示

rados ユーティリティーを使ってオブジェクトの各レプリカにおける不一致を一覧表示することができます。 **--format=json-pretty** オプションを使うとより詳細な出力が返されます。

以下を一覧表示することが可能です。

- プール内で一致しないプレイスメントグループ
- プレイスメントグループ内で一致しないオブジェクト
- プレイスメントグループ内で一致しないスナップショットのセット

プール内で一致しないプレイスメントグループの一覧表示

```
rados list-inconsistent-pg <pool> --format=json-pretty
```

例えば、**data** というプール内で一致しないプレイスメントグループを一覧表示するには、以下を実行します。

```
# rados list-inconsistent-pg data --format=json-pretty
[0.6]
```

プレイスメントグループ内で一致しないオブジェクトの一覧表示

```
rados list-inconsistent-obj <placement-group-id>
```

例えば、ID が **0.6** であるプレイスメントグループ内で一致しないオブジェクトを一覧表示するには、以下を実行します。

```
# rados list-inconsistent-obj 0.6
{
```

```
    "epoch": 14,
    "inconsistents": [
        {
            "object": {
                "name": "image1",
                "nspace": "",
                "locator": "",
                "snap": "head",
                "version": 1
            },
            "errors": [
                "data_digest_mismatch",
                "size_mismatch"
            ],
            "union_shard_errors": [
                "data_digest_mismatch_oi",
                "size_mismatch_oi"
            ],
            "selected_object_info": "0:602f83fe:::foo:head(16'1
client.4110.0:1 dirty|data_digest|omap_digest s 968 uv 1 dd e978e67f od
ffffffff alloc_hint [0 0 0])",
            "shards": [
                {
                    "osd": 0,
                    "errors": [],
                    "size": 968,
                    "omap_digest": "0xffffffff",
                    "data_digest": "0xe978e67f"
                },
                {
                    "osd": 1,
                    "errors": [],
                    "size": 968,
                    "omap_digest": "0xffffffff",
                    "data_digest": "0xe978e67f"
                },
                {
                    "osd": 2,
                    "errors": [
                        "data_digest_mismatch_oi",
                        "size_mismatch_oi"
                    ],
                    "size": 0,
                    "omap_digest": "0xffffffff",
                    "data_digest": "0xffffffff"
                }
            ]
        }
    ]
}
```

不一致の原因を判定するには、以下のフィールドが重要になります。

- **name**: 一致しないレプリカのあるオブジェクト名
- **nspace**: プールの論理分離であるネームスペース。デフォルトでは空白です。

- **locator**: 配置の際にオブジェクト名の代わりとして使用されるキー。
- **snap**: オブジェクトのスナップショット ID。オブジェクトの唯一書き込み可能なバージョンは、**head** と呼ばれます。オブジェクトがクローンの場合、このフィールドにはシーケンシャル ID が含まれます。
- **version**: 一致しないレプリカのあるオブジェクトのバージョン ID。オブジェクトへの書き込み操作がある度にこれが増加します。
- **errors**: シャード間に不一致があることを示すエラー一覧。どのシャードが間違っているかは判定しません。このエラーの詳細については、**shard** アレイを参照してください。
  - **data_digest_mismatch**: ある OSD から読み込まれたレプリカのダイジェストが別の OSD とは異なっています。
  - **size_mismatch**: クローンまたは **head** オブジェクトのサイズが想定値に一致しません。
  - **read_error**: ディスクのエラーで発生した可能性が高い不一致を示すエラーです。
- **union_shard_error**: シャードに固有の全エラーの集合。これらのエラーは、問題のあるシャードに関連しています。**oi** で終わるエラーは、問題のあるオブジェクトからの情報を選択したオブジェクトのものと比較する必要があることを示しています。このエラーの詳細は、**shard** アレイを参照してください。
  上記の例では、**osd.2** に保存されているオブジェクトレプリカのダイジェストが、**osd.0** および **osd.1** に保存されているレプリカのものとは異なっています。具体的には、レプリカのダイジェストが、**osd.2** から読み込まれたシャードからの計算による **0xffffffff** ではなく、**0xe978e67f** になっています。また、**osd.2** から読み込まれたレプリカのサイズは 0 ですが、**osd.0** と **osd.1** がレポートしているサイズは 968 です。

プレイスメントグループ内で一致しないスナップショットセットの一覧表示

```
rados list-inconsistent-snapset <placement-group-id>
```

例えば、ID が **0.23** であるプレイスメントグループ内で一致しないスナップショットのセット (**snapsets**) を一覧表示するには、以下を実行します。

```
# rados list-inconsistent-snapset 0.23 --format=json-pretty
{
    "epoch": 64,
    "inconsistents": [
        {
            "name": "obj5",
            "nspace": "",
            "locator": "",
            "snap": "0x00000001",
            "headless": true
        },
        {
            "name": "obj5",
            "nspace": "",
            "locator": "",
            "snap": "0x00000002",
            "headless": true
        },
        {
```

```
            "name": "obj5",
            "nspace": "",
            "locator": "",
            "snap": "head",
            "ss_attr_missing": true,
            "extra_clones": true,
            "extra clones": [
                2,
                1
            ]
        }
    ]
```

このコマンドは、以下のエラーを返しています。

- **ss_attr_missing**: 1 つ以上の属性がありません。属性は、キーと値のペアー覧としてスナップショットセットにエンコードされた情報です。
- **ss_attr_corrupted**: 1 つ以上の属性のデコードに失敗しました。
- **clone_missing**: クローンがありません。
- **snapset_mismatch**: スナップショットセット自体に不一致があります。
- **head_mismatch**: スナップショットセットは **head** の存在の有無を示しますが、スクラビングの結果ではその逆が示されます。
- **headless**: スナップショットセットの **head** がありません。
- **size_mismatch**: クローンまたは **head** オブジェクトのサイズが想定値に一致しません。

その他の参照先

- 「Inconsistent プレイスメントグループ」
- 「Inconsistent プレイスメントグループの修復」

6.4. INCONSISTENT プレイスメントグループの修復

詳細なスクラビング中のエラーにより、プレイスメントグループに不一致が含まれる場合があります。Ceph はこれらのプレイスメントグループを **inconsistent** とレポートします。

```
HEALTH_ERR 1 pgs inconsistent; 2 scrub errors
pg 0.6 is active+clean+inconsistent, acting [0,1,2]
2 scrub errors
```

警告

修復が可能なのは、特定の不一致のみです。Ceph のログに以下のエラーが含まれる場合は、そのプレイスメントグループを修復しないでください。

```
<pg.id> shard <osd>: soid <object> digest <digest> != known
digest <digest>
<pg.id> shard <osd>: soid <object> omap_digest <digest> !=
known omap_digest <digest>
```

代わりにサポートチケットを開いてください。詳細は *7章Red Hat サポートへの連絡* を参照してください。

inconsistent プレイスメントグループを修復します。

```
ceph pg repair <id>
```

<id> を **inconsistent** プレイスメントグループの ID で置き換えます。

その他の参照先

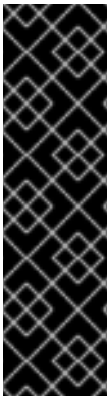
- 「Inconsistent プレイスメントグループ」
- 「不一致の一覧表示」

6.5. PG カウントの増加

プレイスメントグループ (PG) の数が十分でないと、Ceph クラスターおよびデータ配分のパフォーマンスに影響が出ます。これは、**nearfull osds** エラーメッセージの主な原因の 1 つです。

推奨される数は、OSD あたり 100 から 300 の PG です。クラスターにさらに OSD を追加すると、この比率を下げることができます。

pg_num と **pgp_num** のパラメーターで PG カウントを決定します。これらのパラメーターはプールごとに設定されるので、PG カウントが少ないプールは個別に調整する必要があります。



重要

PG カウントを増やす作業は、Ceph クラスターで行う最も集中的なプロセスになります。これは落ち着いて組織的に実行しないと、パフォーマンスに重大な影響を与えかねません。**pgp_num** を増やすと、このプロセスを停止したり戻したりすることはできず、完了させる必要があります。

PG カウントを増やす場合は業務の重要な処理時間外に実行し、パフォーマンスに影響が出る可能性を全クライアントに通知することを検討してください。

クラスターが **HEALTH_ERR** 状態にある場合は、PG カウントを変更しないでください。

手順: PG カウントの増加

1. 個別の OSD および OSD ホストへのデータ配分およびリカバリーの影響を低減します。

   a. **osd max backfills**、**osd_recovery_max_active**、および **osd_recovery_op_priority** パラメーターの値を低くします。

   ```
   # ceph tell osd.* injectargs '--osd_max_backfills 1 --
   osd_recovery_max_active 1 --osd_recovery_op_priority 1'
   ```

   b. 簡易および詳細なスクラブを無効にします。

   ```
   # ceph osd set noscrub
   # ceph osd set nodeep-scrub
   ```

2. Ceph Placement Groups (PGs) per Pool Calculator を利用して **pg_num** と **pgp_num** のパラメーターの値を計算します。

3. **pg_num** の値を希望する数値になるまで少しずつ増やします。

   a. 最初に増やす値を決定します。2 のべき乗の低い数を使用し、クラスターへの影響が分かったら、これを増やします。最適な値は、プールのサイズ、OSD カウント、クライアントの I/O 負荷によって異なります。

   b. **pg_num** の値を増やします。

   ```
   ceph osd pool set <pool> pg_num <value>
   ```

   プール名と新しい値を指定します。例を示します。

   ```
   # ceph osd pool set data pg_num 4
   ```

   c. クラスターのステータス監視します。

   ```
   # ceph -s
   ```

   PG の状態は **creating** から **active+clean** に替わります。すべての PG が **active+clean** 状態になるまで待機します。

4. **pgp_num** の値を希望する数値になるまで少しずつ増やします。

   a. 最初に増やす値を決定します。2 のべき乗の低い数を使用し、クラスターへの影響が分かったら、これを増やします。最適な値は、プールのサイズ、OSD カウント、クライアントの I/O 負荷によって異なります。

   b. **pgp_num** の値を増やします。

   ```
   ceph osd pool set <pool> pgp_num <value>
   ```

   プール名と新しい値を指定します。例を示します。

   ```
   # ceph osd pool set data pgp_num 4
   ```

   c. クラスターのステータス監視します。

   ```
   # ceph -s
   ```

- PG の状態は、**peering**、**wait_backfill**、**backfilling**、**recover** などに替わります。すべての PG が **active+clean** 状態になるまで待機します。

5. PG カウントが足りないすべてのプールで上記のステップを繰り返します。
6. **osd max backfills**、**osd_recovery_max_active**、および **osd_recovery_op_priority** をデフォルト値に設定します。

```
# ceph tell osd.* injectargs '--osd_max_backfills 1 --
osd_recovery_max_active 3 --osd_recovery_op_priority 3'
```

7. 簡易および詳細なスクラブを有効にします。

```
# ceph osd unset noscrub
# ceph osd unset nodeep-scrub
```

その他の参照先

- 「Nearfull OSDs」
- Red Hat Ceph Storage 2 の Administration Guide に記載の Monitoring Placement Group States のセクション。

第7章 RED HAT サポートへの連絡

本ガイドでの情報で問題が解決できない場合は、本章の説明を参考にして Red Hat サポートサービスまでご連絡ください。

7.1. RED HAT サポートエンジニアへの情報提供

Red Hat Ceph ストレージに関連する問題がご自分で解決できない場合は、Red Hat サポートサービスまでご連絡ください。その場合、詳しい情報をご提供いただくと、エンジニアによる問題解決が早まる可能性が高くなります。

手順: Red Hat サポートエンジニアへの情報提供

1. Red Hat カスタマーポータル でサポートケースを作成します。
2. 可能な場合は、チケットに **sosreport** を添付してください。詳細は、Red Hat Enterprise Linux 4.6 以降における sosreport の役割と取得方法 を参照してください。
3. セグメンテーション違反で Ceph デーモンが失敗している場合は、ヒューマンリーダブルなコアダンプファイルの生成を検討してください。詳細は、「ヒューマンリーダブルなコアダンプファイルの生成」 を参照してください。

7.2. ヒューマンリーダブルなコアダンプファイルの生成

Ceph デーモンがセグメンテーション違反で予期せず終了する場合は、その失敗についての情報を収集し、Red Hat サポートエンジニアに提供してください。

この情報があると初期調査が迅速化されます。また、サポートエンジニアがコアダンプファイルからの情報と Red Hat Ceph ストレージの既知の問題を比較することもできます。

はじめに

1. **ceph-debuginfo** パッケージがインストールされていない場合は、これをインストールします。
   a. **ceph-debuginfo** パッケージが格納されているリポジトリーを有効にします。

   ```
   subscription-manager repos --enable=rhel-7-server-rhceph-2-
   <daemon>-debug-rpms
   ```

   ノードのタイプによって、**<daemon>** を **osd** または **mon** で置き換えます。

   b. **ceph-debuginfo** パッケージをインストールします。

   ```
   # yum install ceph-debuginfo
   ```

2. **gdb** パッケージがインストールされていることを確認します。インストールされていない場合は、これをインストールします。

   ```
   # yum install gdb
   ```

手順: ヒューマンリーダブルなコアダンプファイルの生成

1. Ceph のコアダンプファイルの生成を有効にします。

a. **/etc/systemd/system.conf** ファイルに以下のパラメーターを追加して、コアダンプファイルの **ulimits** を設定します。

```
DefaultLimitCORE=infinity
```

b. Ceph デーモンのサービスファイルで **PrivateTmp=true** パラメーターをコメントアウトします。このファイルは、デフォルトでは **/lib/systemd/system/<cluster-name>-<daemon>@.service** にあります。

```
# PrivateTmp=true
```

c. **suid_dumpable** フラグを **2** に設定して、Ceph デーモンがコアダンプファイルを生成できるようにします。

```
# sysctl fs.suid_dumpable=2
```

d. コアダンプファイルの場所を調整します。

```
# sysctl kernel.core_pattern=/tmp/core
```

e. 変更を反映するために **systemd** サービスをリロードします。

```
# systemctl daemon-reload
```

f. 変更を反映するために、Ceph デーモンを再起動します。

```
systemctl restart ceph-<daemon>@<ID>
```

デーモンのタイプ (**osd** または **mon**) とその ID (OSD の場合は番号、モニターの場合は短いホスト名) を以下のように指定します。

```
# systemctl restart ceph-osd@1
```

2. 失敗を再現します。例えば、デーモンの起動を試行します。

3. GNU デバッガー (GDB) を使って、アプリケーションのコアダンプファイルからリーダブルなバックトレースを生成します。

```
gdb /usr/bin/ceph-<daemon> /tmp/core.<PID>
```

以下のようにデーモンのタイプと失敗したプロセスの PID を指定します。

```
$ gdb /usr/bin/ceph-osd /tmp/core.123456
```

4. GDB のコマンドプロンプトで **thr a a bt** を入力し、**backtrace** コマンドをプロセスの全スレッドに適用します。

```
(gdb) thr a a bt
```

5. 上記のコマンドの出力をサポートチケットにコピーします。

その他の参照先

- Red Hat カスタマーポータルの gdb を使用して、アプリケーションコアから読み取り可能なバックトレースを生成する方法
- Red Hat カスタマーポータルの アプリケーションがクラッシュまたはセグメンテーション違反が発生した時にコアファイルのダンプを有効にする

付録A サブシステムのデフォルトのロギングレベル

サブシステム	ログレベル	メモリーレベル
asok	1	5
auth	1	5
buffer	0	0
client	0	5
context	0	5
crush	1	5
default	0	5
filer	0	5
filestore	1	5
finisher	1	5
heartbeatmap	1	5
javaclient	1	5
journaler	0	5
journal	1	5
lockdep	0	5
mds balancer	1	5
mds locker	1	5
mds log expire	1	5
mds log	1	5
mds migrator	1	5
mds	1	5
monc	0	5

サブシステム	ログレベル	メモリーレベル
mon	1	5
ms	0	5
objclass	0	5
objectcacher	0	5
objecter	0	0
optracker	0	5
osd	0	5
paxos	0	5
perfcounter	1	5
rados	0	5
rbd	0	5
rgw	1	5
throttle	1	5
timer	0	5
tp	0	5