

## クラスタ構成に関するトラブル

原因		症状	対策
分類1	分類2		
プロセスが正常に起動しない	リカバリに失敗する	【エラーコード】 20000～ 【内容】 リカバリに関するエラー群 【メッセージ】 リカバリエラーカテゴリの書式に従う 【サーバ】 100002, 100003, 100004, 100005	操作手順誤りによるものと内部異常による原因等が考えられます。詳細は、「リカバリ処理に関するトラブル」を参照してください。
	定義ファイル(gs_cluster.json, gs_node.json)の記述内容に誤りがある	【内容】 コンフィグ設定失敗 【メッセージ】 失敗を検知した最初のエラーに関して、必要項目漏れか、値制限範囲外か、型不一致エラーかのいずれかの失敗原因を表示	gs_node.json/gs_cluster.jsonの設定に誤りがあります。エラーメッセージ及びマニュアルを確認して、定義ファイルの必須項目漏れ、値制限、型不一致の問題が無いかを確認し、修正後、再起動してください。
	gs_node.jsonに記述したサービスアドレスが、DNS名前解決で失敗する	【サーバ】 130000 【内容】 サービスアドレス設定失敗 【メッセージ】 失敗したアドレス、ポート、エラー原因となったプラットフォームエラー番号(多くのケースは-2(EAI_NONAME))	gs_node.jsonの各serviceAddressが正常かどうか確認してください。設定していない場合は、/etc/hostsに記載されたアドレスを使いますので、こちらをご確認下さい。
	gs_cluster.jsonに記述したマルチキャストアドレスに誤りがある	【サーバ】 130006 【内容】 マルチキャストアドレス設定失敗 【メッセージ】 失敗したアドレス、ポート、エラー原因となったプラットフォームエラー番号(22(EINVAL))	gs_cluster.jsonの各notificationAddressが正常(224.0.0.0 ~ 239.255.255.255 )かどうか確認してください。ポートは分けることを推奨しますが、重複しても動作します。
	環境設定不備によりマルチキャストが実行できない	【サーバ】 130006 【内容】 マルチキャストアドレス設定失敗 【メッセージ】 失敗したアドレス、ポート、エラー原因となったプラットフォームエラー番号(19(ENODEV))	最初に、ループバック(lo)以外に有効なネットワーク接続が存在し、有効であるかどうか確認してください。次に、マルチキャストルーティング設定を確認してください。特に指定しない場合はデフォルトゲートウェイ設定が使われますが、この設定が正しいかどうか確認してください。
	サーバとデータベース間のバージョンが一致していない	【サーバ】 80022 【内容】 データベースバージョン不一致 【メッセージ】 モジュール 行番号 エラーコード ログファイルバージョン	エラーメッセージに記載されたバージョン番号を確認して、有効なバージョンのデータベースに移行してください。有効なバージョンに関しては「クライアントとDBの互換性」を、移行方法については『GridDB移行手順書』を参照ください。
プロセス起動中にノード状態がABNORMALになる	gs_node.json中のいずれかのポートがこのファイル内で重複しているか、既に同一マシンで起動中プロセスのポートと一致している	【サーバ】 130024 【内容】 ソケットバインド失敗 【メッセージ】 失敗したアドレス、ポート、リトライに要した時間(デフォルト1分) 【外部ツール】 netstat, ポートスキャン等を用いて、重複チェック及び、空きポートを検索	gs_node.jsonに記載されたポートが重複しないことを外部ツール等を用いて確認し、適切な値に修正後、再起動してください。ノード状態がABNORMALの場合、--forceオプションを指定してgs_stopnodeを実行し、ノードを停止してください。
	クラスタ構成対象のノード間でgs_cluster.jsonの値もしくは、クラスタ名、構成ノード数が一致しない	【サーバ】 180046~180050 【内容】 クラスタ構成失敗 【メッセージ】 接続先アドレス番号、クラスタ構成できなかった原因(クラスタ名不一致、コンフィグ不一致、など)	gs_cluster.jsonの全ての値、及び、gs_joinclusterで指定するクラスタ名、構成ノード数が、構成対象のノード間で全て正しいことを確認して下さい。

クラスタが一定期間経過しても構成されない	クラスタ間のサーバの有効バイナリバージョンが不一致である	【サーバ】 40044 【内容】 サーババージョン不一致 【メッセージ】 接続先アドレス番号、自分のサーババージョン番号、接続先のサーババージョン番号	エラー記載されたバージョン番号を確認して、有効なバージョンのサーババイナリを利用して、クラスタ再起動、再構成し直して下さい。
	(同一マルチキャストアドレスで)同一クラスタ名の別クラスタが既に存在することを、現クラスタのマスタノードが検知した	【サーバ】 180018 【内容】 重複クラスタ検知 【メッセージ】 既存クラスタのマスタノードの接続アドレス、ポート番号	重複検知した状態で継続することでデータ破壊に繋がる可能性がありますので、重複を検知した側のクラスタを停止 (gs_stopcluster)させるか、通知先のクラスタを停止させ、(2クラスタ以上の場合はこちらを繰り返して)同一サブネット上に、同一クラスタ名のクラスタが一つになるように調整を行ってください。なお、gs_cluster.jsonのクラスタ名を設定する場合は、サブネット内でユニークとなるクラスタ名(及びマルチキャストアドレス)を設定することを強く推奨します。
	gs_node.json中のサービスアドレスが有効ではない	【サーバ】 トレースには特に何も残らないが、2ノード以上のクラスタが一定期間経過しても構成されない	各サービスのアドレスの正当性を確認してください。1ノードの場合は起動しますが、2ノード以上だと、ノード間通信が行えないためにクラスタが構成されません。
	クラスタ構成のためのマルチキャストが物理的に届かない	【外部ツール】 tcpdumpやnetstatでgs_cluster.jsonの /cluster/notificationAddress, notificationPortで指定したマルチキャストアドレス、ポートが受信できているかの確認	クラスタを構成する全てのノードが同一サブネット内で起動されていることを確認して下さい。もし、異なるサブネット間での接続が必要な場合は、外部ツールを用いて接続して下さい。また、該当マルチキャストがファイアウォールで遮断されている可能性もありますので、そちらの設定も確認して下さい。
クラスタへのクライアント接続が失敗する	同一ネットワーク内に同一クラスタ名を持つクラスタが存在するため、接続先が混線している	【サーバ】 コンテナ取得失敗など不定だが、この場合、クラスタ中のいずれかで以下が記録される 【サーバ】 10005 【内容】 重複クラスタ検知 【メッセージ】 既存クラスタのマスタノードの接続アドレス、ポート番号	クラスタ名が同一サブネット内でユニークであることをまず確認して下さい。その上で、そのクラスタ名が、クライアントプログラムの接続対象のクラスタ名と一致していることを確認して下さい。
	クライアント-クラスタ間のバイナリバージョンが不一致である	【サーバ】 10054 【内容】 クライアントバージョン不一致 【メッセージ】 接続先アドレス番号、自分のクライアント接続可能なバージョン番号、接続先のクライアントバージョン	エラーメッセージに記載されたバージョン番号を確認して、有効なバージョンのクライアントを利用して下さい。有効なバージョンに関しては「クライアントとDBの互換性」を参照してください。
	クライアントフェイルオーバータイムアウトが発生している	【サーバ】 10008, 10009, 10010 【内容】 フェイルオーバー接続失敗 【クライアント】 クライアントフェイルオーバータイムアウト	クライアントがリトライするたびに本トレースが記録されていることがあります。クライアントフェイルオーバータイムアウト内に接続が確立された場合は、上記ログが残りますが、問題ありません。タイムアウトが発生した場合、最後の失敗原因が記載されます。「クライアントフェイルオーバーに関するトラブル」を参照してください。

	クライアントのクラスタディスカバリのためのマルチキャストが物理的に届かない	【外部ツール】 tcpdumpやnetstatでgs_cluster.jsonの /transaction/notificationAddress, notificationPortで指定したマルチキャストアドレス、ポートが受信できているかの確認	クラスタが存在するサブネットとクライアントが同一サブネット内で起動されていることを確認して下さい。もし、異なるサブネット間での接続が必要な場合は、外部ツールを用いて接続して下さい。また、該当マルチキャストがファイアウォールで遮断されている可能性もありますので、そちらの設定も確認して下さい。
データベースバージョンの不一致	サーバとデータベース間のバージョンの不一致	【サーバ】 80022 【内容】 データベースバージョン不一致 【メッセージ】 モジュール 行番号 エラーコード ログファイルバージョン	エラーメッセージに記載されたバージョン番号を確認して、有効なバージョンのデータベースに移行してください。有効なバージョンに関しては「クライアントとDBの互換性」を、移行方法については『GridDB移行手順書』を参照ください。

gs\_node.json、gs\_cluster.jsonの設定に関しては、『GridDBクイックスタートガイド』の「付録」の「パラメータ一覧」を参考にしてください。  
『移行手順書』は基本サポートサービスで提供されます。インストールメディアやパッケージには同梱されませんのでご注意ください。

### クラスタ拡張、縮小に関するトラブル

原因		症状	対策
分類1	分類2		
増設、離脱可能条件を満たしていないので、コマンドが正常に実行されない	有効ノード数と構成ノード数が一致しない	【サーバ】 180055 【内容】 クラスタ増設/離脱可能状態ではない 【書式】 現クラスタの有効ノード数、構成ノード数 【確認】 gs_stat /cluster/activeCount != /cluster/designatedCount	クラスタ内のノードに障害が発生していないかどうか確認して下さい。もし、障害発生しているならば、そのノードがクラスタに復帰できるならば、まずはそれを試みてください。もし、復帰できない場合、別途新しいノードを用意して、そのノードをクラスタに参加させて下さい。この作業の後、gs_statで、有効ノード数(activeCount)と構成ノード数(designatedCount)が一致したことを確認した上で、再度クラスタ増設/離脱コマンドを実行して下さい。
	構成ノード数以上のノードを参加しようとした	【サーバ】 180045 【内容】 構成ノード数以上のクラスタ構成失敗 【書式】 追加対象ノードにおいて、現クラスタの構成ノード数、マスターアドレス	構成中のクラスタが既にノード数上限(=構成ノード数)に達しているため、新たなノードを参加できない状態です。更に構成ノード数を増やしたい場合は、増設対象のクラスタに参加しているノードに対して、gs_appendclusterを実行してください。
	対象ノードを離脱させることで、現クラスタがデータロストする、或いは、クラスタが解散してしまう	【サーバ】 180030 【内容】 そのノードが離脱できない 【書式】 離脱できない理由(対象ノードを離脱させることでクラスタが過半数ノードを維持できない、一つでもノードを離脱させることでデータロストが発生する)	gs_leavecluster実行によって、データロスト、或いはクラスタ解散の危険がある場合は離脱は実行されません。強制的に実行したい場合は、--forceオプションを付けて実行して下さい。但し、この場合、クラスタがリセットされたり、データロストする可能性がありますので、できるだけ--forceオプションは付けしないでください。

### クライアントフェイルオーバーに関するトラブル

原因		症状	対策
分類1	分類2		
リソース異常に起因するエラー	クライアント・クラスタ間でネットワークに障害が発生した	<p>【サーバ】 130008</p> <p>【内容】 ネットワーク接続エラー</p> <p>【書式】 通信失敗原因</p>	<p>障害が一時的なものであれば、フェイルオーバータイムアウト期間に収まるように、タイムアウトを設定してください。定常的なものであれば、ネットワークの二重化なども検討してください。</p>
タイムアウト設定に起因するエラー	サーバの負荷が非常に高いか、該当トランザクション自体の負荷が高い	<p>【サーバ】 50000, 50001</p> <p>【クライアント】 70000</p> <p>【内容】 トランザクションタイムアウト</p> <p>【書式】 タイムアウト経過期間、パーティション番号、接続アドレス、失敗原因</p>	<p>実行対象のアプリケーションの1ステートメント実行が、トランザクションタイムアウト期間内に収まるかどうかを確認して、適切な値を設定してください。本現象は、サーバ全体として負荷が一時的に高くなっている場合でも発生します。--memoryDetailオプション付きでgs_statを実行することで、通信メッセージの総確保量 (/performance/memoryDetail/work.transactionMessage Total)が出力されますので、定常的に大きくなっていないかを確認して下さい。特に非同期レプリケーションの際、バックアップ側は、処理が一時的に集中し、負荷が高くなる可能性があります。もし、これらが原因でタイムアウトになる場合は、準同期レプリケーションモードに変更してください。</p>
	アプリケーションにおいて、デッドロック、或いは、長期間ロックを保有している	<p>【サーバ】 50000, 50001</p> <p>【クライアント】 70000</p> <p>【内容】 トランザクションタイムアウト</p> <p>【書式】 タイムアウト経過期間、パーティション番号、接続アドレス、失敗原因</p>	<p>--memoryDetailオプション付きでgs_statを実行することで、通信メッセージの総確保量 (/performance/memoryDetail/work.transactionMessage Total)が出力されますので、値を確認して下さい。これが定常的に減らない場合はデッドロック、或いは、ロック待ち状態が発生しています。アプリケーションを見直して、必要ならば終了させてください。無制限にした場合はサーバ側で自動でロック解放することはしませんので、適切な対応を取ってください。</p>
	フェイルオーバータイムアウトが発生した	<p>【クライアント】 70000</p> <p>【内容】 クライアントフェイルオーバータイムアウト</p> <p>【書式】 タイムアウト経過期間、パーティション番号、接続アドレス、失敗原因</p>	<p>クラスタフェイルオーバーに時間を要するような大規模データ、特に、BLOB等一つのロウが非常に大きい場合は、フェイルオーバー(実行時の同期処理)に時間を要する場合がありますので、フェイルオーバータイムアウトを長く設定しておいてください。また、障害検知はハートビート間隔で行っていますので、ハートビート間隔が大きい場合は、検知までの時間を要するため、その場合もフェイルオーバー時間が長くなります。</p>

レプリケーション失敗に起因するエラー	レプリケーション処理が正常に完了しない状態で、フェイルオーバー処理が開始された	<p>【サーバ】 50002</p> <p>【内容】 更新操作連続性チェックエラー</p> <p>【書式】 パーティション番号、接続アドレス、失敗原因</p>	レプリケーション処理におけるメッセージ送受信と、ノード障害のタイミングにより、バックアップへのデータが欠損することで本症状が発生する場合があります。特に非同期レプリケーションの場合に発生する確率が高いため、可用性を重視するならば、性能とのトレードオフを考慮した上で準同期レプリケーションのクラスタでの運用を推奨します。
フェイルオーバー先のデータサービスが停止状態に起因するエラー	フェイルオーバー開始時点で有効であったクラスタ状態がフェイルオーバー中にリセットされ、SUB_CLUSTER状態になっている	<p>【サーバ】 10010</p> <p>【内容】 クラスタ非構成時のアクセス</p> <p>【書式】 パーティション番号、接続アドレス、失敗原因</p>	クラスタ障害が発生した原因については、「クラスタ障害に関するトラブル」を参照してください。もし、半数以上のノードがダウンしたことによりクラスタ構成がリセットされた場合は、新たにノードを用意して、構成ノード数のノードが確保できる状態までクラスタを戻してください。
	gs_cluster.jsonで設定したレプリカ数以上のノードで同時にノード障害が発生した	<p>【サーバ】 10007</p> <p>【内容】 (データロス検知によるデータサービス停止)</p> <p>【書式】 (※マスタノードのみ) パーティション番号、該当パーティションでダウンノードも含めた最新データのLSN(Log Sequence Number)、現在のクラスタにおける最大のLSN、最新データを保有していると推定されるノードアドレス(但し、確実性は保証しない)</p> <p>【コマンド確認】 gs_partition --lossで上記エラー内容と同じ情報を取得できる</p>	継続させることによりデータ一貫性が崩れることをクラスタが検知したため、該当パーティションに対するデータサービスを停止させた状態です。可用性と性能とのトレードオフになりますが、可用性を重視するならば、gs_cluster.jsonのレプリカ数を想定される同時ダウン数以上に設定してください。
	gs_cluster.jsonで設定したレプリカ数以下のノードでの同時障害であったが、発生時点で、レプリカが一時的に不足状態していたパーティションが存在していた	<p>【サーバ】 : 10007</p> <p>【内容】 (データロス検知によるデータサービス停止)</p> <p>【書式】 (※マスタノードのみ) パーティション番号、該当パーティションでダウンノードも含めた最新データのLSN、現在のクラスタにおける最大のLSN、最新データを保有していると推定されるノードアドレス</p> <p>【コマンド確認】</p> <p>発生後は、gs_partition --lossで確認。レプリカ不足状態で稼働している場合は、gs_stat /cluster/partitionStatusで、REPLICA_LOSSと表示されているので、現在の可用性を確認する。個別のパーティション状態を知りたい場合は、gs_partitionを用いて、パーティションごとのレプリカ数を確認する</p>	GridDBではフェイルオーバー時のダウンタイムを小さくするために、あるパーティションに対する同期実行で、ある程度時間を要する(適用するログが大きいかどうかで判定)とクラスタが判定した場合、先に短時間で同期可能なノード集合だけで同期を実行して、データサービスを部分的に開始します。この場合、バックグラウンドの非同期実行で不足分のレプリカを作成しますが、復元されるまでは可用性が低下した状態になります。よって、仮にgs_cluster.jsonでレプリカ数を十分に設定していたとしても、バックグラウンドのレプリカ復元が間に合わなければこの状態になることに注意してください。
	ハートビートが安定しないなど、クラスタが定常的に構成異常を検知して、フェイルオーバーが繰り返される	<p>【サーバ】 : 50003</p> <p>【内容】 (クラスタ非構成時のアクセス)</p> <p>【書式】 パーティション番号、接続アドレス、失敗原因</p> <p>【サーバ】 : 50004</p> <p>【内容】 (クラスタ構成中のアクセス)</p> <p>【書式】 パーティション番号、接続アドレス、失敗原因</p>	クラスタ障害が定常的に発生しており不安定となっています。「クラスタ障害に関するトラブル」を参照してください。

	ネットワーク分断が発生後、過半数以上のノードが確保できるクラスタが存在したため、継続稼働したが、分散先に最新データがある場合	【サーバ】：10007 【内容】(データロス検知によるデータサービス停止) 【書式】(※マスタノードのみ) パーティション番号、該当パーティションでダウンノードも含めた最新データのLSN、現在のクラスタにおける最大のLSN、最新データを保有していると推定されるノードアドレス	ネットワーク分断は、分断されたノードがダウンとしたのと同じ症状になります。分断先に最新データがある場合は、該当パーティションのデータサービスは一時的に停止されますが、分断されたネットワークが元に戻った時点で、自動的に停止させていたデータサービスが再開されます。
--	--	---	--

### クラスタ障害に関するトラブル

原因		症状	対策
分類1	分類2		
クラスタ構成変更は無いが、レプリカ不足状態での部分運転や、一部のパーティションのデータサービス停止となる障害が発生する	現オーナ、バックアップ間の大きなレプリケーション遅延をクラスタが異常状態と判定してバックアップから除外した	【サーバ】：10011 【内容】(バックアップ異常検知) 【書式】(※マスタノードのみ) パーティション番号、オーナアドレス、オーナLSN, バックアップアドレス、バックアップLSN	レプリケーション遅延は特に非同期レプリケーションモードの際に発生しやすい傾向があります。発生が頻発するようでしたら、準同期レプリケーションモードに変更する (gs_cluster.jsonファイルの /transaction/replicationMode=1にする)か、 gs_cluster.jsonファイルに/cluster/ownerBackupLsnGapを追記して値をデフォルト値(50000)より大きくしてください(*1)。 例： "cluster":{ "ownerBackupLsnGap":"100000", : }
	同期タイムアウト内に同期が完了しなかったため、(一部の)データサービスが、指定レプリカ数以下で部分運転が開始された	【サーバ】：10016 【内容】(同期タイムアウト検知) 【書式】(※オーナノードのみ) パーティション番号、部分運転開始時のオーナアドレス	アプリダウンタイムとのトレードオフになりますが、可用性を重視する場合は同期タイムアウト時間(gs_cluster.jsonファイルの/sync/timeoutInterval)を大きくしてください。この値は、ダウンタイムの最大値の目安になります。
	あるパーティションの最新データを保持するノードがダウンして、継続させた場合にデータ不整合が発生するために、該当パーティションのデータサービスを停止させた	【サーバ】：10007 【内容】(データロス検知によるデータサービス停止) 【書式】(※マスタノードのみ) パーティション番号、該当パーティションでダウンノードも含めた最新データのLSN、現在のクラスタにおける最大のLSN、最新データを保有していると推定されるノードアドレス	「クライアントフェイルオーバーに関するトラブル」も参照してください。 また、レプリカ数を増やす、準同期レプリケーションに変更する(gs_cluster.jsonファイルの /transaction/replicationMode=1にする)ことで、発生確率を小さくすることができます。
	マスタノードが、フォロワノードダウン、或いはノード状態がABNORMALへ遷移、或いはgs_leaveclusterが実行されたことを検知した	【サーバ】：10010 【内容】(フェイルオーバー開始) 【書式】(※マスタノードのみ) 異常を検知したノード一覧、フェイルオーバー番号	それぞれの異常内容については、それぞれのノードのイベントログ、エラーコードを確認して下さい。

クラスタ構成変更が必要となる障害が発生する	ネットワーク障害もしくは大きな遅延が発生したことによりハートビート異常を検知した	<p>【サーバ】：10008, 10009</p> <p>【内容】(ハートビートタイムアウト)</p> <p>【書式】(※マスタノードのみ) 異常を検知したノード、ハートビート限界時刻、最終ハートビート到着時刻</p> <p>【外部ツール】 ネットワーク帯域内か確認</p>	<p>断続的な切断でなく、定常的に発生するようでしたら、ハートビート間隔を長くして下さい。但し、ハートビート間隔を長くしすぎると、障害異常検知、及び復旧時間が遅くなるので、可用性とのトレードオフを考慮して設定してください。ネットワーク帯域による遅延であるならば、gs_cluster.jsonの各serviceAddress/servicePortを、それぞれ他ネットワークと別に設定することで発生確率を低下させることもできます。</p>
	ネットワーク以外のリソース高負荷によるハートビート異常が検知した	<p>【サーバ】：10008, 10009</p> <p>【内容】(ハートビートタイムアウト)</p> <p>【書式】(※マスタノードのみ) 異常を検知したノード、ハートビート限界時刻、最終ハートビート到着時刻</p> <p>【外部ツール】 リソース検視ツール</p>	<p>断続的な切断でなく、定常的に発生するようでしたら、ハートビート間隔を長くして下さい。ネットワークと異なり、メモリ不足によるスワップ、ディスクI/O待ち、高負荷アプリケーション実行によりサーバがビジー状態、同一マシン内での別アプリ起動によるもの、等、様々な要因があります。gs_statだけでなく、該当マシン全体でのリソース状態も確認して下さい。</p>
クラスタ構成を維持できない障害が発生する	半数以上のノードの異常検知、或いは離脱によりクラスタがリセットされた	<p>【サーバ】：10014</p> <p>【内容】(過半数ノード確保できないためのクラスタ解散)</p> <p>【書式】(※マスタノードのみ) クラスタ維持に必要なノード数、有効ノード数</p> <p>直前にはハートビート異常検知も記録される</p> <p>【サーバ】：10008, 10009</p> <p>【内容】(ハートビートタイムアウト)</p> <p>【書式】(※マスタノードのみ) 異常を検知したノード、ハートビート限界時刻、最終ハートビート到着時刻</p>	<p>障害が発生してダウンしまったノードを復旧させてください。もし、復旧不可能な場合は、別途新たなノードを起動して、構成ノード数のノードとなるようにクラスタを復元させてください。但し、ダウンしているノード内に、最新データが保持されている可能性もあるので、追加の際は"gs_partition --loss"等で確認して下さい。</p>
	稼働中クラスタにおいてネットワーク分断が発生し、その後、クラスタ再構成を行おうとするが、どの部分クラスタも過半数ノードを確保できなかった	<p>【サーバ】：10014</p> <p>【内容】(過半数ノード確保できないためのクラスタ解散)</p> <p>【書式】(※マスタノードのみ) クラスタ維持に必要なノード数、アクティブノード数</p> <p>直前にはハートビート異常検知も記録される</p> <p>【サーバ】：10008, 10009</p> <p>【内容】(ハートビートタイムアウト)</p> <p>【書式】(※マスタノードのみ) 異常を検知したノード、ハートビート限界時刻、最終ハートビート到着時刻</p>	<p>ネットワーク分断の場合、分断が復旧された時点で自動的にクラスタが再開されますが、もし、分断復帰の見込みが無い場合、手で構成台数を減らしてクラスタを再構成する必要があります。但し、分散先ノード内に、最新データが保持されている可能性もあるので、追加の際は"gs_partition --loss"等で確認して下さい。</p>

リバランス(レプリカ不足ノードに対するレプリカ作成およびノード間の均等配分に失敗する	チェックポイント競合などが発生し、実行中のリバランス処理を継続できなかった	【サーバ】 : 10012 【内容】 (リバランス失敗) 【書式】 失敗が発生したパーティション番号、パーティショングループ番号、チェックポイント番号、失敗原因	リバランス実行中に、チェックポイントが実行された場合、データファイルが更新され、更にログファイル削除される場合があります。この場合、実行中のリバランス処理は途中で打ち切られます。但し、もし、打ち切られたとしても、定期的にチェックしてリトライされます。但し、時間ロスが発生するため、チェックポイント時間を大きくすることと、ログ保持個数を予め大きくしておくことで発生確率を下げるができます。
	リバランスタイムアウト期間内にリバランスが完了できなかった	【サーバ】 : 10014 【内容】 (リバランスタイムアウト) 【書式】 (※マスタノードのみ) タイムアウト検知したパーティション番号、タイムアウト時間	リバランスタイムアウト時間を長く設定してください。
クラスタノードで継続不可能な障害が発生した	プラットフォームエラーによるノード停止(ノード状態がABNORMAL)	【サーバ】 : 各プラットフォームエラー番号 【内容】 プラットフォームエラー	ノード状態がABNORMALとなっている場合、プラットフォームエラーにより、継続不可能な障害が発生している場合があります。詳しくは、下記項目を参照ください。必要な情報を採取し、原因を取り除いた上で、ノードの強制停止(gs_stopnode --forceなど)、ノードの再起動(gs_startnodeなど)、クラスタ参加(gs_joinclusterなど)を実行してください。不要なファイルを削除してディスク領域を確保するか、ディスク増設を行うか、新たなディスク領域が確保できる新規ノードを用意して下さい。
	ディスクフルによるノード停止	【サーバ】 : 各プラットフォームエラー番号 【内容】 プラットフォームエラー(ディスクフル) 【外部ツール】 df, duで確認	物理的なディスク障害や、リソース枯渇によるファイル書き込み失敗などのケースと、誤って必要ファイルを手動で削除してしまった等、幾つかのケースが考えられます。後者は、「リカバリ処理に関するトラブル」を参照してください。
	ディスクI/O障害によるノード停止	【サーバ】 : 各プラットフォームエラー番号 【内容】 プラットフォームエラー(ディスクI/O)	物理メモリに対して、メモリ上限(storeMemoryLimit)を大きくしすぎていないかをまず確認して下さい(gs_stat)。また、processMemoryが肥大化している場合は、要求処理が滞留している可能性もあるので、--memoryDetailオプション付きでgs_statを実行することで出力される、通信メッセージの総確保量(/performance/memoryDetail/work.transactionMessageTotal)を確認して下さい。storeMemoryLimitおよびprocessMemoryは、"gs_stat"で確認してください。storeMemoryLimitは、gs_node.jsonを編集および"gs_paramconf"より変更できます。
	メモリ障害によるノード停止	【サーバ】 : 各プラットフォームエラー番号 【内容】 プラットフォームエラー(メモリアロケート) 【外部ツール】 vmstat, top 【コマンド確認】 gs_stat : /performance/processMemory	物理メモリに対して、メモリ上限(storeMemoryLimit)を大きくしすぎていないかをまず確認して下さい(gs_stat)。また、processMemoryが肥大化している場合は、要求処理が滞留している可能性もあるので、--memoryDetailオプション付きでgs_statを実行することで出力される、通信メッセージの総確保量(/performance/memoryDetail/work.transactionMessageTotal)を確認して下さい。storeMemoryLimitおよびprocessMemoryは、"gs_stat"で確認してください。storeMemoryLimitは、gs_node.jsonを編集および"gs_paramconf"より変更できます。

(\*1)

gs\_cluster.jsonファイルのcluster/ownerBackupLsnGap : パーティションのバックアップの異常判定、およびオーナ (パーティションのマスタ) への昇格判定のためのLSN閾値

将来、本パラメータは名前の変更や削除される可能性があります。

## リカバリ処理に関するトラブル



原因		症状	対策
分類1	分類2		
操作エラーによる失敗	定義ファイルの設定値とデータベースファイルの内容が一致しない	【サーバ】：7036, 160012, 160013 【内容】(パーティショングループ数不一致) 【書式】定義ファイルのパーティショングループ数、データベースファイルのパーティショングループ数	gs_node.jsonの/dataStore/concurrencyの値を確認します。 バックアップからのリカバリの場合、バックアップ時点の設定値(gs_backup_info.jsonの/configInfo/groupNum)と一致しているか確認してください。
		【サーバ】：68024, 160018 【内容】(パーティション数不一致) 【書式】定義ファイルのパーティション数、データベースファイルのパーティション数	gs_cluster.jsonの/dataStore/partitionNumの値を確認します。 バックアップからのリカバリの場合、バックアップ時点の設定値(gs_backup_info.jsonの/configInfo/partitionNum)と一致しているか確認してください。
		【サーバ】：68026, 68064 【内容】(ブロックサイズ不一致) 【書式】定義ファイルのブロックサイズ、データベースファイルのブロックサイズ	gs_cluster.jsonの/dataStore/storeBlockSizeの値を確認します。 イベントログのactualFileSize(データベースファイルのブロックサイズ)の値と一致しているか確認してください。
	リカバリに必要なファイルが存在しない	【サーバ】：20004~20008 【内容】(必要ファイルチェックエラー) 【書式】不足が確定したファイル内容	バックアップから起動する場合、すべてのファイルがコピーされているか確認します("gs_restore --test"など)。 バックアップに関する詳細は、『GridDB バックアップガイド』を参照ください。
内部異常による失敗	リカバリ処理中に継続できないエラーが発生した	【サーバ】：20009 【内容】(内部エラー) 【書式】異常を検知した時点のトレースによる	サポートサービスにお問合せください。バックアップが存在するならばそのデータで復元することも可能です。

### コンテナ操作に関するトラブル

原因		症状	対策
分類1	分類2		
全ての操作に共通する対応	タイムアウトが発生した場合	クライアントフェイルオーバータイムアウト	「クライアントフェイルオーバーに関するトラブル」を参照ください。
	接続障害が発生した場合	クライアントフェイルオーバータイムアウト	「クライアントフェイルオーバーに関するトラブル」を参照ください。
	オブジェクトのクローズ後の呼び出しに起因する失敗	【サーバ】： 140036/145036/ 140038/145038/ 140040/145040 【内容】(クローズに関するエラー)	該当操作は、(コンテナ/リソースの)クローズ処理を呼び出す前に実施して下さい。エラーメッセージを確認し対応ください。

全ての操作に共通する対応	登録済みコンテナに不整合が発生している場合	<p>【サーバ】：60151</p> <p>【内容】(コンテナに不整合が発生)</p> <p>【書式】パーティションID、コンテナ名</p>	<p>指定した操作ができません。サポートサービスにお問い合わせください。</p> <p>特定のコンテナ内のデータで不整合な状態が検出されました。サービスは継続されますが、当該コンテナに関しては以降の更新操作はできません。検索結果も不正な結果になる可能性があります。</p>
		<p>【サーバ】：10017</p> <p>【内容】(コンテナのスキーマバージョンが不一致)</p> <p>【書式】パーティションID、コンテナ名、要求スキーマバージョンID、現スキーマバージョンID</p>	<p>コンテナオブジェクトを再取得してください。</p>
	メモリ使用制限サイズをオーバーした場合	<p>【サーバ】：130033</p> <p>【内容】(メモリ上限オーバー)</p> <p>【書式】メモリ上限サイズ</p>	<p>使用可能メモリサイズの上限を上げる場合は、gs_node.jsonファイルに</p> <p>/dataStore/resultSetMemoryLimitまたは</p> <p>/transaction/totalMemoryLimitを追記し、値をデフォルト値より大きくしてください(*1)。</p> <p>メモリサイズオーバーが発生しているのならば、ヒット件数を抑えることでメモリの使用量を減らすようにクエリを調整することも可能です。</p>
<p>コンテナの登録に失敗する</p> <p>GridStore::putCollection()</p> <p>GridStore::putContainer()</p> <p>GridStore::putTimeSeries()</p>	コンテナの制約違反が発生した場合	<p>【エラーコード】：60015</p> <p>【内容】(コンテナ制約違反)</p> <p>【書式】以下のエラー内容</p> <p>コンテナ名サイズ制限オーバー、</p> <p>カラム数制限オーバー、</p> <p>カラム名重複エラー、</p> <p>ロウキー指定エラー</p> <p>ロウキー型サポートエラー、</p> <p>ロウキー値制約エラー、</p> <p>配列型サポートエラー、</p> <p>期限解放分割値制限オーバー、</p> <p>期限解放値範囲外エラー、</p> <p>間引き圧縮における設定カラム数制限オーバー、</p> <p>アフィニティサイズ制限オーバー、</p> <p>etc</p>	<p>エラーメッセージを確認し、コンテナ情報を修正ください。</p> <p>コンテナの設定については、『運用管理ガイド』の「クラスタ運用管理コマンド・インタプリタ(gs_sh)」の「コンテナ管理」を参照ください。</p>

	登録済みコンテナとのスキーマの不一致が発生した場合	<p>【エラーコード】：60149/60016</p> <p>【内容】(既存コンテナとのスキーマ不一致)</p> <p>【書式】以下のエラー内容</p> <p>ロウキー定義の不一致エラー、          同一カラム名の型不一致エラー、          全カラム名不一致エラー、          アフィニティ値不一致エラー、          期限解放設定不一致エラー、          圧縮設定不一致エラー、          etc</p>	エラーメッセージを確認し、不一致となっているコンテナ情報を修正ください。コンテナ情報は、統合運用管理GUI(gs_admin) または"gs_sh"の"showcontainer"コマンドを利用すると確認できます。
<p>コンテナのスキーマ変更に失敗する</p> <p>GridStore::putCollection()          GridStore::putContainer()          GridStore::putTimeSeries()          )</p>	コンテナの制約違反が発生した場合	<p>【エラーコード】：60015</p> <p>【内容】(コンテナ制約違反)</p> <p>【書式】以下のエラー内容</p> <p>コンテナ名サイズ制限オーバー、          カラム数制限オーバー、          カラム名重複エラー、          ロウキー指定エラー          ロウキー型サポートエラー、          ロウキー値制約エラー、          配列型サポートエラー、          期限解放分割値制限オーバー、          期限解放値範囲外エラー、          間引き圧縮における設定カラム数制限オーバー、          アフィニティサイズ制限オーバー、          etc</p>	エラーメッセージを確認し、コンテナ情報を修正ください。コンテナの設定については、『運用管理ガイド』の「クラスタ運用管理コマンド・インタプリタ(gs_sh)」の「コンテナ管理」を参照ください。
	登録済みコンテナと変更するスキーマの制約違反が発生した場合	<p>【サーバ】：60149</p> <p>【内容】(コンテナ制約違反)</p> <p>【書式】以下のエラー内容</p> <p>ロウキー定義の不一致エラー、          同一カラム名の型不一致エラー、          全カラム名不一致エラー、          Affinity値不一致エラー、          期限解放設定不一致エラー、          圧縮設定不一致エラー、          etc</p>	エラーメッセージを確認し、不一致となっているコンテナ情報を修正ください。コンテナ情報は、統合運用管理GUI(gs_admin) または"gs_sh"の"showcontainer"コマンドを利用すると確認できます。

	スキーマ変更許可パラメータが設定されていない場合	【サーバ】：60016 【内容】(スキーマ変更許可がない)	既存のコンテナのカラムレイアウトの変更が許可されていません。カラムレイアウトの変更を許可するには、"modifiable"パラメータにtrueを設定してください。
コンテナの取得に失敗する GridStore::getCollection() GridStore::getContainer() GridStore::getTimeSeries() )	同名の異なるコンテナタイプが存在する場合	【サーバ】：60026 【内容】(コンテナタイプが不一致)	指定したコンテナのコンテナタイプに誤りがないかご確認ください。
	指定の型が適切でない場合	【サーバ】：140023/145023 【内容】指定の型と既存のカラムレイアウトが一致しないことに起因する失敗 指定の型がロウオブジェクトの型として適切でない場合	指定したカラムレイアウトに誤りがないかご確認ください。カラム情報は、統合運用管理GUI(gs_admin)または"gs_sh"の"showcontainer"コマンドを利用すると確認できます。
コンテナの削除に失敗する GridStore::dropCollection() GridStore::dropContainer() )	同名の異なるコンテナタイプが存在する場合	【サーバ】：60026 【内容】(コンテナタイプが不一致)	指定したコンテナのコンテナタイプに誤りがないかご確認ください。
索引の登録に失敗する Container::createIndex()	対応する名前のカラムが存在しない場合	【サーバ】：140008/145008 【内容】(不明なカラム名)	指定したカラム名に誤りがないかご確認ください。カラム情報は、統合運用管理GUI(gs_admin)または"gs_sh"の"showcontainer"コマンドを利用すると確認できます。
	索引設定がサポートされないカラムタイプを指定した場合	【サーバ】：1007 【内容】(サポート外)	指定したカラム(カラムタイプ)に誤りがないかご確認ください。索引については、『APIリファレンス』の「API一覧(Java)」の「createIndex」を参照ください。
索引の削除に失敗する Container::dropIndex()	対応する名前のカラムが存在しない場合	【サーバ】：140008/145008 (不明なカラム名)	カラム名に誤りがないかご確認ください。カラム情報は、統合運用管理GUI(gs_admin)または"gs_sh"の"showcontainer"コマンドを利用すると確認できます。
トリガーの登録に失敗する Container::createTrigger()	トリガーの制約違反が発生した場合	【サーバ】：140001/145001/170003/10040 【内容】(トリガー制約違反) 【書式】以下のエラー内容 トリガ名がnull、または空の場合 監視対象更新操作の指定がない場合 通知先URIが規定の構文に合致しない場合 トリガ種別でJMSが指定され、かつJMSデスティネーション種別がnull、または空、または指定の書式に合致しない場合 トリガ種別でJMSが指定され、かつJMSデスティネーション名がnull、または空の場合 この処理のタイムアウト、このコンテナの削除、接続障害が発生した場合、またはクローズ後に呼び出された場合	エラーメッセージを確認しトリガ情報を修正ください。トリガ機能については、『APIリファレンス』の「トリガ機能」を参照ください。

	トリガーの上限数を越えた場合	<p>【サーバ】：1008</p> <p>【内容】（トリガー上限違反）</p> <p>【書式】以下のエラー内容</p> <p>トリガー名の制限値オーバー</p> <p>トリガー名の使用禁止文字</p> <p>URIの長さの制限値オーバー</p> <p>トリガー登録数の制限値オーバー</p>	エラーメッセージを確認しトリガ情報を修正ください。トリガ機能については、『APIリファレンス』の「API一覧 (Java)」の「createTrigger」を参照ください。
トリガーの削除に失敗する Container::dropTrigger()	共通原因以外無		
ロウの登録・更新に失敗する Container::put() TimeSeries::append()	ロウキーに対応するカラムが存在しないにもかかわらず、キーが指定された場合	<p>【サーバ】：140024/145024</p> <p>【内容】（キーが見つからない）</p> <p>【サーバ】：60079</p>	指定されたロウキーが存在しません。設定をご確認ください。
	カラム値に制約違反がある場合	<p>【内容】（カラム制約違反）</p> <p>【書式】以下のエラー内容</p> <p>配列長制限オーバー、</p> <p>文字列、空間、BLOB等の可変長型の制限オーバー</p> <p>時刻型のサポート範囲外値、</p> <p>etc</p>	エラーメッセージを確認しカラム値を修正ください。カラム値については、『APIリファレンス』の「概要」の「型」を参照ください。
ロウの一括登録・更新に失敗する Container::multiput()	指定したコンテナが存在しない場合	<p>【サーバ】：10016</p> <p>【内容】（コンテナが見つからない）</p>	指定したコンテナが存在しません。コンテナ名をご確認ください。
	指定の型が適切でない場合	<p>【サーバ】：60015</p> <p>【内容】指定のロウの型と既存のカラムレイアウトが一致しないことに起因する失敗</p> <p>指定のロウの型がロウオブジェクトの型として適切でない場合</p>	カラム情報を確認し、適切なカラムレイアウト(カラムタイプ)に修正ください。
	カラム値に制約違反がある場合	<p>【サーバ】：60079</p> <p>【内容】（カラム制約違反）</p> <p>【書式】以下のエラー内容</p> <p>配列長制限オーバー、</p> <p>文字列、空間、BLOB等の可変長型の制限オーバー</p> <p>時刻型のサポート範囲外値、</p> <p>etc</p>	エラーメッセージを確認しカラム値を修正ください。カラム値については、『APIリファレンス』の「概要」の「型」を参照ください。
	対象位置のロウが存在しない場合	<p>【サーバ】：140037/145037</p> <p>【内容】（カーソルの指定先が存在しない）</p>	RowSetのカーソル位置を確認
	ロックを有効にせずに取得したRowSetに対して呼び出された場合	<p>【サーバ】：140039/145039</p> <p>【内容】（ロックされていない）</p>	ロックが必要

RowSet経由のロウの更新に失敗する RowSet::update()	カラム値に制約違反がある場合	【サーバ】：60079 【内容】（カラム制約違反） 【書式】以下のエラー内容 配列長制限オーバー、 文字列、空間、BLOB等の可変長型の制限オーバー 時刻型のサポート範囲外値、 etc	エラーメッセージを確認しカラム値を修正ください。カラム値については、『APIリファレンス』の「概要」の「型」を参照ください。
ロウの削除に失敗する Container::remove()	ロウキーに対応するカラムが存在しない場合	【サーバ】：140024/145024 【内容】（キーが見つからない）	指定されたロウキーが存在しません。設定をご確認ください。
	圧縮が指定されたコンテナに対してロウの削除を指定した場合	【サーバ】：60086 【内容】（圧縮コンテナに対する操作不正）	圧縮が指定された時系列コンテナではロウは削除できません。
RowSet経由のロウの削除に失敗する RowSet::remove()	対象位置のロウが存在しない場合	【サーバ】：140037/145037 【内容】（カーソルの指定先が存在しない）	RowSetのカーソル位置を確認
	ロックを有効にせずに取得したRowSetに対して呼び出された場合	【サーバ】：140039/145039 【内容】（ロックされていない）	ロックが必要
Commitに失敗する Container::commit()	自動コミットモードであるにもかかわらず呼び出した場合	【サーバ】：140035/145035 【内容】（不正なコミットモード）	gs_node.jsonのlogWriteMode(ログ書き込みモード)の値が"DELAYED_SYNC"の場合、指定間隔でログの書き込みを行います。明示的にコミットを実行する場合は、logWriteModeを"SYNC"に設定してください。
Abortに失敗する Container::abort()	自動コミットモードであるにもかかわらず呼び出した場合	【サーバ】：140035/145035 【内容】（不正なコミットモード）	gs_node.jsonのlogWriteMode(ログ書き込みモード)の値が"DELAYED_SYNC"の場合、指定間隔でログの書き込みを行います。明示的にアボートを実行する場合は、logWriteModeを"SYNC"に設定してください。
ロウの登録・更新が反映されない Container::put() TimeSeries::append()	圧縮が指定されたコンテナに対してロウの更新、または最新登録時刻よりも過去時刻を登録した場合	【トレース】 「insert(old time)/update not support on Compression Mode」	圧縮が指定された時系列コンテナにおける仕様です。 圧縮オプションが設定された状態の時系列コンテナに対しては、最も新しい時刻を持つ既存ロウより新しい時刻のロウのみを新規作成できます。最も新しい時刻を持つ既存ロウの時刻が指定の時刻と一致する場合、何も変更は行わず既存ロウの内容を保持します。
ロウのget検索に失敗する Query::get()	ロウキーに対応するカラムが存在しない場合	【サーバ】：140024/145024 【内容】（キーが見つからない）	指定されたロウキーが存在しません。設定をご確認ください。
	キー値に制約違反がある場合	【サーバ】：70002 【内容】（キー制約違反） 【書式】以下のエラー内容 文字列型の制限オーバー 時刻型のサポート範囲外値	キー値が型の制約から外れないよう、エラーメッセージを確認し、設定を変更ください。設定してください。値については、『APIリファレンス』の「概要」の「型」を参照ください。
	自動コミットモードであるにもかかわらず、更新用ロックを要求しようとした場合	【サーバ】：140035/145035 【内容】（不正なコミットモード）	gs_node.jsonのlogWriteMode(ログ書き込みモード)の値が"DELAYED_SYNC"の場合、指定間隔でログの書き込みを行います。更新用ロックを要求する場合は、logWriteModeを"SYNC"に設定してください。

複数コンテナ一括ロウ取得検索に失敗する Query::multiGet()	指定したコンテナが存在しない場合	【サーバ】：10016 【内容】（コンテナが見つからない）	指定されたコンテナが存在しません。設定をご確認ください。
	キー値に制約違反がある場合	【サーバ】：70002 【内容】（キー制約違反） 【書式】以下のエラー内容 文字列型の制限オーバー 時刻型のサポート範囲外値	キー値が型の制約から外れないよう、エラーメッセージを確認し、設定を変更ください。設定してください。値については、『APIリファレンス』の「概要」の「型」を参照ください。
ロウの線形補完検索に失敗する TimeSeries::interpolate()	対応する名前のカラムが存在しない場合。 また、サポートされていない型のカラムが指定された場合	【サーバ】：140008/145008 【内容】（不明なカラム名）	ロウキーが設定されていないコンテナに対しては実施できません。
	キー値に制約違反がある場合	【サーバ】：70002 【内容】（キー制約違反） 【書式】以下のエラー内容 文字列型の制限オーバー 時刻型のサポート範囲外値	キー値が型の制約から外れないよう、エラーメッセージを確認し、設定を変更ください。設定してください。値については、『APIリファレンス』の「概要」の「型」を参照ください。
ロウのサンプリング検索に失敗する TimeSeries::query()	サンプリングの期間の単位に制約違反がある場合	【サーバ】：60151 【内容】（期間の単位がサポート範囲外）	YEAR,MONTH,MILLISECOND以外の単位の設定が必要です。
	サンプリング間隔に制約違反がある場合	【サーバ】：70004 【内容】（サンプリング間隔制約違反） 【書式】以下のエラー内容 intervalの値が0または負の値	正の値を設定してください(0は含みません)。
	指定したカラム名が存在しない場合	【サーバ】：140008/145008 【内容】（不明なカラム名）	指定されたカラムが存在しません。設定をご確認ください。
	キー値に制約違反がある場合	【サーバ】：70002 【内容】（キー制約違反） 【書式】以下のエラー内容 文字列型の制限オーバー 時刻型のサポート範囲外値	キー値が型の制約から外れないよう、エラーメッセージを確認し、設定を変更ください。設定してください。値については、『APIリファレンス』の「概要」の「型」を参照ください。
ロウの集約関数に失敗する TimeSeries::aggregate()	指定の演算方法で許可されていない型のカラムを指定した場合	【サーバ】：60100 【内容】（許可されていないカラム操作）	演算式を見直し、適用可能なカラムに設定を変更してください。各演算の条件は、『APIリファレンス』の「条件式構文・演算機能」を参照ください。
	指定したカラム名が存在しない場合	【サーバ】：140008/145008 【内容】（不明なカラム名）	指定されたカラムが存在しません。設定をご確認ください。

(\*1)

gs\_node.jsonファイルの/dataStore/resultSetMemoryLimit：検索結果(ResultSet)のメモリ上限サイズ。デフォルト値は10240MB。単位省略時はMB

gs\_node.jsonファイルの/transaction/totalMemoryLimit：トランザクション処理メモリのプールが保持する空きメモリの上限サイズ。デフォルト値は1024MB。単位省略時はMB

将来、本パラメータは名前の変更や削除される可能性があります。

## TQLに関するトラブル

原因		症状	対策
分類1	分類2		

TQLの記述で共通した記述誤りで失敗する	指定したカラムが存在しない場合	【エラーコード】： 150012 【内容】(カラムが見つからない) 【書式】以下のエラー内容 カラムが見つからない カラムIDが見つからない	エラーメッセージを確認しカラム名やカラムIDをご確認ください。
	TQLの解釈に失敗する	【エラーコード】： 151001 【内容】(シンタックスエラー)	シンタックスエラーが発生しました。TQL構文をご確認ください。TQL構文については『APIリファレンス』の「TQL構文・演算機能」を参照ください。
		【エラーコード】： 151002 【内容】(トークンが不正)	不正なキーワードが検出されました。TQL構文をご確認ください。TQL構文については『APIリファレンス』の「TQL構文・演算機能」を参照ください。
		【エラーコード】： 151003 【内容】(ダブルコーテーションで囲われていない) 【書式】以下のエラー内容 dequoteできない	ダブルコーテーションが不正です。TQL構文をご確認ください。TQL構文については『APIリファレンス』の「TQL構文・演算機能」を参照ください。
データが存在しない場合	【エラーコード】： 152009 【内容】(配列のインデックスが範囲外) 【書式】以下のエラー内容 配列に指定したインデックスが範囲外	配列のインデックスが不正です。TQL構文をご確認ください。配列(複合型)については『APIリファレンス』の「型」を参照ください。	
FROM節の記述誤りで失敗する	コンテナ名が正しくない場合	【エラーコード】： 150010 【内容】(コンテナ名が不正) 【書式】以下のエラー内容 APIで与えられたコンテナ名がFROM節と等しくない	指定されたコンテナが存在しません。コンテナ名をご確認下さい。
WHERE節の記述誤りで失敗する	TQLの解釈に失敗するの制約違反の場合	【エラーコード】： 150013 【内容】(WHERE条件に*が使用されている) 【書式】以下のエラー内容 WHERE条件に*は使用できない	WHERE条件に*は使用できません。TQL構文をご確認ください。TQL構文については『APIリファレンス』の「TQL構文・演算機能」を参照ください。
	0で除算しようとした場合	【エラーコード】： 150016 【内容】(0で除算している) 【書式】以下のエラー内容 0で除算している	0で除算はできません。TQL構文をご確認ください。
	データ型の制約違反の場合	【エラーコード】： 150018 【内容】(バイナリ演算が未サポート) 【書式】以下のエラー内容 バイナリ演算が定義されていない	指定されたデータ型同士のバイナリ演算は実行できません。TQL構文を見直してください。
		【エラーコード】： 150019 【内容】(カラム条件検索での制約違反) 【書式】以下のエラー内容 WHERE条件に使用されたカラムがboolean型でない	WHERE条件にカラムが単独で使用されている場合、そのカラムはbooleanである必要があります。TQL構文を見直してください。



		<p>【エラーコード】：152010</p> <p>【内容】(空間検索での制約違反)</p> <p>【書式】以下のエラー内容 空間範囲条件に指定されたカラムが空間型でない</p>	空間範囲検索では空間型のカラムを使って条件を指定します。TQL構文をご確認ください。
ORDER BY節の記述誤りで失敗する	整列方法の表現に起因する失敗	<p>【エラーコード】：151004</p> <p>【内容】(ORDER BY節の整列方法の制約違反)</p> <p>【書式】以下のエラー内容 ORDER BY節にはカラム名のみ許可されている</p>	ORDER BY節に指定した整列方法をご確認ください。ORDER BY節については『APIリファレンス』の「TQL構文・演算機能」の「検索結果の並び替え (ORDER BY)」を参照ください。
	使用条件に違反する失敗	<p>【エラーコード】：151005</p> <p>【内容】(ORDER BY節の使用条件の制約違反)</p> <p>【書式】以下のエラー内容 集約関数とORDER BYが同時に使用されている</p>	集約関数とORDER BY節は同時に使用できません。TQL構文をご確認ください。ORDER BY節については『APIリファレンス』の「TQL構文・演算機能」の「検索結果の並び替え (ORDER BY)」を参照ください。
関数の記述誤りで失敗する	関数名に起因する失敗	<p>【エラーコード】：150014</p> <p>【内容】(関数が見つからない)</p> <p>【書式】以下のエラー内容 そのような関数はない</p>	関数名をご確認ください。TQL構文については『APIリファレンス』の「TQL構文・演算機能」を参照ください。
	関数の使用条件に違反する場合	<p>【エラーコード】：152003/152004</p> <p>【内容】(コレクション/時系列コンテナ専用の関数の不正使用)</p> <p>【書式】以下のエラー内容 コレクション用の選択関数が時系列コンテナに対して使用された</p>	コレクションに対して時系列コンテナ専用の関数が使用されていないかご確認ください。または、時系列コンテナに対してコレクション専用の関数が使用されていないかご確認ください。TQL構文については『APIリファレンス』の「TQL構文・演算機能」を参照ください。
	引数の数に起因する失敗	<p>【エラーコード】：152002/152007/152011</p> <p>【内容】(引数の数が不正)</p> <p>【書式】以下のエラー内容 関数の引数の数が不正 引数が空を期待する関数で引数が空でない</p>	関数の引数をご確認ください。TQL構文については『APIリファレンス』の「TQL構文・演算機能」を参照ください。
	引数のデータ型に起因する失敗	<p>【エラーコード】：152002/152007/152010/152020</p> <p>【内容】(引数の型が不正)</p> <p>【書式】以下のエラー内容 関数の引数の型が不正 補間対象がカラムではない 数値でないカラムに対して補間かけようとしたカラムではない引数が使用された</p>	関数の引数に与えたデータの型をご確認ください。TQL構文については『APIリファレンス』の「TQL構文・演算機能」を参照ください。

	引数値に起因する失敗	<p>【エラーコード】：150005/152010/152012</p> <p>【内容】(引数の値が不正)</p> <p>【書式】以下のエラー内容</p> <p>範囲外の時刻データが使用された</p> <p>座標値が不正</p> <p>エスケープ文字が1文字でない</p> <p>自然数を期待するが0以下の値が使用されている</p> <p>文字列を時刻型に変換できない</p>	関数の引数に与えたデータが正しくありません。TQL構文については『APIリファレンス』の「TQL構文・演算機能」を参照ください。
空間型データの記述誤りで失敗する	引数の数に起因する失敗	<p>【エラーコード】：152007/152010</p> <p>【内容】(引数の数が不正)</p> <p>【書式】以下のエラー内容</p> <p>引数を必要とするが空である</p> <p>1つの点を指定して線分を作成しようとした</p>	空間型データ表現の引数をご確認ください。TQL構文については『APIリファレンス』の「TQL構文・演算機能」を参照ください。
	引数のデータ型に起因する失敗	<p>【エラーコード】：152010</p> <p>【内容】(引数の型が不正)</p> <p>【書式】以下のエラー内容</p> <p>引数の型が不正</p>	空間型データ表現の引数の型をご確認ください。TQL構文については『APIリファレンス』の「TQL構文・演算機能」を参照ください。
TQLの実行に失敗する	メモリが確保できなかった場合	<p>【エラーコード】：1001</p> <p>【内容】(メモリ確保エラー)</p> <p>【書式】以下のエラー内容</p> <p>TQLパーサのメモリを確保できない</p> <p>WKTパーサのメモリを確保できない</p>	使用可能メモリサイズの上限を上げる場合は、gs_node.jsonファイルに /dataStore/resultSetMemoryLimitまたは /transaction/totalMemoryLimitを追記し、値をデフォルト値より大きくしてください(*1)。 メモリサイズオーバーが発生しているのならば、ヒット件数を抑えることでメモリの使用量を減らすようにクエリを調整することも可能です。

(\*1)

gs\_node.jsonファイルの/dataStore/resultSetMemoryLimit：検索結果(ResultSet)のメモリ上限サイズ。デフォルト値は10240MB。単位省略時はMB

gs\_node.jsonファイルの/transaction/totalMemoryLimit：トランザクション処理メモリのプールが保持する空きメモリの上限サイズ。デフォルト値は1024MB。単位省略時はMB

将来、本パラメータは名前の変更や削除される可能性があります。

## 参考 互換性

<互換性> サーバのバージョンに対応するクライアントバージョンとデータベースバージョンは、以下になります。

サーバのバージョン	クライアント		データベース	
	内部バージョン	クライアントのバージョン	内部バージョン	DB作成時のサーバのバージョン
v1.0	1	v1.0/1.1	1	v1.0
v1.1	1	v1.0/1.1	1、2	v1.0/1.1
v1.5	2	v1.5	1、2、3	v1.0/1.1/1.5
v2.0	3	v2.0	4	v2.0
v2.1	4	v2.1	5	v2.1
v2.5	5	v2.5	5	v2.1/v2.5
v2.7	5、6	v2.5/v2.7	6、7	v2.1/v2.5/v2.7

v2.9	8	v2.9	9、10	v2.9
v3.0	9	v3.0	11	v3.0
v3.1	11	v3.1	12、13	v3.1
v3.2	12	v3.2	14、15	v3.2

<メモ>・内部バージョンは、内部処理で使用しているバージョンでエラーメッセージに表示されます。