IoT向けスケールアウト型データベースGridDBで実現するサイト間データベースレプリケーション

Site-to-Site Database Replication Function of GridDB Scale-Out Database

金輪　拓也 KANAWA Takuya　　近藤　雄二 KONDO Yuji

広範囲なサービスを提供する企業は，サイト（拠点）ごとに収集したIoT（Internet of Things）データを異なるサイトに複製し，大規模災害に備えるとともに，複数のサイトを横断したデータ分析を行うことが求められている。

東芝デジタルソリューションズ（株）は，IoTデータ管理に適したスケールアウト型データベース（DB）であるGridDBを提供している。この度，GridDBのサイト内レプリケーション（データの複製と同期）機能を拡張し，高い処理性能とサイト間のデータ一貫性を保証したサイト間DBレプリケーションを実現した。これにより，IoTデータに対して障害時の復旧とサイト横断分析を両立させた運用が可能となる。

Many companies delivering a broad range of services have found it necessary to implement cross-site data analyses as well as precautionary measures in preparation for emergencies via data, which are collected from Internet of Things (IoT) devices at individual sites and replicated across other sites.

Toshiba Digital Solutions Corporation supplies GridDB, a scale-out database for appropriate management of various data from IoT devices. We have developed and released a site-to-site database replication function that provides excellent processing performance and ensures data consistency between different sites by expanding the existing intra-site replication function of GridDB, thereby enabling disaster recovery operations and cross-site data analyses using replicas of IoT data.

1. まえがき

広範囲なサービスを提供する企業は，サイト（拠点）ごとに収集したIoTデータをサイト間で複製しておくことで，災害時の復旧の迅速化や，複数のサイトを横断してのデータ分析などを可能にする，サイト間DBレプリケーションの実現が求められている。

しかし，大規模かつ高頻度で更新されるIoTデータを対象としたサイト間DBレプリケーションは，処理性能の低下やサイト間のデータ一貫性の確保が難しいといった問題があった。

東芝デジタルソリューションズ（株）は，IoTデータの管理に適したスケールアウト型DBであるGridDB[(1)]を提供しており，DBをクラスター管理することで高性能かつ高信頼性を両立させている。データはクラスター内のノード間で自動的にレプリケーション（データの複製と同期）され，ノード障害が発生した場合はレプリカ（複製）を利用してすぐに復旧できる。

これまで，GridDBのレプリケーションは同一サイトのクラスター内に限定されていたが，今回，異なるサイトのクラスター間同士でも可能とし，高い処理性能とサイト間データ一貫性を保証したサイト間DBレプリケーションを実現した。

ここでは，GridDBの特徴と，サイト間DBレプリケーションを実現するための課題及び方法について述べる。

2. サイト間DBレプリケーションの問題

サイト間DBレプリケーションとは，異なるサイトにあるプライマリー DB（稼働中）とスタンバイDB（待機中）の間で，レプリケーションを行う仕組みである。**図1**のように，サイト間で継続的にレプリケーションを行って異なるサイトのスタン

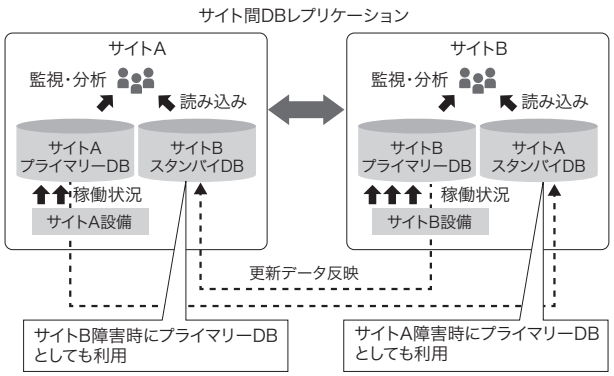


図1. サイト間DBレプリケーションの概要
レプリカデータを用いて，RTO・RPOの短縮とデータの有効利用ができる。
Outline of site-to-site database replication function

バイDBにレプリカデータとして複製しておき，サイト障害が発生した場合はレプリカデータを用いてシステムを復旧させることができる。またデータの一貫性が担保される。

システム障害の復旧戦略の指標として，RTO（Recovery Time Objective）及び RPO（Recovery Point Objective）がある。RTOは災害後にシステムが復旧するまでの目標時間，RPOは災害時に許容されるデータ損失の範囲であり，遡ってどの時点のデータまで復元するかを示す。

サイト間データの冗長化には，コールドスタンバイ方式とホットスタンバイ方式がある。

コールドスタンバイ方式はスタンバイDBをオフラインとし，災害発生時にオンラインにする方式であり，RTOやRPOは数時間から日単位となることが多い。また，災害発生までオフラインであるため，正常稼働時のレプリカデータ利用の範囲も限定される。一方，ホットスタンバイ方式は，プライマリー DBとスタンバイDBの両方をオンラインで稼働させることが特徴であり，レプリケーションをリアルタイム又は短時間で実行することでRTO及びRPOを数秒～数分に短縮できる。また，レプリカデータをオンラインで参照できるため，複数サイトのDBのレプリカを保持しておけば，サイトを横断したリアルタイムの監視及び分析を実現できる。

しかし，ホットスタンバイ方式を，データ量が多く頻繁に更新される特性を持つIoTデータに適用すると，プライマリー DBに負荷が掛かり，処理性能が大きく低下することがあった。また，DB間のデータ一貫性を保証するのが難しいという問題もあった。

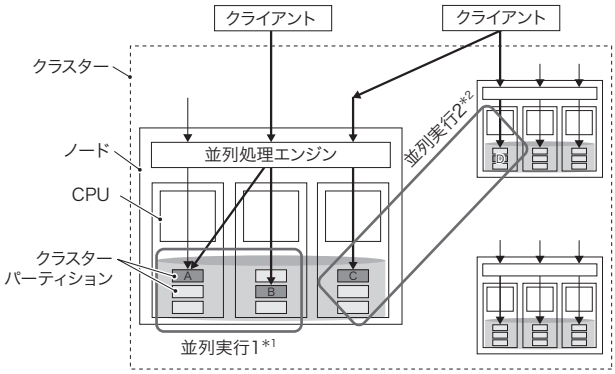
このような背景から，IoTデータに対してホットスタンバイ方式を用いた場合も，高性能と高信頼性を両立できるDBが求められている。

3. GridDBの特徴

GridDBは，大量のIoTデータの蓄積と集計・分析が可能なDBであり，以下の特徴を持つ。

⑴ クラスターを用いたDB管理　　複数のノードを組み合わせたクラスターと呼ぶ構成単位でDBを管理する。クラスター内のいずれかのノードに障害が発生した場合でも自動的に検知し，クラスターを即座に再構成することでサービスを継続できる。一般的には，クラスターは同一サイト内のノードで構成される。

⑵ 自動データ分散　　データは，**図2**のように自動的にクラスターパーティションという複数の断片（部分DB）に分割され，それらはクラスター内の各ノードに適切に配置される。これにより，CPU（コア）及びノードのスケーラビリティーを向上させることができる。



＊1　クライアントが，同じノード内の異なるCPUが担当するクラスターパーティションA・Bに，同時にアクセスする場合である。並列処理エンジンが処理を振り分けて，並列実行される

＊2　クライアントが，異なるノードのクラスターパーティションC・Dに，同時にアクセスする場合である。別ノードのCPUが処理するので，待ち時間なく並列処理される

図2. 自動データ分散の概要

クラスター内ノード間でデータを適切に分散配置し，CPUとノードのスケーラビリティーを向上させる。

Configuration of automatic data distribution

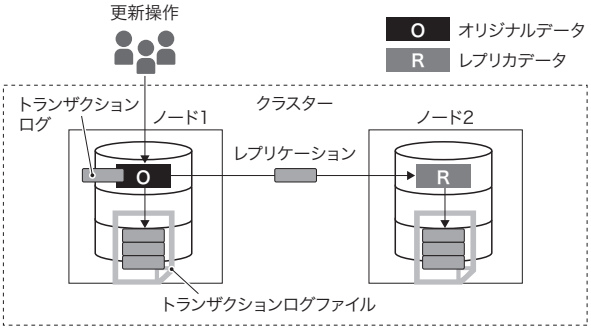


図3. GridDBのクラスター内レプリケーション機能

GridDBは同一クラスター内のノード間でレプリケーションを実行する。

Data replication in cluster

⑶ クラスター内レプリケーション機能　　GridDBは，更新操作に対して即時レプリケーションを行うことで，クラスター内のレプリカデータを常に同期状態に保つ，クラスター内レプリケーション機能を持っている（**図3**）。

図4はクラスター内の一つのノードで障害が発生した場合の挙動を示したものである。クラスターはノードの障害を自動で検知し，正常なノードのレプリカデータに即座に切り替えることで，サービスの停止なく運用を継続できる。

また，障害発生によりクラスター内のレプリカが一時的に失われた場合は，自動的にレプリケーションが実行され別ノードにレプリカの再作成が行われる。これにより，クラスター内のレプリカを常に維持できる。

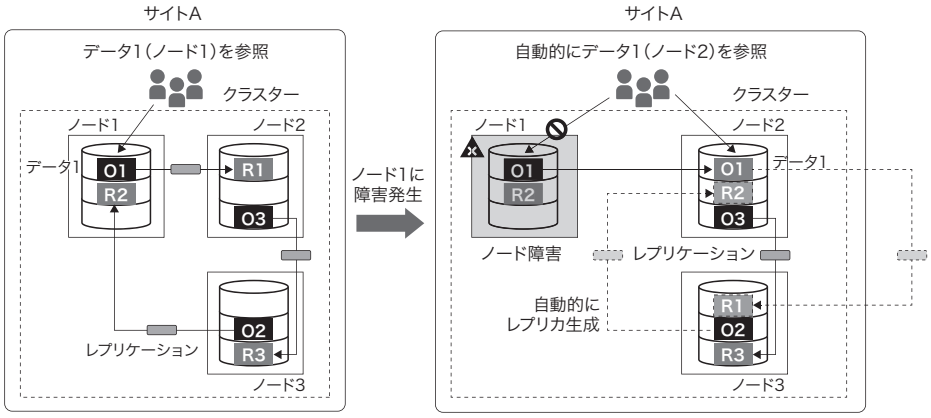


図4. クラスター内ノードに障害が発生した場合の挙動

自動的にオリジナルデータからレプリカデータへの切り替えが行われ，サービスを継続できる。

Behavior of cluster nodes at time of failure

4. GridDBによるサイト間DBレプリケーションの課題

3章で述べたGridDBのクラスター内レプリケーション機能は，同一サイト内のノードに対して実行される。これをサイト間に拡張するには，別サイトのクラスター間でレプリケーションを行う必要がある。以下に，その課題を述べる。

4.1 クラスター間のレプリケーション欠損への対応

サイトが異なるクラスター間でレプリケーションを行う場合は，ネットワーク環境の不安定さやサイト障害などの要因で，レプリケーション欠損が発生することがある。レプリケーション欠損とはプライマリーDBのトランザクションログの一部がスタンバイDBに反映されず，DB間のデータ一貫性が失われることである。

レプリケーションの実現には，データ更新時にメモリー上で即時実行するメモリーベース方式と，定期的に実行するファイルベース方式がある。これらの挙動の違いを**図5**に示す。

メモリーベース方式には，三つの同期プロトコルがある。

(1) 同期型　　プライマリーDBがスタンバイDBのレプリケーション成功まで待つ。

(2) 準同期型　　プライマリーDBがスタンバイDBとの通信成功まで待つが，レプリケーション成功までは待たない。

(3) 非同期型　　プライマリーDBがスタンバイDBの応答を一切待たない。

一方，ファイルベース方式は，プライマリーDBで永続化されたトランザクションログファイルを定期的にスタンバイDBに転送して同期を行う。

クラスター間のレプリケーション方式は，これらのいずれかが用いられる。**表1**に，方式ごとの特徴を比較して示す。

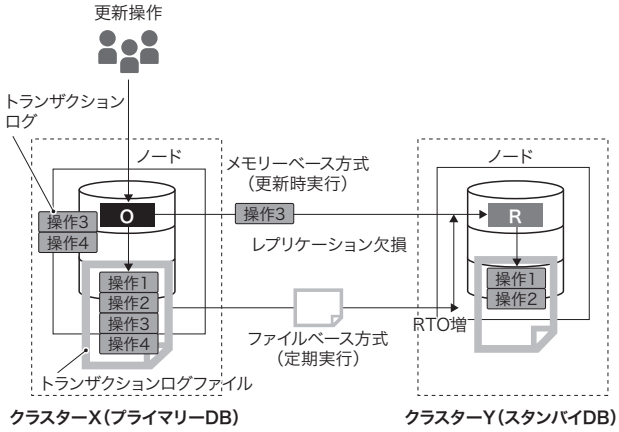


図5. クラスター間レプリケーション方式

クラスター間のレプリケーション方式では，プライマリーDBのトランザクションログが適切にスタンバイDBに複製されないレプリケーション欠損が発生するリスクがある。

Data replication between clusters

どの方式でも，プライマリーDBの性能，データ一貫性(レプリケーション欠損がない)，小さいRTOを全て満たすことは難しい。

4.2 クラスター間のデーター貫性の保証

DB間のレプリケーションは，オリジナルだけデータ更新できる片方向レプリケーションと，オリジナルとレプリカの両方でデータ更新が可能な双方向レプリケーションの2種類がある。しかし，双方向レプリケーションは，異なるDB上で同時に更新操作が行われるとデータの一貫性を維持することが難しく，多くのDB製品ではサポートされていない。GridDBのクラスター内レプリケーション機能も片方向レプリケーション方式であり，同一クラスター内のレプリカノードの更新を

表1. 各レプリケーション方式の特徴の比較

Features of two types of data replication functions

方式	プロトコル	プライマリーDBの性能劣化	レプリケーション欠損発生確率	RTO
メモリーベース	同期型	大	なし	小（秒単位）
	準同期型	中	中	小（秒単位）
	非同期型	小	大	小（秒単位）
ファイルベース	任意（不問）	小	なし	大（分単位）

＊網掛けした項目は，要求を満たさない

禁止している。サイト間クラスター同士を片方向レプリケーションで実現するには，スタンバイDBとなるクラスターを構成する全てのノードに対する更新操作を禁止する制御が必要になる。

4.3 スタンバイDBのレプリケーション適用判断

サイトをまたがるクラスター間レプリケーションは，スタンバイDBとなるクラスターで不完全なトランザクションログを受信するリスクがある。この結果，ログを誤って適用してしまうとDBの整合性が失われ，DB破壊につながるおそれがあるため，安全なレプリケーション手順を確立することが重要となる。

4.4 サイト間DBレプリケーション中の構成変更への対応

プライマリーDBとスタンバイDBは，構成ノード数が同じである必要はなく，異なるノード数で構成することが望ましい場合もある。例えば，スタンバイDBのノード数をプライマリーDBよりも少なくすることで，コストを抑えながら必要な冗長性を確保できる。また，サービスを中断することなく構成変更が行えれば，システムの可用性を維持しながら，必要に応じてリソースを追加又は削減する柔軟性を提供できる。

5. GridDBによるサイト間DBレプリケーションの実現

4章で述べたそれぞれの課題を解決する機能を開発し，GridDBを用いたサイト間DBレプリケーションを実現した。これらの概要を**図6**に示し，機能ごとに詳しく説明する。

(1) 機能1：メモリー・ファイル併用型レプリケーション機能　　メモリーベースとファイルベースの両方のクラスター間レプリケーションをサポートするメモリー・ファイル併用型レプリケーション機能を開発した。通常はメモリーベース方式（準同期か非同期）でクラスター間レプリケーションを行い，通信障害などで一定時間レプリケーション失敗が続いた場合，自動的にファイルベース方式へ切り替える。これにより，プライマリーDBの処理性能を劣化させずにレプリケーション欠損を防止できる(注1)。

(2) 機能2：リードクラスター機能　　クライアントからクラスターへの更新操作を禁止するリードクラスター機能を開発した。この機能をスタンバイDBとなるクラスターに適用することで，プライマリーDBとスタンバイDBで片方向レプリケーションを実現し，DB間のデータ一貫性を保証できる。

(3) 機能3：オンラインサイト間レプリケーション機能

スタンバイDBで，他サイトからのレプリケーションをオンラインで安全に実行できる機能を開発した。スタン

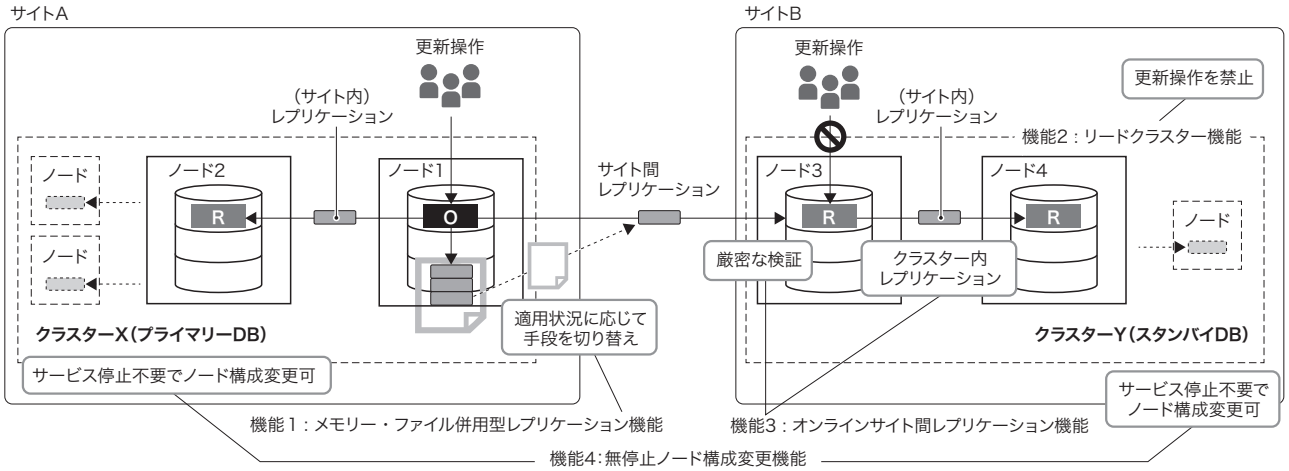


図6. GridDBによるサイト間DBレプリケーションの概要

クラスター同士でレプリケーションを行う機能を開発し，クラスター間のデータ一貫性と処理性能を両立させた。

Outline of GridDB site-to-site database replication function

（注1） 2024年7月時点のGridDBはファイルベース方式だけサポートしている。

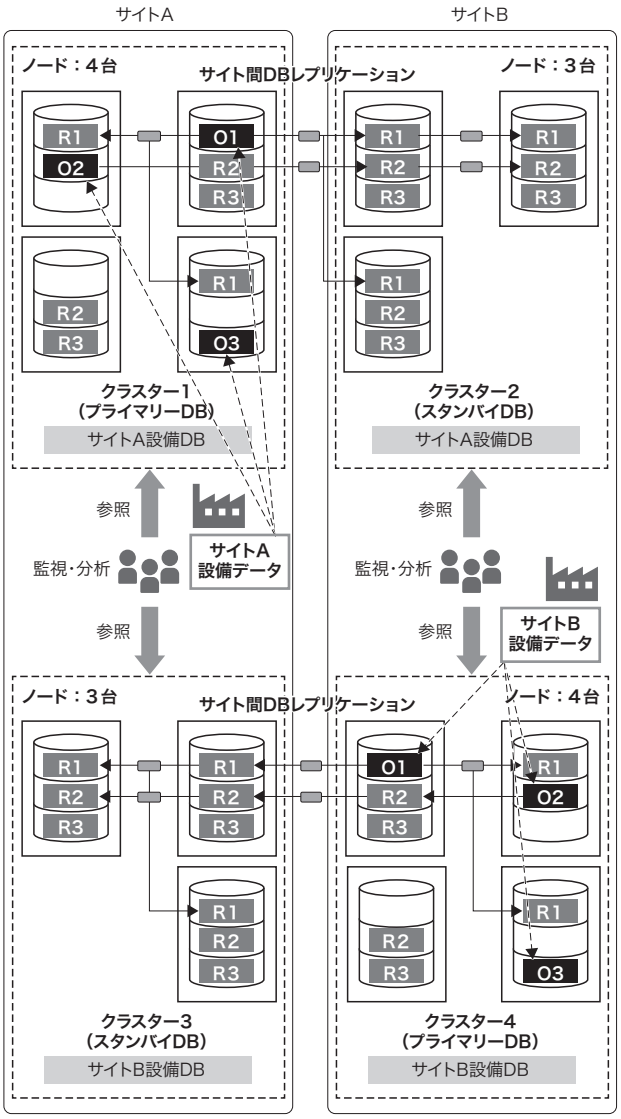


図7. IoTデータに対するサイト間DBレプリケーションの適用例
各サイトのIoTデータをクラスター管理し，クラスター同士でサイト間DBレプリケーションを実行する。
Example of site-to-site database replication function application to IoT data at each site

バイDBで受信したトランザクションログを，連続性，冪等（べきとう）性，信頼性などの厳格な検証を行った後に同期し，同時にクラスター内レプリケーションを実行するので，スタンバイDBの可用性も高められる。これにより，スタンバイDBでノード障害が発生しても，プライマリーDBとスタンバイDBは影響を受けずにサービスを継続できる。

⑷ 機能4：無停止ノード構成変更機能　プライマリーDBとスタンバイDBを停止させないで，各ノードの構成を変更できる機能を開発した。これにより，運用中に各サイトの負荷に応じて構成変更が可能となり，運用コストを大幅に削減できる。

図7は，サイトA，サイトBの設備データに対してサイト間DBレプリケーションを適用した例である。各サイトのクラスター及びDB構成は以下のとおりである。

サイトA：クラスター1（ノード4台構成，A設備DB，プライマリー），クラスター3（ノード3台構成，B設備DB，スタンバイ）

サイトB：クラスター2（ノード3台構成，A設備DB，スタンバイ），クラスター4（ノード4台構成，B設備DB，プライマリー）

サイトAとBは互いの設備データを参照できるため，複数サイトを横断した監視分析が可能となる。また，サイト内の個々のノードが停止した場合はクラスター内のレプリカを用いてサービスを秒単位に再開させ，サイト全体が停止した場合は別サイトのスタンバイDBを用いて分単位でサービスを再開できる。

6. あとがき

GridDBのクラスター内レプリケーション機能をクラスター間に拡張し，高い処理性能とサイト間のデータ一貫性を保証したサイト間DBレプリケーションを実現した。サイト障害が発生した場合の復旧と，サイトを横断した監視分析の両立を可能とし，企業のIoTデータの有効活用に大きく寄与できる。

今後は更に大規模，かつ，多数のサイト構成で管理するシナリオでの検証を進め，適用範囲を広げていく。

文　献

⑴ 服部雅一，ほか．ペタバイト級IoTデータを高速に処理するスケールアウト型データベース GridDB．東芝レビュー．2020，**75**，5，p.39–43．<https://www.global.toshiba/content/dam/toshiba/migration/corp/techReviewAssets/tech/review/2020/05/75_05pdf/f01.pdf>，（参照 2024-06-10）．

金輪　拓也　KANAWA Takuya
東芝デジタルソリューションズ(株)
ソフトウェアシステム技術開発センター
ソフトウェア開発部
Toshiba Digital Solutions Corp.

近藤　雄二　KONDO Yuji
東芝デジタルソリューションズ(株)
ソフトウェアシステム技術開発センター
ソフトウェア開発部
Toshiba Digital Solutions Corp.