



Red Hat Data Grid 8.1

Data Grid Guide to Cross-Site Replication

グローバル Data Grid クラスターのデータのバックアップ

Red Hat Data Grid 8.1 Data Grid Guide to Cross-Site Replication

グローバル Data Grid クラスターのデータのバックアップ

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

法律上の通知

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Data_Grid_Guide_to_Cross-Site_Replication.ent file |.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

概要

Data Grid は、グローバルクラスターを形成して、地理的な場所全体にデータを複製できます。本ガイドでは、キャッシュのバックアップの場所を設定し、クロスサイト操作を実行する方法を説明します。また、グローバルクラスターの監視およびトラブルシューティングの方法についても説明します。

目次

第1章 RED HAT DATA GRID	4
1.1. DATA GRID のドキュメント	4
1.2. DATA GRID のダウンロード	4
1.3. 多様性を受け入れるオープンソースの強化	4
第2章 DATA GRID クロスサイトレプリケーション	5
2.1. クロスサイトレプリケーション	5
2.1.1. サイトマスター	5
2.2. キャッシュへのバックアップの追加	6
2.3. バックアップストラテジー	6
2.3.1. 同期バックアップ	6
2.3.2. 非同期バックアップ	6
2.3.3. 同期バックアップと非同期バックアップ	7
2.4. バックアップを自動的にオフラインにする	7
2.5. 状態遷移	8
2.6. サイト全体でのクライアント接続	9
2.6.1. 同時書き込みおよび競合エントリ	10
2.7. 有効期限とクロスサイトのレプリケーション	11
第3章 クロスサイトレプリケーション用の DATA GRID の設定	12
3.1. クロスサイトレプリケーションのためのクラスタートランスポートの設定	12
3.1.1. JGroups RELAY2 スタック	12
3.1.2. カスタム JGroups RELAY2 スタック	13
3.2. キャッシュへのバックアップの場所の追加	14
3.3. 異なる名前でのキャッシュへのバックアップ	15
3.4. クロスサイトビューの確認	15
3.5. クロスサイトレプリケーション用の HOT ROD クライアントの設定	16
第4章 クロスサイトレプリケーション操作の実行	17
4.1. CLI を使用したクロスサイト操作の実行	17
4.1.1. バックアップ場所のオフラインおよびオンライン化	17
4.1.2. バックアップ場所への状態のプッシュ	17
4.2. REST API を使用したクロスサイト操作の実行	18
4.2.1. すべてのバックアップロケーションのステータス取得	18
4.2.2. 特定のバックアップ場所のステータスの取得	18
4.2.3. バックアップ先をオフラインにする	19
4.2.4. バックアップ場所をオンラインにする	19
4.2.5. バックアップ場所への状態のプッシュ	19
4.2.6. 状態転送のキャンセル	19
4.2.7. 状態転送ステータスの取得	19
4.2.8. 状態転送ステータスのクリア	20
4.2.9. オフラインテイク条件の変更	20
4.2.10. 受信サイトからの状態転送のキャンセル	21
4.2.11. バックアップ場所のステータスの取得	21
4.2.12. バックアップ先をオフラインにする	22
4.2.13. バックアップ場所をオンラインにする	22
4.2.14. 状態転送の開始	22
4.2.15. 状態転送のキャンセル	22
4.3. JMX を使用したクロスサイト操作の実行	22
4.3.1. JMX MBean を登録するための Data Grid の設定	22
4.3.2. クロスサイト操作の実行	23

第5章 グローバル DATA GRID クラスターの監視およびトラブルシューティング	24
5.1. DATA GRID 統計の有効化	24
5.2. DATA GRID メトリクスの有効化	25
5.2.1. Data Grid メトリクスの収集	25
5.3. JMX MBEAN を登録するための DATA GRID の設定	26
5.3.1. クロスサイトレプリケーション用の JMX MBean	26
5.4. ログの収集およびクロスサイトレプリケーションのトラブルシューティング	27
5.4.1. クロスサイトログメッセージ	28

第1章 RED HAT DATA GRID

Data Grid は、高性能の分散型インメモリデータストアです。

スキーマレスデータ構造

さまざまなオブジェクトをキーと値のペアとして格納する柔軟性があります。

グリッドベースのデータストレージ

クラスター間でデータを分散および複製するように設計されています。

エラスティックスケールリング

サービスを中断することなく、ノードの数を動的に調整して要件を満たします。

データの相互運用性

さまざまなエンドポイントからグリッド内のデータを保存、取得、およびクエリーします。

1.1. DATA GRID のドキュメント

Data Grid のドキュメントは、Red Hat カスタマーポータルで入手できます。

- [Data Grid 8.1 ドキュメント](#)
- [Data Grid 8.1 コンポーネントの詳細](#)
- [Data Grid 8.1 でサポートされる構成](#)
- [Data Grid 8 機能のサポート](#)
- [Data Grid で非推奨の機能](#)

1.2. DATA GRID のダウンロード

Red Hat カスタマーポータルで [Data Grid Software Downloads](#) にアクセスします。



注記

Data Gridソフトウェアにアクセスしてダウンロードするには、Red Hat アカウントが必要です。

1.3. 多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。これは大規模な取り組みであるため、これらの変更は今後の複数のリリースで段階的に実施されます。詳細は、[Red Hat CTO である Chris Wright のメッセージ](#) をご覧ください。

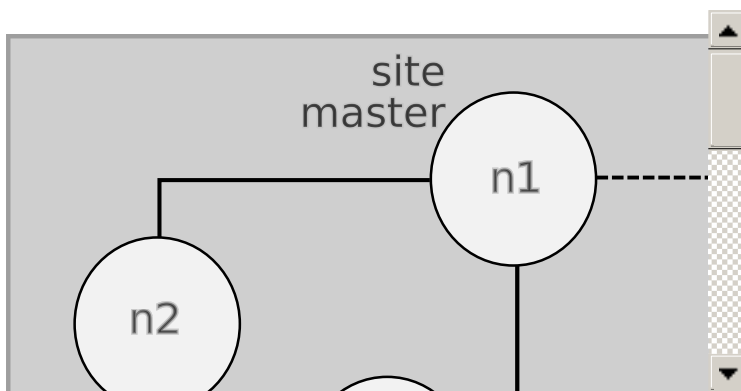
第2章 DATA GRID クロスサイトレプリケーション

クロスサイトレプリケーションを使用すると、ある Data Grid クラスタから別の Data Grid クラスタにデータのバックアップを作成できます。クラスタを設定する前に、Data Grid クロスサイトレプリケーションの仕組みを理解するための概念を説明します。

2.1. クロスサイトレプリケーション

異なる場所で実行されている Data Grid クラスタは、相互に検出および通信できます。

サイトは通常、地理的にさまざまな場所にあるデータセンターです。以下の図のように、クロスサイトレプリケーションはサイト内の Data Grid クラスタをブリッジし、グローバルクラスタを形成します。



LON は、イギリスのロンドンにあるデータセンターです。
NYC は、米国のニューヨーク市にあるデータセンターです。



注記

Data Grid は、2 つ以上のサイトにまたがってグローバルクラスタを形成できます。

たとえば、LON および NYC のバックアップ場所として、サンフランシスコで実行している 3 番目の Data Grid クラスタ SFO を設定します。

2.1.1. サイトマスター

サイトマスターは、バックアップの場所から要求を送受信する Data Grid クラスタのノードです。

ノードがサイトマスターではない場合、バックアップ要求をローカルサイトマスターに転送する必要があります。サイトマスターのみが要求をバックアップできる場所に送信できます。

最適なパフォーマンスを得るには、すべてのノードをサイトマスターとして設定する必要があります。これにより、バックアップ要求をサイトマスターに転送することなく、クラスタの各ノードは直接リモートサイトにバックアップできるため、バックアップ要求の速度が向上します。



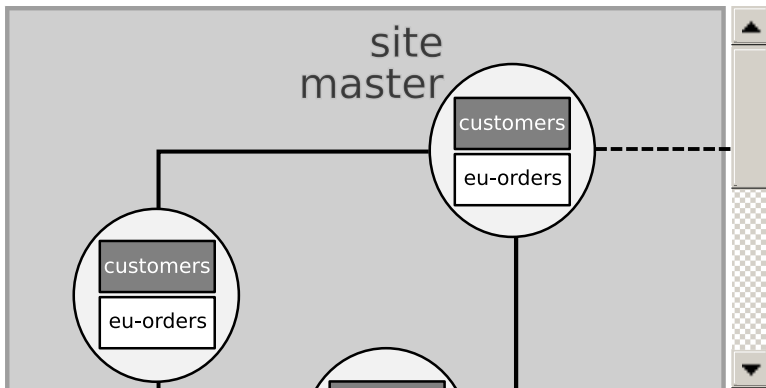
注記

本書では、JGroups RELAY2 プロトコルのデフォルトであるため、本書の図では 1 つのサイトマスターを持つ Data Grid クラスタについて説明します。同様に、クラスタ内の各サイトマスターがリモートクラスタ内の各サイトマスターと通信するので、単一サイトマスターを簡単に説明します。

2.2. キャッシュへのバックアップの追加

キャッシュ定義でバックアップの場所としてリモートサイトに名前を付けます。

たとえば、以下の図では、3つのキャッシュ "customers"、"eu-orders"、および "us-orders" を示しています。



- LON では、"customers" はバックアップの場所として NYC を指定します。
- NYC では、"customers" はバックアップの場所として LON を指定します。
- "eu-orders" および "us-orders" にはバックアップはなく、それぞれのクラスタのローカルになります。

2.3. バックアップストラテジー

Data Grid クラスタは、異なるストラテジーを使用してリモートサイトにデータをバックアップできます。

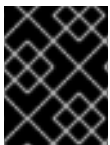
Data Grid は、ローカルキャッシュに書き込む同時に複数のサイト間で複製されます。たとえば、クライアントが LON に "k1" と書き込む場合、Data Grid は同時に "k1" を NYC バックアップします。

2.3.1. 同期バックアップ

Data Grid がデータを複製してバックアップの場所をレプリケートすると、操作が完了するのを待ってからローカルキャッシュに書き込みます。

バックアップ操作が失敗した場合は、Data Grid がローカルキャッシュへの書き込みを処理する方法を制御できます。たとえば、リモートサイトへのバックアップが失敗した場合は、ローカル書き込みを中止するよう Data Grid を設定し、例外をスローすることができます。

同期バックアップは、楽観的なトランザクションに参加するキャッシュを持つ 2 フェーズコミットもサポートします。バックアップの最初のフェーズはロックを取得します。2 番目のフェーズは変更をコミットします。



重要

クロスサイトレプリケーションのある 2 フェーズコミットは、ネットワーク全体で 2 つのラウンドトリップが必要なため、パフォーマンスに大きく影響します。

2.3.2. 非同期バックアップ

Data Grid がデータをバックアップ場所に複製する場合、操作が完了するのを待たずにローカルキャッシュに書き込みます。

非同期バックアップ操作およびローカルキャッシュへの書き込みは、互いに独立しています。バックアップ操作が失敗した場合は、ローカルキャッシュへの書き込み操作は続行され、例外は発生しません。

2.3.3. 同期バックアップと非同期バックアップ

同期バックアップは、サイト全体でのデータの一貫性を最も強力に保証します。**strategy=sync** の場合、**cache.put()** 呼び出しが返されると、ローカルキャッシュとバックアップの場所で値が最新の状態であることがわかります。

この一貫性のために犠牲となるのがパフォーマンスです。同期バックアップは、非同期バックアップと比較すると、レイテンシーが大幅に高くなります。

一方、非同期バックアップは、クライアントリクエストにレイテンシーを追加しないので、パフォーマンスに影響を及ぼすことはありません。ただし、**strategy=async** の場合、**cache.put ()** 呼び出しが返されると、バックアップの場所の値がローカルキャッシュと同じかどうかを確認できません。

2.4. バックアップを自動的にオフラインにする

リモートサイトが利用できなくなった場合に、バックアップの場所が自動的にオフラインになるように設定できます。これにより、Data Grid ノードが、オフラインのバックアップ場所にデータを継続的に複製しようとして、エラーメッセージが表示されたり、リソースが消費されたりすることを防ぎます。

バックアップ操作のタイムアウト

バックアップ設定には、データを複製する操作のタイムアウト値が含まれます。タイムアウトが発生する前に操作が完了しない場合、Infinispan はそれらを失敗として記録します。

```
<backup site="NYC" strategy="ASYNC" timeout="10000"> ❶
...
</backup>
```

❶ NYC にデータを複製する操作は、10 秒後に完了しないと失敗として記録されます。

失敗の数

バックアップの場所がオフラインになる前に発生する可能性のある **連続する** 失敗の数を指定できます。

たとえば、NYC の以下の設定は、オフラインになる前に失敗する操作の数を 5 に設定しています。

```
<backup site="NYC" strategy="ASYNC" timeout="10000">
<take-offline after-failures="5"/> ❶
</backup>
```

❶ クラスターが NYC へのデータの複製を試みると、連続して 5 回の操作に失敗すると、Data Grid は自動的にバックアップをオフラインにします。

待機時間

バックアップ操作の失敗時に、サイトをオフラインにするまで待機する時間を指定することもできます。待機時間がなくなる前にバックアップ要求が成功した場合、Data Grid はサイトをオフラインにしません。

```
<backup site="NYC" strategy="ASYNC" timeout="10000">
  <take-offline after-failures="5"
    min-wait="15000"/> ❶
</backup>
```

- ❶ クラスタで NYC へのデータの複製を試行し、連続して 5 回失敗し、最初の操作に失敗した後 15 秒経過すると、Data Grid は自動的にバックアップをオフラインにします。

ヒント

サイトがオフラインになるまでの待機時間を最小限にしたい場合は、**after-failures** 属性に負またはゼロの値を設定します。

```
<take-offline after-failures="-1" min-wait="10000"/>
```

2.5. 状態遷移

状態遷移は、サイト間でデータを同期する管理操作です。

たとえば、LON がオフラインになると、NYC がクライアント要求の処理を開始します。LON をオンラインに戻すと、LON の Data Grid クラスタには NYC のクラスタと同じデータはありません。

LON と NYC の間でデータの一貫性を保つには、NYC から LON に状態をプッシュできます。

- 状態遷移は双方向です。たとえば、NYC から LON へ、または LON から NYC へ状態のプッシュを実行できます。
- 状態をオフラインサイトにプッシュすると、オンラインに戻ります。
- 状態転送は、発信元サイトと受信サイトの両方のサイトに存在するデータのみを上書きします。Data Grid はデータを削除しません。たとえば、"k2" は LON および NYC に存在します。"k2" は、LON がオフライン時に NYC から削除されます。LON をオンラインに戻すと、"k2" は引き続きその場所に存在します。NYC から LON に状態をプッシュすると、転送は LON の "k2" には影響しません。

ヒント

状態遷移後にキャッシュの内容が同じになるようにするには、状態をプッシュする前に受信サイトのキャッシュからすべてのデータを削除します。**clear ()** メソッドを使用します。

- 状態遷移は、プッシュの開始後に発生するデータへの更新を上書きしません。たとえば、"k1,v1" は LON および NYC に存在します。LON がオフラインになったため、NYC から LON に状態転送をプッシュします。これにより、LON が再びオンラインになります。状態遷移が完了する前に、クライアントは LON に "k1,v2" を配置します。

この場合、プッシュを開始した後に変更が発生したため、NYC からの状態遷移は "k1,v2" を上書きしません。

参照

- [org.infinispan.Cache.clear\(\)](#)
- [Clearing Caches with the CLI](#)

ヒント

コマンドの詳細と例については、CLI から **help clearcache** を実行します。

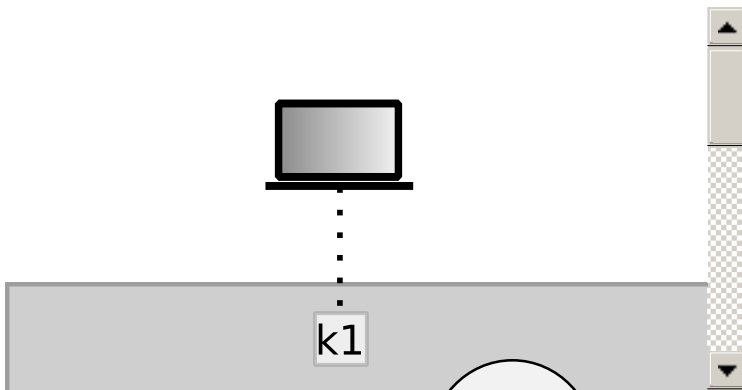
- [Clearing Caches with the REST API](#)

2.6. サイト全体でのクライアント接続

クライアントは、Active/Passive または Active/Active 設定のいずれかで Data Grid クラスターに書き込みできます。

Active/Passive

以下の図は、Data Grid が1つのサイトのみからのクライアント要求を処理する Active/Passive を示しています。



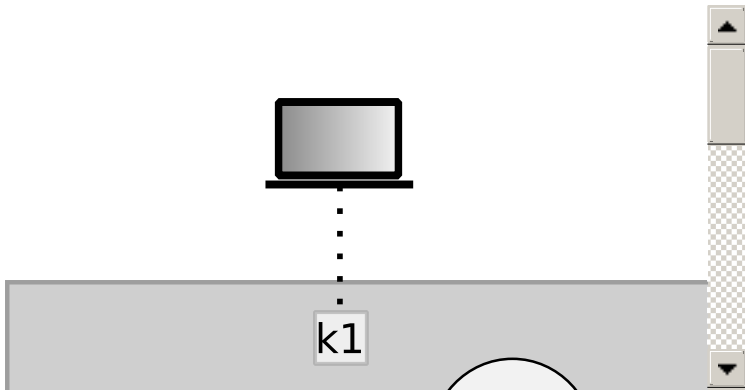
上記のイメージでは、以下のようになります。

1. クライアントは **LON** で Data Grid クラスターに接続します。
2. クライアントは "k1" をキャッシュに書き込みます。
3. **LON** のサイトマスター "n1" は "k1" を **NYC** のサイトマスター (「nA」) に複製する要求を送信します。

Active/Passive では、**NYC** はデータの冗長性を提供します。**LON** の Data Grid クラスターが何らかの理由でオフラインになると、クライアントは **NYC** への要求の送信を開始できます。**LON** をオンラインに戻すと、**NYC** とデータを同期し、クライアントを **LON** に戻すことができます。

Active/Active

以下の図は、Data Grid が2つのサイトでクライアント要求を処理する Active/Active を示しています。



上記のイメージでは、以下のようになります。

1. クライアント A は、LON で Data Grid クラスタに接続します。
2. クライアント A は "k1" をキャッシュに書き込みます。
3. クライアント B は、NYC の Data Grid クラスタに接続します。
4. クライアント B は "k2" をキャッシュに書き込みます。
5. LON および NYC のサイトマスターは、"k1" が NYC に複製され、「k2」が LON に複製されるように要求を送信します。

Active/Active では、NYC と LON の両方は、クライアント要求の処理中にデータをリモートキャッシュに複製します。NYC または LON のいずれかがオフラインになると、クライアントはオンラインサイトへの要求の送信を開始できます。その後、オフラインサイトをオンラインに戻し、状態をプッシュしてデータを同期し、必要に応じてクライアントを切り替えることができます。

2.6.1. 同時書き込みおよび競合エントリ

クライアントが同時に同じエントリに書き込むが、サイトが異なる場合は、Active/Active サイト設定でエントリの競合が発生する可能性があります。

たとえば、クライアント B が NYC の「k1」に書き込むのと同時に、クライアント A は LON の "k1" に書き込みます。この場合、「k1」の値は LON と NYC とで異なります。レプリケーションの発生後、「k1」のどの値がどのサイトに存在するかは保証されません。

データの整合性を確保するために、Data Grid はベクトルロックアルゴリズムを使用して、次の図のように、バックアップ操作中に競合するエントリを検出します。

	LON	NYC
k1=(n/a)	0,0	0,0
k1=2	1,0 -->	1,0 k1=2
k1=3	1,1 <--	1,1 k1=3
k1=5	2,1	1,2 k1=8
	-->	2,1 (conflict)
(conflict)	1,2 <--	

ベクトルクロックは、エントリーへの書き込みごとにインクリメントするタイムスタンプのメタデータです。上記の例では、**0,0** は、"k1" 上のベクトルクロックの初期値を表します。

クライアントは **LON** に "k1=2" を配置し、ベクトルクロックは **1,0** で、Data Grid はこれを **NYC** に複製します。その後、クライアントは **NYC** に "k1=3" を配置し、ベクトルクロックは **1,1** に更新されます。Data Grid はこれを **LON** に複製します。

ただし、クライアントが **NYC** に "k1=8" を配置すると同時に、クライアントが **LON** に "k1=5" を配置すると、Data Grid は競合するエントリーを検出します。これは、"k1" のベクター値が **LON** と **NYC** の間で厳密に大きかったり小さかったりしないためです。

競合するエントリーを見つけると、Data Grid は Java **compareTo(String anotherString)** メソッドを使用してサイト名を比較します。どのキーが優先されるかを決定するために、Data Grid は辞書式順序で他のキーよりも小さいサイト名を選択します。**AAA** という名前のサイトからのキーは、**AAB** という名前のサイトからのキーよりも優先されます。

同じ例に従って、"k1" の競合を解決するために、Data Grid は **LON** からの "k1" の値を使用します。これにより、Data Grid が競合を解決し、値を複製した後に、**LON** と **NYC** の両方で "k1=5" となります。

ヒント

競合するエントリーを解決するための優先順位を表す簡単な方法として、サイト名の前に番号を付けます。たとえば、**1LON** や **2NYC** などです。

参照

- [java.lang.String#compareTo\(\)](#)

2.7. 有効期限とクロスサイトのレプリケーション

Data Grid の有効期限は、エントリーがキャッシュに残っている期間を制御します。

- **ライフスパン** の有効期限は、サイト間のレプリケーションに適しています。エントリーが最大ライフスパンに到達すると、Data Grid はリモートサイトとは独立して有効期限が切れます。
- **max-idle** の有効期限は、クロスサイトのレプリケーションでは動作しません。Data Grid は、キャッシュエントリーがリモートサイトでアイドルタイムアウトに達したタイミングを判断できません。

第3章 クロスサイトレプリケーション用の DATA GRID の設定

複数のサイト間でデータを複製するように Data Grid を設定すると、最初にクラスタートランスポートを設定して、Data Grid クラスタが相互に検出し、サイトマスターが通信できるようにします。その後、Data Grid 設定のキャッシュ定義にバックアップの場所を追加します。

3.1. クロスサイトレプリケーションのためのクラスタートランスポートの設定

Data Grid クラスタがバックアップの場所と通信できるように、JGroups RELAY2 をトランスポート層に追加します。

手順

1. `infinispan.xml` を開いて編集します。
2. 以下の例のように、RELAY2 プロトコルを JGroups スタックに追加します。

```
<jgroups>
  <stack name="xsite" extends="udp">
    <relay.RELAY2 site="LON" xmlns="urn:org:jgroups" max_site_masters="1000"/>
    <remote-sites default-stack="tcp">
      <remote-site name="LON"/>
      <remote-site name="NYC"/>
    </remote-sites>
  </stack>
</jgroups>
```

3. 以下の例のように、スタックを使用するように Data Grid クラスタートランスポートを設定します。

```
<cache-container name="default" statistics="true">
  <transport cluster="{cluster.name}" stack="xsite"/>
</cache-container>
```

4. `infinispan.xml` を保存して閉じます。

参照

- [JGroups RELAY2 スタック](#)
- [Data Grid 設定スキーマ](#)

3.1.1. JGroups RELAY2 スタック

Data Grid クラスタは、クラスタ間の検出および通信に JGroups RELAY2 を使用します。

```
<jgroups>
  <stack name="xsite" ①
    extends="udp"> ②
    <relay.RELAY2 xmlns="urn:org:jgroups" ③
      site="LON" ④
```



```

        max_site_masters="1000"/> 5
    <remote-sites default-stack="tcp"> 6
        <remote-site name="LON"/> 7
        <remote-site name="NYC"/>
    </remote-sites>
</stack>
</jgroups>

```

- 1 Data Grid クラスタートランスポートに使用するプロトコルを宣言する「xsite」という名前のスタックを定義します。
- 2 クラスタ内トラフィックにデフォルトの JGroups UDP スタックを使用します。
- 3 クラスタ間トランスポートのスタックに **RELAY2** を追加します。
- 4 ローカルサイトに名前を付けます。Data Grid は、このサイトのキャッシュ内のデータをバックアップの場所に複製します。
- 5 ローカルクラスタの最大 1000 のサイトマスターを設定します。バックアップ要求でパフォーマンスを最適化するには、Data Grid クラスタのノード数を **max_site_masters** >= に設定する必要があります。
- 6 すべてのサイト名を指定し、クラスタ間トランスポートにデフォルトの JGroups TCP スタックを使用します。
- 7 各リモートサイトにバックアップの場所として名前を付けます。

3.1.2. カスタム JGroups RELAY2 スタック

```

<jgroups>
  <stack name="relay-global" extends="tcp"> 1
    <MPING stack.combine="REMOVE"/>
    <TCPPING initial_hosts="192.0.2.0[7800]" stack.combine="INSERT_AFTER"
stack.position="TCP"/>
  </stack>
  <stack name="xsite" extends="udp">
    <relay.RELAY2 site="LON" xmlns="urn:org:jgroups"
      max_site_masters="10" 2
      can_become_site_master="true"/>
    <remote-sites default-stack="relay-global">
      <remote-site name="LON"/>
      <remote-site name="NYC"/>
    </remote-sites>
  </stack>
</jgroups>

```

- 1 TCP スタックを拡張し、検出に MPING の代わりに TCPPING を使用するカスタムの RELAY2 スタックを追加します。
- 2 サイトマスターの最大数を設定し、必要に応じて追加の RELAY2 プロパティを設定します。JGroups RELAY2 ドキュメントを参照してください。

また、以下のように外部で定義された JGroups スタックファイルを参照することもできます。

```
<stack-file name="relay-global" path="jgroups-relay.xml"/>
```

前述の設定で、**jgroups-relay.xml** は以下のような JGroups スタックを提供します。

```
<config xmlns="urn:org:jgroups"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:org:jgroups http://www.jgroups.org/schema/jgroups-4.2.xsd">

  <!-- Use TCP for inter-cluster transport. -->
  <TCP bind_addr="127.0.0.1"
    bind_port="7200"
    port_range="30"

    thread_pool.min_threads="0"
    thread_pool.max_threads="8"
    thread_pool.keep_alive_time="5000"
  />

  <!-- Use TCPPING for inter-cluster discovery. -->
  <TCPPING timeout="3000"
    initial_hosts="127.0.0.1[7200]"
    port_range="3"
    ergonomics="false"/>

  <!-- Provide other JGroups stack configuration as required. -->
</config>
```

参照

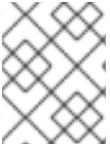
- [Setting Up Data Grid Clusters](#)
- [JGroups RELAY2](#)
- [複数のサイト間のリレー \(RELAY2\)](#)

3.2. キャッシュへのバックアップの場所の追加

Data Grid がデータをそれらの場所にバックアップできるように、リモートサイトの名前を指定します。

手順

1. **backups** 要素をキャッシュ定義に追加します。
2. 各リモートサイトの名前を **backup** 要素で指定します。
たとえば、**LON** 設定で、**NYC** をリモートサイトとして指定します。
3. 各サイトが他のすべてのサイトのバックアップとなるように、前述の手順を繰り返します。たとえば、**NYC** を **LON** のバックアップとして追加せずに、**LON** を **NYC** のバックアップとして追加することはできません。



注記

キャッシュ設定はサイト間で異なり、異なるバックアップストラテジーを使用する場合があります。Data Grid は、キャッシュ名に基づいてデータを複製します。

LONでの "customers" の設定例

```
<replicated-cache name="customers">
  <backups>
    <backup site="NYC" strategy="ASYNC" />
  </backups>
</replicated-cache>
```

NYCでの "customers" の設定例

```
<distributed-cache name="customers">
  <backups>
    <backup site="LON" strategy="SYNC" />
  </backups>
</distributed-cache>
```

参照

- [Data Grid 設定スキーマ](#)

3.3. 異なる名前でのキャッシュへのバックアップ

デフォルトでは、Data Grid は同じ名前のキャッシュ間でデータを複製します。

手順

- **backup-for** を使用して、リモートサイトからのデータをローカルサイトの別の名前でキャッシュに複製します。

たとえば、以下の設定は、LON の "customers" キャッシュを NYC の "eu-customers" キャッシュにバックアップします。

```
<distributed-cache name="eu-customers">
  <backups>
    <backup site="LON" strategy="SYNC" />
  </backups>
  <backup-for remote-cache="customers" remote-site="LON" />
</distributed-cache>
```

3.4. クロスサイトビューの確認

クロスサイトレプリケーション用に Data Grid を設定した後、Data Grid クラスタがクロスサイトビューを正常に形成することを確認する必要があります。

手順

- **ISPN000439: Received new x-site view** メッセージのログメッセージを確認します。

たとえば、**LON** の Data Grid クラスターが、**NYC** の Data Grid クラスターを使用してクロスサイトビューを形成した場合、以下のメッセージが提供されます。

```
INFO [org.infinispan.XSITE] (jgroups-5,${server.hostname}) ISPN000439: Received new x-site view:
[NYC]
INFO [org.infinispan.XSITE] (jgroups-7,${server.hostname}) ISPN000439: Received new x-site view:
[NYC, LON]
```

3.5. クロスサイトレプリケーション用の **HOT ROD** クライアントの設定

異なるサイトで Data Grid クラスターを使用するように Hot Rod クライアントを設定します。

hotrod-client.properties

```
# Servers at the active site
infinispan.client.hotrod.server_list = LON_host1:11222,LON_host2:11222,LON_host3:11222

# Servers at the backup site
infinispan.client.hotrod.cluster.NYC =
NYC_hostA:11222,NYC_hostB:11222,NYC_hostC:11222,NYC_hostD:11222
```

ConfigurationBuilder

```
ConfigurationBuilder builder = new ConfigurationBuilder();
builder.addServers("LON_host1:11222;LON_host2:11222;LON_host3:11222")
    .addCluster("NYC")

    .addClusterNodes("NYC_hostA:11222;NYC_hostB:11222;NYC_hostC:11222;NYC_hostD:11222")
```

ヒント

以下の方法を使用して、Hot Rod クライアントをデフォルトのクラスターまたは別のサイトのクラスターに切り替えます。

- **RemoteCacheManager.switchToDefaultCluster()**
- **RemoteCacheManager.switchToCluster(\${site.name})**

参照

- [org.infinispan.client.hotrod.configuration package description](#)
- [org.infinispan.client.hotrod.configuration.ConfigurationBuilder](#)
- [org.infinispan.client.hotrod.RemoteCacheManager](#)

第4章 クロスサイトレプリケーション操作の実行

サイトをオンラインとオフラインにします。キャッシュ状態をリモートサイトに転送します。

4.1. CLI を使用したクロスサイト操作の実行

Data Grid コマンドラインインターフェースを使用すると、Data Grid サーバーにリモートで接続したり、サイトを管理したり、状態遷移をバックアップの場所にプッシュしたりできます。

前提条件

- Data Grid CLIを起動している。
- 実行中のData Gridクラスターに接続している。

4.1.1. バックアップ場所のオフラインおよびオンライン化

バックアップ場所を手動でオフラインにし、オンラインに戻します。

手順

1. Data Grid への CLI 接続を作成します。
2. **site status** コマンドを使用して、バックアップの場所がオンラインかオフラインかを確認します。

```
//containers/default]> site status --cache=cacheName --site=NYC
```



注記

--site はオプションの引数です。設定されていない場合、CLI はすべてのバックアップ場所を返します。

3. 次のようにバックアップ場所を管理します。
 - **bring-online** コマンドを使用して、バックアップの場所をオンラインにします。

```
//containers/default]> site bring-online --cache=customers --site=NYC
```

- **take-offline** コマンドを使用して、バックアップの場所をオフラインにします。

```
//containers/default]> site take-offline --cache=customers --site=NYC
```

詳細と例については、**help site** コマンドを実行してください。

4.1.2. バックアップ場所への状態のプッシュ

キャッシュの状態をリモートのバックアップ場所に転送します。

手順

1. Data Grid への CLI 接続を作成します。

2. 次の例のように、**site**コマンドを使用して状態の転送をプッシュします。

```
//containers/default]> site push-site-state --cache=cacheName --site=NYC
```

詳細と例については、**help site** コマンドを実行してください。

参照

[Data Grid コマンドラインインターフェース](#)

4.2. REST API を使用したクロスサイト操作の実行

Data Grid サーバーは、クロスサイト操作を実行できるようにする REST API を提供します。

4.2.1. すべてのバックアップロケーションのステータス取得

GET リクエストですべてのバックアップロケーションのステータスを取得します。

```
GET /v2/caches/{cacheName}/x-site/backups/
```

Data Grid は、以下の例のように、各バックアップロケーションのステータスを JSON 形式で応答します。

```
{
  "NYC": "online",
  "LON": "offline"
}
```

表4.1 リターンステータス

値	説明
online	ローカルクラスター内のすべてのノードには、バックアップの場所を含むクロスサイトビューがあります。
offline	ローカルクラスター内のノードには、バックアップの場所とのクロスサイトビューがありません。
mixed	ローカルクラスター内の一部のノードにはバックアップの場所を含むクロスサイトビューがあり、ローカルクラスター内の他のノードにはクロスサイトビューがありません。応答は、各ノードのステータスを示します。

4.2.2. 特定のバックアップ場所のステータスの取得

GET リクエストでバックアップロケーションのステータスを取得する。

```
GET /v2/caches/{cacheName}/x-site/backups/{siteName}
```

Data Gridは、以下の例のように、サイト内の各ノードのステータスをJSON形式で応答します。

```
{
  "NodeA":"offline",
  "NodeB":"online"
}
```

表4.2 リターンステータス

値	説明
online	ノードはオンラインです。
offline	ノードはオフラインです。
failed	ステータスを取得できません。リモートキャッシュがシャットダウンしているか、リクエスト中にネットワークエラーが発生した可能性があります。

4.2.3. バックアップ先をオフラインにする

POSTリクエストと**?action=take-offline**パラメータを使用して、バックアップの場所をオフラインにします。

```
POST /v2/caches/{cacheName}/x-site/backups/{siteName}?action=take-offline
```

4.2.4. バックアップ場所をオンラインにする

?action=bring-onlineパラメータを使用してバックアップ場所をオンラインにします。

```
POST /v2/caches/{cacheName}/x-site/backups/{siteName}?action=bring-online
```

4.2.5. バックアップ場所への状態のプッシュ

?action=start-push-stateパラメータを使用して、キャッシュ状態をバックアップ場所にプッシュします。

```
POST /v2/caches/{cacheName}/x-site/backups/{siteName}?action=start-push-state
```

4.2.6. 状態転送のキャンセル

?action=cancel-push-stateパラメータを使用して状態転送操作をキャンセルします。

```
POST /v2/caches/{cacheName}/x-site/backups/{siteName}?action=cancel-push-state
```

4.2.7. 状態転送ステータスの取得

?action=push-state-statusパラメータを使用して状態転送操作のステータスを取得します。

```
GET /v2/caches/{cacheName}/x-site/backups?action=push-state-status
```

Data Gridは、以下の例のように、各バックアップ拠点の状態移行の状況をJSON形式で応答します。

```
{
  "NYC":"CANCELED",
  "LON":"OK"
}
```

表4.3 リターンステータス

値	説明
SENDING	バックアップ場所への状態転送が進行中です。
OK	状態の転送が正常に完了しました。
ERROR	状態転送でエラーが発生しました。ログファイルを確認してください。
CANCELLING	状態移行のキャンセルが進行中です。

4.2.8. 状態転送ステータスのクリア

?action=clear-push-state-statusパラメーターを使用して送信サイトの状態転送ステータスをクリアします。

```
POST /v2/caches/{cacheName}/x-site/local?action=clear-push-state-status
```

4.2.9. オフラインテイク条件の変更

特定の条件が満たされると、サイトはオフラインになります。オフラインにするパラメータを変更して、バックアップロケーションが自動的にオフラインになるタイミングを制御します。

手順

1. **GET**リクエストと**take-offline-config**パラメータで設定されたテイクオフラインパラメータを確認します。

```
GET /v2/caches/{cacheName}/x-site/backups/{siteName}/take-offline-config
```

Data Gridのレスポンスには、以下のように**after_failures**と**min_wait**フィールドがあります。

```
{
  "after_failures": 2,
  "min_wait": 1000
}
```

2. **PUT**リクエストの本文のオフライン取得パラメータを変更します。


```
PUT /v2/caches/{cacheName}/x-site/backups/{siteName}/take-offline-config
```

4.2.10. 受信サイトからの状態転送のキャンセル

2つのバックアップ場所間の接続が切断された場合、プッシュを受信しているサイトでの状態転送をキャンセルできます。

?action=cancel-receive-stateパラメータで、リモートサイトからの状態転送をキャンセルし、ローカルキャッシュの現在の状態を維持する。

```
POST /v2/caches/{cacheName}/x-site/backups/{siteName}?action=cancel-receive-state
```

4.2.11. バックアップ場所のステータスの取得

GET要求により、キャッシュ・マネージャからすべてのバックアップ・ロケーションのステータスを取得します。

```
GET /rest/v2/cache-managers/{cacheManagerName}/x-site/backups/
```

Data Gridは、以下の例のようにJSON形式でステータスを応答します。

```
{
  "SFO-3":{
    "status":"online"
  },
  "NYC-2":{
    "status":"mixed",
    "online":[
      "CACHE_1"
    ],
    "offline":[
      "CACHE_2"
    ]
  }
}
```

表4.4 リターンステータス

値	説明
online	ローカルクラスター内のすべてのノードには、バックアップの場所を含むクロスサイトビューがあります。
offline	ローカルクラスター内のノードには、バックアップの場所とのクロスサイトビューがありません。

値	説明
mixed	ローカルクラスター内の一部のノードにはバックアップの場所を含むクロスサイトビューがあり、ローカルクラスター内の他のノードにはクロスサイトビューがありません。応答は、各ノードのステータスを示します。

4.2.12. バックアップ先をオフラインにする

?action=take-offlineパラメータで、バックアップローケーションをオフラインにします。

```
POST /rest/v2/cache-managers/{cacheManagerName}/x-site/backups/{siteName}?action=take-offline
```

4.2.13. バックアップ場所をオンラインにする

?action=bring-onlineパラメーターを使用してバックアップ場所をオンラインにします。

```
POST /rest/v2/cache-managers/{cacheManagerName}/x-site/backups/{siteName}?action=bring-online
```

4.2.14. 状態転送の開始

?action=start-push-stateパラメーターを使用して、すべてのキャッシュの状態をリモートサイトにプッシュします。

```
POST /rest/v2/cache-managers/{cacheManagerName}/x-site/backups/{siteName}?action=start-push-state
```

4.2.15. 状態転送のキャンセル

?action=cancel-push-stateパラメーターを使用して、進行中の状態転送操作をキャンセルします。

```
POST /rest/v2/cache-managers/{cacheManagerName}/x-site/backups/{siteName}?action=cancel-push-state
```

4.3. JMX を使用したクロスサイト操作の実行

Data Grid は、状態遷移のプッシュやサイトのオンライン化など、クロスサイトの操作から JMX ツールを提供します。

4.3.1. JMX MBean を登録するための Data Grid の設定

Data Grid は、統計の収集と管理操作の実行に使用できる JMX MBean を登録できます。ただし、統計を JMX とは別に有効にしないと、Data Grid はすべての統計情報属性に **0** 値を提供します。

手順

- JMX を宣言的またはプログラムの的に有効にします。

宣言的に

```
<cache-container>
  <jmx enabled="true" /> ❶
</cache-container>
```

- ❶ Data Grid JMX MBean を登録します。

プログラムで

```
GlobalConfiguration globalConfig = new GlobalConfigurationBuilder()
    .jmx().enable() ❶
    .build();
```

- ❶ Data Grid JMX MBean を登録します。

4.3.2. クロスサイト操作の実行

JMX クライアント経由でクロスサイト操作を実行します。

前提条件

- JMX MBean を登録するように Data Grid を設定します。

手順

1. 任意の JMX クライアントで Data Grid に接続します。
2. 以下の MBean から操作を呼び出します。
 - **XSiteAdmin** は、キャッシュのクロスサイト操作を提供します。
 - **GlobalXSiteAdminOperations** は、Cache Manager のクロスサイト操作を提供します。
たとえば、サイトをオンラインに戻すには、**bringSiteOnline(siteName)** を呼び出します。

利用可能なクロスサイト操作の詳細については、**Data Grid JMX Components** のドキュメントを参照してください。

参照

- [XSiteAdmin MBean](#)
- [GlobalXSiteAdminOperations MBean](#)

第5章 グローバル DATA GRID クラスターの監視およびトラブルシューティング

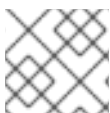
Data Grid は、JMX または Data Grid サーバーの `/metrics` エンドポイント経由のクロスサイトレプリケーション操作の統計を提供します。

クロスサイトのレプリケーション統計はキャッシュレベルで利用できるため、キャッシュの統計を明示的に有効にする必要があります。同様に、JMX 経由で統計を収集する場合は、Data Grid が MBean を登録するように設定する必要があります。

Data Grid には `org.infinispan.XSITE` ログカテゴリも含まれるため、ネットワークおよび状態遷移操作に関する一般的な問題を監視し、トラブルシューティングすることができます。

5.1. DATA GRID 統計の有効化

Data Grid を使用すると、キャッシュマネージャーおよびキャッシュの統計を有効にできます。ただし、Cache Manager の統計を有効にしても、制御するキャッシュの統計が有効になりません。キャッシュの統計を明示的に有効にする必要があります。



注記

Data Grid サーバーは、デフォルトで Cache Manager の統計を有効にします。

手順

- 統計を宣言的またはプログラムで有効にします。

宣言的に

```
<cache-container statistics="true"> ①
  <local-cache name="mycache" statistics="true"/> ②
</cache-container>
```

- ① キャッシュマネージャーの統計を有効にします。
- ② 名前付きキャッシュの統計を有効にします。

プログラムで

```
GlobalConfiguration globalConfig = new GlobalConfigurationBuilder()
    .cacheContainer().statistics(true) ①
    .build();

...

Configuration config = new ConfigurationBuilder()
    .statistics().enable() ②
    .build();
```

- ① キャッシュマネージャーの統計を有効にします。

- 2 名前付きキャッシュの統計を有効にします。

5.2. DATA GRID メトリクスの有効化

ゲージとヒストグラムをエクスポートするように Data Grid を設定します。

手順

- メトリクスを宣言的またはプログラムの設定する。

宣言的に

```
<cache-container statistics="true"> 1
  <metrics gauges="true" histograms="true" /> 2
</cache-container>
```

- 1 Cache Manager の統計を計算し、収集します。
- 2 統計をゲージおよびヒストグラムメトリクスとして収集します。

プログラムで

```
GlobalConfiguration globalConfig = new GlobalConfigurationBuilder()
  .statistics().enable() 1
  .metrics().gauges(true).histograms(true) 2
  .build();
```

- 1 Cache Manager の統計を計算し、収集します。
- 2 統計をゲージおよびヒストグラムメトリクスとして収集します。

5.2.1. Data Grid メトリクスの収集

Prometheus などのモニタリングツールを使用して、Data Grid メトリクスを収集します。

前提条件

- 統計を有効にします。統計を有効にしないと、Data Grid はメトリクスに **0** と **-1** の値を指定します。
- 必要に応じて、ヒストグラムを有効にします。デフォルトでは、Data Grid はゲージを生成しますが、ヒストグラムは生成しません。

手順

- Prometheus (OpenMetrics) 形式でメトリクスを取得します。

```
$ curl -v http://localhost:11222/metrics
```

- MicroProfile JSON 形式でメトリクスを取得します。

```
$ curl --header "Accept: application/json" http://localhost:11222/metrics
```

次のステップ

Data Grid メトリクスを収集するようにモニタリングアプリケーションを設定します。たとえば、以下を **prometheus.yml** に追加します。

```
static_configs:
  - targets: ['localhost:11222']
```

参照資料

- [Prometheus Configuration](#)
- Data Grid 統計の有効化

5.3. JMX MBean を登録するための DATA GRID の設定

Data Grid は、統計の収集と管理操作の実行に使用できる JMX MBean を登録できます。ただし、統計を JMX とは別に有効にしないと、Data Grid はすべての統計情報属性に **0** 値を提供します。

手順

- JMX を宣言的またはプログラムの有効にします。

宣言的に

```
<cache-container>
  <jmx enabled="true" /> ❶
</cache-container>
```

- ❶ Data Grid JMX MBean を登録します。

プログラムで

```
GlobalConfiguration globalConfig = new GlobalConfigurationBuilder()
    .jmx().enable() ❶
    .build();
```

- ❶ Data Grid JMX MBean を登録します。

5.3.1. クロスサイトレプリケーション用の JMX MBean

Data Grid は、統計を収集し、リモート操作を実行できるクロスサイトレプリケーションに JMX MBean を提供します。

org.infinispan:type=Cache コンポーネントは、以下の JMX MBean を提供します。

- **XSiteAdmin** は、特定のキャッシュインスタンスに適用されるクロスサイト操作を公開しません。

- **StateTransferManager** は、状態遷移操作の統計を提供します。
- **InboundInvocationHandler** は非同期および同期のクロスサイトリクエストの統計および操作を提供します。

org.infinispan:type=CacheManager コンポーネントには以下の JMX MBean が含まれます。

- **GlobalXSiteAdminOperations** は、キャッシュコンテナのすべてのキャッシュに適用されるクロスサイト操作を公開します。

JMX MBean および利用可能な操作および統計の説明に関する詳細は、**Data Grid JMX Components** のドキュメントを参照してください。

参照

[Data Grid JMX Components](#)

5.4. ログの収集およびクロスサイトレプリケーションのトラブルシューティング

Data Grid のクロスサイトレプリケーションに関連する問題を診断し、解決します。Data Grid コマンドラインインターフェース (CLI) を使用して、実行時にログレベルを調整し、クロスサイトのトラブルシューティングを実行します。

手順

1. **\$RHDG_HOME** でターミナルを開きます。
2. Data Grid CLI コネクションを作成します。
3. 必要な場合には、ランタイムのロギングレベルを調整して DEBUG メッセージをキャプチャします。
たとえば、以下のコマンドは、org.infinispan.XSITE カテゴリの DEBUG ログメッセージを有効にします。

```
[//containers/default]> logging set --level=DEBUG org.infinispan.XSITE
```

次に、**`\${rhdg.server.root}/log** ディレクトリーで、クロスサイトメッセージの Data Grid ログファイルを確認できます。

4. **site** コマンドを使用して、バックアップの場所のステータスを表示し、トラブルシューティングを実行します。

たとえば、バックアップの場所に "LON" を使用する "customers" キャッシュのステータスを確認します。

```
[//containers/default]> site status --cache=customers
{
  "LON" : "online"
}
```

Data Grid CLI を使用してトラブルシューティングするもう1つのシナリオは、状態遷移操作中にバックアップの場所間のネットワーク接続が切断された場合です。

この場合、状態遷移を受信する Data Grid クラスターは、操作が完了するまで継続的に待機します。この場合、受信サイトへの状態遷移をキャンセルして、通常の運用状態に戻す必要があります。

たとえば、以下のように「NYC」の状態遷移をキャンセルします。

```
[//containers/default]> site cancel-receive-state --cache=mycache --site=NYC`
```

参照

- [Data Grid Server Troubleshooting](#)
- [Working with Data Grid Server Logs](#)

5.4.1. クロスサイトログメッセージ

クロスサイトレプリケーションに関連するログメッセージのユーザーアクションを検索します。

ログレベル	識別子	メッセージ	説明
DEBUG	ISPN000400	ノード null が疑われました	Data Grid は、バックアップの場所に到達できない場合にこのメッセージを出力します。サイトがオンラインであることを確認し、ネットワークステータスを確認します。
INFO	ISPN000439	新しい x-site ビューの受信: \${site.name}	Data Grid は、サイトがグローバルクラスターに参加および離脱するときに、このメッセージを出力します。
INFO	ISPN100005	サイト \${site.name} はオンラインです。	Data Grid は、サイトがオンラインになると、このメッセージを出力します。
INFO	ISPN100006	サイト \${site.name} はオフラインです。	Data Grid は、サイトがオフラインになると、このメッセージを出力します。サイトを手動でオフラインにしなかった場合、このメッセージは障害が発生したことを示している可能性があります。ネットワークのステータスを確認し、サイトをオンラインに戻してみてください。
WARN	ISPN000202	キャッシュ \${cache.name} のデータをサイト \${site.name} にバックアップする際の問題:	Data Grid は、例外とともに状態転送操作で問題が発生した場合に、このメッセージを出力します。必要な場合は、Data Grid のロギングを調整して、より詳細なロギングメッセージを取得します。

ログレベル	識別子	メッセージ	説明
WARN	ISPN000289	X-Site 状態チャンクを \${site.name} に送信できません。	状態遷移操作中に Data Grid が キャッシュエントリーのバッチを 転送できないことを示します。サ イトがオンラインであることを確 認し、ネットワークステータスを 確認します。
WARN	ISPN000291	X-Site 状態チャンクを適用できま せん。	状態遷移操作中に Data Grid が キャッシュエントリーのバッチを 適用できないことを示します。サ イトがオンラインであることを確 認し、ネットワークステータスを 確認します。
WARN	ISPN000322	サイト \${site.name} への x サイト の状態遷移を再開できません。	Data Grid が、バックアップ場所 への状態遷移操作を再開できな いことを示します。サイトがオンラ インであることを確認し、ネット ワークステータスを確認します。
ERROR	ISPN000477	サイト \${site.name} に対して操作 \${operation.name} を実行できま せん	Data Grid がバックアップの場所 で操作を正常に完了できないこと を示します。必要な場合は、Data Grid のロギングを調整して、より 詳細なロギングメッセージを取得 します。
FATAL	ISPN000449	Xsite の状態遷移タイムアウトは、 1 以上でなければなりません。	timeout 属性の値が 0 または負の 値である場合の結果です。キャッ シュ定義の状態遷移設定の timeout 属性に 1 以上の値を指定 します。
FATAL	ISPN000450	再試行間の Xsite 状態遷移の待機 時間は、1 以上である必要がありま す。	wait-time 属性の値が 0 または負 の値である場合の結果です。 キャッシュ定義の状態遷移設定の wait-time 属性に 1 以上の値を指 定します。
FATAL	ISPN000576	ローカルキャッシュでクロスサイ トレプリケーションは利用できま せん。	クロスサイトレプリケーション は、ローカルキャッシュモードで は機能しません。ローカルキャッ シュ定義からバックアップ設定を 削除するか、分散またはレプリ ケートされたキャッシュモードを 使用します。

