ビッグデータの活用に最適なスケールアウト型 新データベース「InfoFrame Relational Store」

祐成 光樹・田村 稔

要 旨

ビッグデータを活用するに当たり、既存のリレーショナルデータベースはデータ量・アクセス数の増加に柔軟に対応（スケールアウト）できない課題がありました。一方、先進技術であるキーバリューストアはSQLを用いたデータアクセスや基幹業務で必要なトランザクション処理が利用できない課題がありました。「InfoFrame Relational Store（IRS）」は、（1）SQLインタフェース（2）トランザクション処理（3）基幹業務に適用できる高信頼性を備えた、ビッグデータの活用に最適なスケールアウト型データベースです。本稿ではIRSの特長及びアーキテクチャについて紹介します。

キーワード

●ビッグデータ ●キーバリューストア ●スケールアウト ●SQL ●トランザクション ●スモールスタート ●NewSQL ●基幹業務 ●メモリデータベース ●高信頼

1. まえがき

近年、クラウドサービスの拡大やスマートフォン、センサデバイスの急速な普及などを背景として、増え続けるユーザー情報、さまざまなセンサから収集されるデータ、SNS上に書き込まれるテキストデータなど、企業内外に流通するデータは爆発的に増加しています。このような社内外に溢れる膨大なデータ「ビッグデータ」の活用が、企業競争力の向上に不可欠となっています。

ビッグデータを活用するには、大量かつ増え続けるデータを蓄積するためのデータベースが必要です。従来のリレーショナルデータベース（Relational DataBase：RDB）でも大量のデータを複数のデータベースに分割して格納（パーティショニング）することは可能でした。しかし、RDBでは、アプリケーションがデータベース間の共有データを用いてデータ処理する必要があったため、データ量・アクセス数の増加に応じて運用中に新たなサーバを追加してデータベース規模を拡張する、つまりスケールアウトすることが困難でした。

一方、スケールアウトを実現するデータベースとしてキーバリューストア（Key Value Store：KVS）が注目されています。KVSは「データの索引となるキー」と「実データとなるバリュー」をペアとしてデータを管理する構造でスケールアウトを実現します。しかし、KVSはトランザクション処理を備えないことに加え、SQL（Structured Query Language）のような統一された標準インタフェースがないため、アプリケーション開発者は利用するKVS製品ごとに設計技術を習得する必要があります。

このたびNECでは、データ量・アクセス数の増加に応じた柔軟な拡張性と信頼性の高いシステムを容易に設計できる、ビッグデータの活用に最適なスケールアウト型データベースソフトウェア「InfoFrame Relational Store（IRS）」を開発し、提供を開始（2012年4月）しました。

2. ビッグデータの活用に最適なデータベース InfoFrame Relational Store

IRSは、既存のRDBのメリットと先進技術であるKVSのメリットを併せ持つ、ビッグデータ時代の新たなデータベースソフトウェアです（**表**）。コモディティサーバでシステム構築でき、サーバ台数を増やすことでデータ処理能力をリニアに向上できます。本章ではIRSの特長と効果について説明します。

表　InfoFrame Relational Storeの特長

	RDB	KVS	InfoFrame Relational Store
スケールアウト（データ量・アクセス数の増加に柔軟に対応）	×	○	◎
SQLインタフェース	◎	×	○
トランザクション処理	○	×	○
高信頼性	○	×	○

2.1 スケールアウト

IRSに蓄積するデータ量やIRSに対するアクセス数が増加した場合、新たにサーバを追加すればデータベース規模（蓄積可能なデータ量、収容可能なアクセス数）をリニアに拡張（スケールアウト）できます。IRSはスケールアウト中であってもアプリケーションから発行されるクエリを処理できるため、RDBのようなシステム拡張に伴う業務の運用停止が不要です。

RDBでは、将来に見込まれる需要（最大データ量・アクセス数）を予測し、需要予測値に対応できる規模のデータベースを初期導入する必要があるため、初期導入コストが莫大となります。一方、IRSは、現時点の需要に見合ったデータベースを構築すればよく、RDBと比較して初期導入コストを抑えたスモールスタートが可能です。

2.2 SQLインタフェース

IRSでは、SQLインタフェースを使って業務アプリケーションを開発できます。

RDB向けに開発してきたアプリケーション資産を流用できるとともに、慣れ親しんだSQLを使ってアプリケーションを開発できます。BIGLOBEの画像管理サービスにIRSを試験的に先行導入した結果、既存のアプリケーションプログラムの99%という高い流用性を実証しています（流用率はアプリケーションプログラム中で発行しているSQLの種類や数に依存します）。

2.3 トランザクション処理

KVSは、複製されたキーバリューデータが複数のサーバに分散配置されるアーキテクチャであるため、データ一貫性を保持してデータを更新するトランザクション処理が困難でした。IRSは、NEC北米研究所で開発した「マイクロシャーディング機構[1)]」を採用することにより、基幹業務で必須となるトランザクション処理を実現しています。

トランザクション処理をサポートすることで、データ一貫性を重視する業務であるためにKVSを利用できなかったオンライントランザクション系サービス（ECサイト、キャリア、金融など）への適用が可能となります。

2.4 高信頼性

トランザクションを処理するサーバ（トランザクションサーバ）及び実データを格納するサーバ（ストレージサーバ）では、データを複数のサーバに多重保存します。

トランザクションサーバでは、マスタサーバが管理するデータをバックアップサーバ（2台以上を推奨）に同期レプリケーションします。マスタサーバの障害を検出したバックアップサーバは、1秒以内にマスタサーバから業務を引き継いで復旧を完了できます。

3. InfoFrame Relational Storeのアーキテクチャ

IRSは、（1）アプリケーションからのデータアクセスを処理する「Partiqleサーバ（パーティクルサーバ）」、（2）ディスクアクセス回数を最小化してインメモリでトランザクションを処理する「トランザクションサーバ」、（3）実データ（キーバリューデータ）をディスクに格納することで永続化する「ストレージサーバ」で構成します（**図1**）。いずれのサーバもRed Hat Enterprise Linux 5系が動作するコモディティサーバを使用して構築でき、ストレージサーバもコモディティサーバの内蔵ディスクにキーバリューデータを格納します。

アプリケーションからのアクセス数の増加にはPartiqleサーバ、更新データ量の増加にはトランザクションサーバ、蓄積するデータ量の増加にはストレージサーバを追加することで、目的に応じたきめ細かなスケールアウトが可能です。また、いずれのサーバもアプリケーションを運用しながらスケール

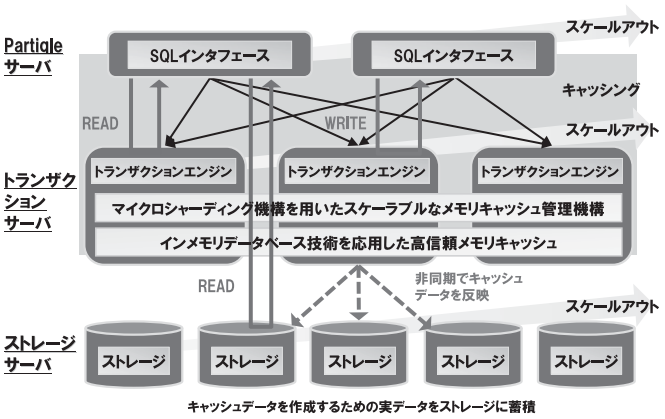

図1　InfoFrame Relational Storeのアーキテクチャ

アウトできるため、ミッションクリティカルな基幹業務への適用が可能です。

3.1 Partiqleサーバ

アプリケーションはIRS用JDBCドライバをロードすることで、Partiqleサーバが提供するSQLインタフェースを利用できます。アプリケーションは任意のPartiqleサーバに接続した後、SELECT、UPDATE、INSERTといったSQLをPartiqleサーバに発行します。Partiqleサーバはアプリケーションから受信したSQLを解釈して、キーバリューデータを操作するためのAPI（以下、KVS-API）からなる実行プランを自動作成します。トランザクションの実行プランの場合は、ストレージサーバからトランザクション内で更新・参照するデータをトランザクションサーバのメモリ上に取得し、取得したデータに対して更新・参照を行います。一方、トランザクションではない参照リクエストの実行プランの場合は、トランザクションサーバを中継することなく、ストレージサーバに蓄積するデータを直接参照します。

SQLインタフェースを利用できるので、アプリケーション開発者はKVS-APIを意識する必要はありません。しかし、Partiqleサーバが自動生成した実行プランは、従来のKVSと同様にキーを指定してキーバリューデータを操作します。このため、例えばSELECTなどの参照リクエストでは、キー（RDBにおけるインデックス）を使った大量データからの一本釣り（WHERE句にキーをイコール条件で指定する検索）には強いが、全件検索には向かないという性質があり、これらの性質を意識したワークロードの設計が必要となります。

3.2 トランザクションサーバ

マイクロシャーディング機構によってトランザクション処理を利用できます（**図2**）。トランザクションを受信したトランザクションサーバは、トランザクション区間で更新・参照するすべてのキーバリューデータをストレージサーバから収集します。トランザクションサーバはインメモリデータベース技術を応用しており、収集した複数のキーバリューデータをメモリキャッシュとしてマイクロシャードと呼ぶ単位のキーバリューデータに集約します。トランザクションサーバは、メモリ上のマイクロシャードに更新ログを書き込んだ時点でトランザクションを完了し、コミット成功をアプリケーションに返却します。

マイクロシャーディング機構を使ったトランザクション処理では、トランザクションが複数のサーバにまたがることなく1台のトランザクションサーバ上で完結するので、2フェーズコミットのような複雑なコミット処理を使わずに高効率なトランザクションを実現できます。また、トランザクションサーバのメモリ上のマイクロシャードに更新ログを追記するだけでトランザクションを完了するので、ストレージサーバへのディスクアクセス回数を最小化した高速なトランザクションを実現できます。

メモリ上のマイクロシャードは、複数のトランザクションサーバに同期レプリケーションされて多重保存されます。更に、マイクロシャードに集約された複数キーバリューデータの更新履歴は、一定周期でストレージサーバのディスクに非

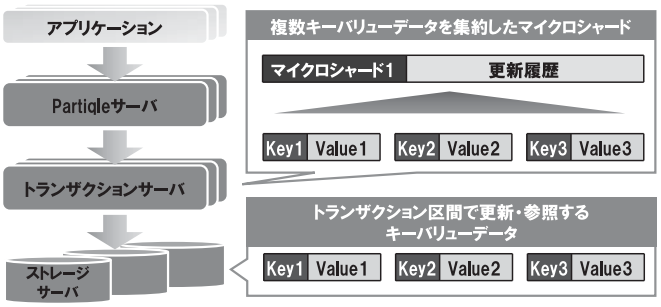

図2　マイクロシャーディング機構

同期で反映され永続化します。データの多重化及び永続化によって、トランザクションサーバ故障時もデータを確実に保護できます。

また、マスタサーバとして稼働するトランザクションサーバが故障した場合にも、バックアップのトランザクションサーバが、トランザクションサーバ間で定期交換するハートビートからその故障を検出し、マスタサーバに自動昇格することで業務を継続します。

他にもトランザクションサーバには、以下に示す技術的特長があります。

- **楽観的ロックによるトランザクション同時実行制御**
  RDBではトランザクション実行時に更新対象であるデータをロックして、他のトランザクションを排他処理する悲観的ロックが一般的です。一方、IRSは更新対象であるデータ（マイクロシャード）に埋め込まれたバージョン番号を事前に読み出し、データの書き込み完了時のバージョン番号と比較することで排他処理する楽観的ロックを採用しています。つまり、読み出し時のバージョン番号と書き込み時のバージョン番号が異なる場合は、同時並行していた他のトランザクションが先行してデータ更新したと判断し、コミット失敗となります。楽観的ロックではロック制御情報などの共有資源を使用しないためスケールアウトが容易になります。
- **トランザクションサーバ処理負荷の自動平準化**
  トランザクションサーバのメモリの使用状況を常時監視します。そして、トランザクションサーバごとの処理負荷が平準化されるように、トランザクション処理量を自動的に再配置します。これにより、スケールアウトを繰り返してサーバ台数が増大した場合でも、すべてのトランザクションサーバのリソースをバランスよく有効活用できます。

3.3 ストレージサーバ

ストレージサーバには、オープンソースのKVSであるVoldemort [2] を採用しています。トランザクションサーバから受信するキーバリューデータを、複数のストレージサーバに多重保存してデータを永続化します。

ストレージサーバを新規追加した場合、他のストレージサーバが蓄積するデータの一部を、新規追加したサーバに移動します。これにより、ストレージサーバ間のディスク使用量の偏りが平準化されます。また、データ移動中に移動元のサーバで参照リクエストを受信し、要求されたデータが存在しなかった場合は、データの移動先であるサーバを特定して、参照リクエストを移動先のサーバにルーティングすることで、業務を運用停止することなくストレージサーバをスケールアウトすることを可能とします。

4. むすび

InfoFrame Relational Storeは、（1）SQLを用いたデータアクセス、（2）マイクロシャーディング機構とインメモリデータベース技術によるトランザクション処理、（3）ミッションクリティカルな基幹業務に適用できる高信頼性を備えた、ビッグデータ活用へのハードルを大きく引き下げるスケールアウト型データベースソフトウェアです。

今後は、性能を発揮しづらい広範囲のデータを対象とした処理（バッチ処理など）に対応するため、Hadoop連携などを検討するとともに、SQLのサポート範囲を拡大する予定です。弊社では、お客様との共同検証などを通じて本製品の継続的な強化を行うとともに、ビッグデータの活用に向けた取り組みを推進します。

---

*Linuxは、Linus Torvalds氏の日本およびその他の国における登録商標または商標です。

*Red Hat 、Red Hat Enterprise Linuxは、Red Hat, Inc.の米国およびその他の国における商標です。

*Hadoopは、The Apache Software Foundationの登録商標または商標です。

参考文献

1) Junichi Tatemura, Oliver Po, Hakan Hacigumus: “Microsharding:A Declarative Approach to Support Elastic OLTP Workloads,” ACM SIGOPS Operating Systems Review, Vol. 46, Issue 1, 2012.1

2) Project Voldemort:A distributed database,（参照2012.7）http://project-voldemort.com/

執筆者プロフィール

祐成 光樹
ITソフトウェア事業本部
第三ITソフトウェア事業部
主任

田村 稔
ITソフトウェア事業本部
第三ITソフトウェア事業部
マネージャー

NEC 技報のご案内

NEC技報の論文をご覧いただきありがとうございます。
ご興味がありましたら、関連する他の論文もご一読ください。

NEC技報WEBサイトはこちら

NEC技報(日本語)　　NEC Technical Journal(英語)

Vol.65 No.2　ビッグデータ活用を支える基盤技術・ソリューション特集

ビッグデータ活用を支える基盤技術・ソリューション特集によせて
ビッグデータを価値に変えるNECのITインフラ

◇ 特集論文

データ管理 / 処理基盤

超高速データ分析プラットフォーム「InfoFrame DWH Appliance」
SDN技術で通信フローを制御する「UNIVERGE PFシリーズ」
大量データをリアルタイムに処理する「InfoFrame Table Access Method」
大量データを高速に処理する「InfoFrame DataBooster」
ビッグデータの活用に最適なスケールアウト型新データベース「InfoFrame Relational Store」
高い信頼性と拡張性を実現したExpress5800/スケーラブルHAサーバ
大規模データ処理に対するOSS Hadoopの活用
大容量・高信頼グリッドストレージiStorage HSシリーズ(HYDRAstor)

データ分析基盤

ファイルサーバのデータ整理・活用を支援する「Information Assessment System」
超大規模バイオメトリック認証システムとその実現
WebSAMの分析技術と応用例～インバリアント分析の特長と適用領域～

データ収集基盤

スマートな社会を実現するM2Mとビッグデータ
微小な振動を検知する超高感度振動センサ技術開発とその応用

ビッグデータ処理を支える先進技術

多次元範囲検索を可能とするキーバリューストア「MD-HBase」
高倍率・高精細を実現する事例ベースの学習型超解像方式
ビッグデータ活用のためのテキスト分析技術
ビッグデータ時代の最先端データマイニング
ジオタグ付きデータをクラウドでスケーラブルに処理するジオフェンシングシステム
柔軟性と高性能を備えたビッグデータ・ストリーム分析プラットフォーム「Blockmon」とその使用事例

◇ 普通論文

地デジTVを活用した「まちづくりコミュニティ形成支援システム」

◇ NEC Information

NEWS

スケールアウト型新データベース「InfoFrame Relational Store」が2つの賞を受賞

Vol.65 No.2
(2012年9月)

特集TOP