

ハイブリッドクラウドにおけるデータベース同期手法

細谷 柚子 (指導教員: 小口 正人)

1 はじめに

近年、パブリッククラウドやプライベートクラウドが普及しつつあり、その両者をシームレスに結合するハイブリッドクラウドが注目されつつある。パブリッククラウドでは、スケールアウト/スケールダウンできることによりコスト削減や技術面のリスク軽減に繋がる。一方で、社外のサービスを利用することによるセキュリティへの不安も残るために、パブリッククラウドを使いながらも、一部の個人情報や社外秘情報などは社内システムであるプライベートクラウドを利用するというように、ハイブリッドクラウドの導入により両方のメリットを得ることができる。

しかし、実社会においては、ハイブリッドクラウドの導入はあまり進んでいない。これは、2つのクラウドを併用するには、データの一貫性をとらなければならないという技術的な問題があるためである。データベースサーバは企業の基幹を構成しているため、クラウドで動作させるべき重要度の高いシステムである。従って、本研究ではハイブリッドクラウドでデータベースを同期させることに注目した。

LAN環境を前提としてデータベースサーバを同期する Pangea[1] という既存のミドルウェアがある。これをハイブリッドクラウドに適用し、TPC-W ベンチマークを用いて評価実験を行った。その評価をもとに、ハイブリッドクラウドにおいて有用な新しい同期方式を検討する。

2 Pangea

Pangea は、LAN環境を前提として複数のデータベースサーバ間でスナップショット分離を保証するデータベースレプリケーションミドルウェアである。クライアントからサーバ側に直接アクセスするのではなく、ミドルウェアを介してデータベースにアクセスする方式となっていることから、データベースを改造することなく、サーバを増やすことで性能を向上させることが可能である。Pangea では照会処理は1台のレプリカで、更新処理は全レプリカで実行される。クライアントからミドルウェアまでの処理を Global transaction、ミドルウェアからサーバまでの処理を Local transaction といい、全レプリカでの Local transaction がコミットされ次第、Global transaction のコミット処理が完了する。また、レプリカの中の1台を Leader、その他は Follower とされ、更新処理の場合は Leader に対して更新をした後に、他の Follower に対しても同様に処理を行う。本研究では Pangea をハイブリッドクラウド環境に適用し、適切な処理が行われるよう Pangea の修正を検討していく。

3 基礎実験

3.1 実験環境

Pangea の基本性能評価を行った。データベースサーバ用にマシン2台を用いた。どちらのマシンも表1に示すスペックである。データベースサーバには PostgreSQL9.2.6 を使用した。クライアントとデータベー

スサーバの間に Pangea を接続して同期を行う。Webサーバとアプリケーションサーバには Tomcat6.0.37 を用いた。性能評価は TPC-W ベンチマークを使用した。TPC-W は仮想的なブラウザ (以下 EB とする) が、データベースにトランザクションを発行する。TPC-W には3種類ワークロードがあるが、それらは表2に示すように照会処理と更新処理の割合が異なる。性能評価指標は、スループット (1秒あたりのWEB画面表示:(WIPS)) とレスポンス時間 (1画面データの転送時間:(秒)) とした。実験環境を図1に示す。

表1: マシンのスペック

OS	Linux 2.6.32
CPU	Intel(R) Xeon(R) CPU @ 1.60GHz
Memory	2GByte

表2: ワークロード

ワークロード	Read-only	Update
Browsing mix	95 %	5 %
Shopping mix	80 %	20 %
Ordering mix	50 %	50 %

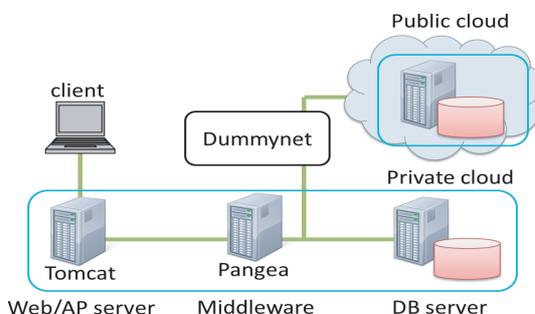


図1: 実験環境

3.2 基本性能評価

本節では、レプリケーションに最も負荷が大きい ordering mix での性能を測定した。ローカル環境におけるスループットとレスポンス時間 (秒) を図2に示す。上のグラフはスループット、下のグラフはレスポンス時間を表している。EB が290のとき最大スループットが38.9WIPSとなる。EB数を増やしていくにつれ、スループットも上がっていく。レスポンス時間は指数関数的に上昇していく。レスポンス時間が上昇していく、1より大きくなるあたりでオーバーロード状態になるため、EB数を増やしてもスループットは下降していく。



図2: ローカル環境における ordering mix の性能

4 応用実験

4.1 広域ネットワーク環境における性能評価

パブリッククラウドは遠隔地にあることを想定し、Dummysnet を使って遅延を入れた場合の ordering mix における測定を行った。近隣の街にバックアップを置く場合と、海外のような遠隔地にバックアップを置く場合を想定して RTT16ms と 256ms で評価した。ローカル環境との比較を図 3 に示す。

遅延を入れた場合においても、スループットとレスポンス時間の変化はローカル環境と同じく、EB 数を増やしていくにつれスループットも大きくなり、レスポンス時間は指数関数的に上昇する。ある点において最大のスループットとなり、その後は EB 数を増やしても下降していく。RTT16ms における最大スループットは EB 数が 290 のときで 38.8WIPS、RTT256ms における最大スループットは EB 数が 300 のときで 36.1WIPS である。RTT16ms の場合は、ローカル環境とスループット、レスポンス時間ともにほぼ差がなく、性能の低下はみられなかった。RTT256ms の場合は、ローカル環境と比較するとスループットが下回り、レスポンス時間がほとんどのケースで 1 秒を超えていることから、高遅延環境では性能低下が無視できない。

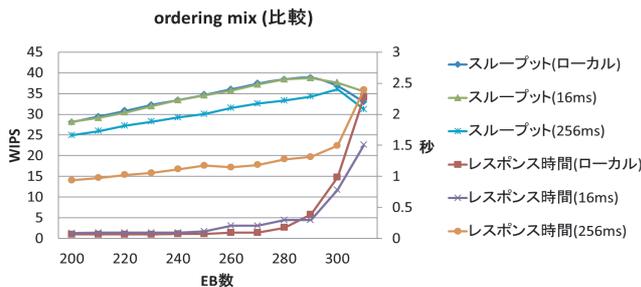


図 3: ordering mix の性能比較

4.2 Pangea の修正提案

広域ネットワーク環境での性能向上を図るために、Pangea の修正を検討する。修正前の Pangea の更新クエリのプロトコルを表 3 に示した。第 2 節でも示した通り、このプロトコルの場合、遠隔にあるデータベースからの応答を待つ時間がかかってしまうために、性能低下がみられると考える。そのため、全レプリカからの応答を待たず、Leader から応答が帰ってきた時点でクライアントに応答を返すように修正する。修正プロトコルを表 4 に示す。

表 3: Pangea の更新クエリプロトコル

1	クライアントから Pangea にリクエスト送信
2	Pangea から Leader にリクエスト送信
3	Leader から応答が返る
4	全 follower にリクエスト送信
5	全 follower から応答が返る
6	Pangea からクライアントに応答が返る

表 4: 更新クエリプロトコル修正

1	クライアントから Pangea にリクエスト送信
2	Pangea から Leader にリクエスト送信
3	Leader から応答が返る
4	Pangea からクライアントに応答が返る
5	全 follower にリクエスト送信
6	全 follower から応答が返る

修正プロトコルの場合、クライアントは応答を受け取る次のクエリを送るため、Follower に対するクエリが前のクエリを追い越して順番が前後になると、コンシステンシが崩れてしまうことが懸念される。しかし Pangea は、次のクエリが追い越さないように全 Follower からの応答を受け取ってから、次のクエリを受信するため、この問題は起こり得ないといえる。

4.3 実験結果

表 4 のように修正した Pangea での測定結果を図 4 に示す。遅延による影響が見られた ordering mix における RTT256ms での測定を行った。4.2 節で示した修正によりクライアントに早く応答が返るため、レスポンス時間が短くなると予想していたが、差がほとんどみられなかった。これは、クライアントに応答を即座に返しても Pangea が処理をするのは Follower からの応答を待ってからであるため、更新クエリの応答を早く返したとしても、レスポンス時間に関して改善がみられなかったためであると考えられる。

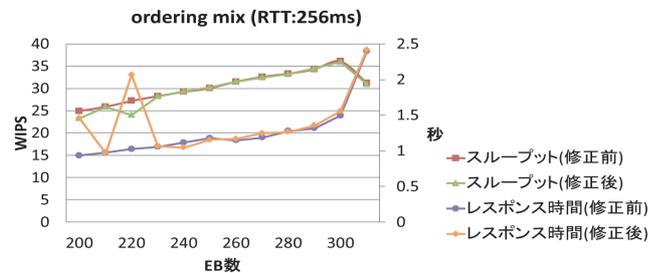


図 4: 修正後の ordering mix の性能

5 まとめと今後の課題

データベース同期システム Pangea についての基本性能評価を行った。ordering mix におけるローカル環境での最大スループットを確認した。通信遅延が生じる場合の測定も行い、ローカル環境と比較すると高遅延環境での性能低下が確認できた。これは、遅延が入ることで、距離がボトルネックになることが要因だと考えられる。また、性能向上をめざし、更新クエリに対しての修正を提案したが、本稿で提案した手法では性能の向上がみられなかったため、更なる検討を行う必要がある。

今後の課題としては、他の 2 種のワークロードでの測定と考察を行いたい。また、更新クエリの中でも更に細かいクエリの種別を考慮にいった Pangea の効率的なスケジューリングを検討したい。

参考文献

- [1] T.Mishima and H.Nakamura, "Pangea: An Eager Database Replication Middleware guaranteeing Snapshot Isolation without Modification of Database Servers", Proc.VLDB 2009, pp.1066-1077, August 2009.
- [2] 細谷柚子, 三島健, 小口正人: ハイブリッドクラウドにおけるデータベース同期方式の検討, DEIM2014, 2014 年 3 月発表予定