In-Memory DB の適用分野に関する性能評価検証
－ベンチマークテストによる Disk-Based DB との比較－

竹内 孝文† 吉田 忠城† 星 隆司†

†NTT サイバースペース研究所 〒238-0847 神奈川県横須賀市光の丘 1-1

E-mail: †{takeuchi.takafumi, yoshida.tadashiro , hoshi.takashi}@lab.ntt.co.jp

あらまし In-Memory DB は全てのデータをメモリ上に展開することで高速な検索性能を実現しており, これまで, 高速なレスポンスが要求される通信分野などに適用されてきた．最近，従来 Disk-Based DB が使われていた分野に In-Memory DB を適用する実用例が増えてきている．しかし，In-Memory DB のストレージ性能が，レスポンスやスループットに対して, 実際にどの程度寄与するのかといった評価は不十分であり, 曖昧な部分が多く残されている．

本稿では，複数のトランザクションパタンを用いて In-Memory DB と Disk-Based DB のレスポンス・スループット性能比較と共に，機能ブロック毎の処理時間コストを評価することにより，ストレージ性能の全体性能に占める影響やその他の性能要因に関して評価を行った．評価結果より，ショートトランザクションでは，ストレージ性能が大きく寄与するが，ロングトランザクションでは，ストレージ性能よりも中間演算処理性能やロック性能が全体性能に寄与することを確認した．

キーワード In-Memory DB，Disk-Based DB，ストレージ，中間演算処理，ロック性能，プロセス間通信

1. はじめに

近年, In-Memory DB が様々な分野で適用されつつある．In-Memory DB は，全てのデータをメモリ上に展開することにより，高速な検索性能が期待できることだけでなく，SQL92 への対応の向上や，Oracle[1]や SQLServer[2]といった従来から DBMS 市場でシェアを持つ主要な Disk-Based DB と連携するためのインタフェースが提供されてきたこともあり，様々なシステムでの適用が検討されている. In-Memory DB の主な適用先としては，従来では，検索に特化したシステムとしてのデータ・ウェアハウスや，大規模バッチ処理などの用途が主流であったが，現在では，オンライントレードを扱う証券システムのような，更新性能も重視される分野への適用例も見られるようになった．

一方で，従来の Disk-Based DB を単純に In-Memory DB に置き換えることで，レスポンスやスループットの向上が期待できるのか，という観点からの評価が曖昧なのではないかという問題がある. また, Disk-Based DB のストレージ部分をメモリに置き換えることにより, In-Memory DB と同様なレスポンス性能が期待できるのかという観点もあると思われる．

そこで本稿では，DB のストレージ部のレスポンス性能への影響を明確化することを目的とし，In-Memory/Disk-Based DB のレスポンス性能の評価検証を行った結果及び考察を報告する．

以下，2 章では，性能評価及び適用分野の評価を行うにあたり，評価方針の明確化を行う．3 章では，評価方法及び評価項目について述べる．4 章では，実施した評価検証の結果を示す．5 章では評価検証結果から，In-Memory DB の適用分野を考察し， ストレージ性能の応答性能全体への影響などについて述べる．6 章で本稿をまとめる．

2. 評価方針

2.1. 性能影響因子の抽出

レスポンス・スループット性能の評価を行うにあたり，まず性能に影響する要因となると思われる項目の抽出を行った．

1） **ストレージ性能**
データを実際に格納・抽出するストレージとしてのメモリとディスクの性能

2） **中間演算処理・ロック性能**
・SQL に基づく各種演算処理を実行する部分（実行処理部）の性能
・共有リソースのロック性能

3） **プロセス間通信性能**
アプリケーションと DB 間の接続方式（Socket 間通信/Library 接続）による性能

これらの項目の処理性能が, 処理時間コストとしてレスポンス時間のどの程度の割合を占めているのか調査

するために，機能ブロックを図 1 のように分割し，このブロック単位で処理時間コストを計測し，トランザクションパタン毎にどの機能ブロックが性能の支配時間となっているのかを考察する．

SQL 構文解析部(Parser 部)及び実行計画作成部(Planner 部)に関しては，ストレージ部に着目した DB 評価を行うにあたり，In-Memory DB/Disk-Based DB で同様な実行計画を使用する必要があるため，実行計画を，事前に作成して使用することにより，この機能ブロックの性能影響を排除し，今回の評価では性能影響因子としての考察を行わないこととした．

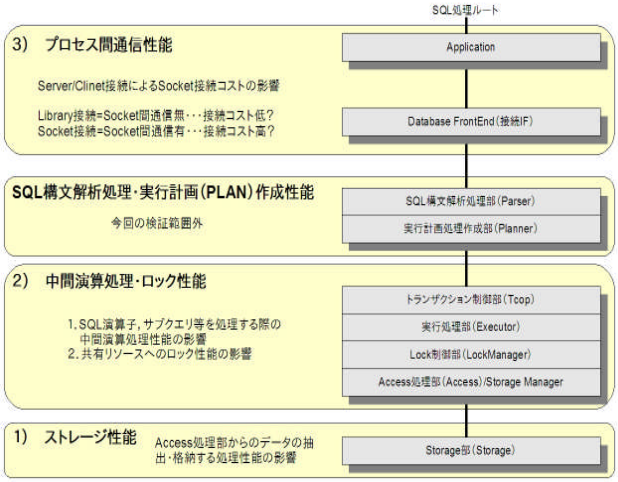

図 1．DB 機能ブロック概略図

2.2. 評価観点

評価を行う観点として，従来からの In-Memory DB の適用先であったレスポンス性能を要求されるショートトランザクションでのレスポンス性能という観点，及び従来から Disk-Based DB が支配的な市場を持つスループット性能を要求されるロングトランザクションでのレスポンス・スループット性能の観点から評価を行った．

1） **ショートトランザクション性能評価**

単純な SQL を使用し，DB 内で複雑な SQL 演算処理等の中間処理を行わないことで，ストレージの性能が支配的となると想定されるモデルでの，レスポンス性能評価．SQL は，インデックスを必ず使用する実行計画（PLAN）を使用する．

DB のレスポンス性能比較と同時に，高速なレスポンスの要求されるサービスに適用するための，DB の条件について考察する．

2） **ロングトランザクション性能評価**

ロングトランザクションを設定し，同時に複雑な SQL を使用し，DB 内で SQL 演算処理を行うことにより，中間演算処理や共有リソースのロックが発生する状態で，ストレージの性能による全体性能への影響を評価する．レスポンス性能及び，指定時間内で処理可能なトランザクション数の評価．

DB のレスポンス及びスループット性能比較と同時に，高スループット性能の要求されるサービスに適用するための，DB の条件について考察する．

図 2 にトランザクションモデルの概念図を示す．

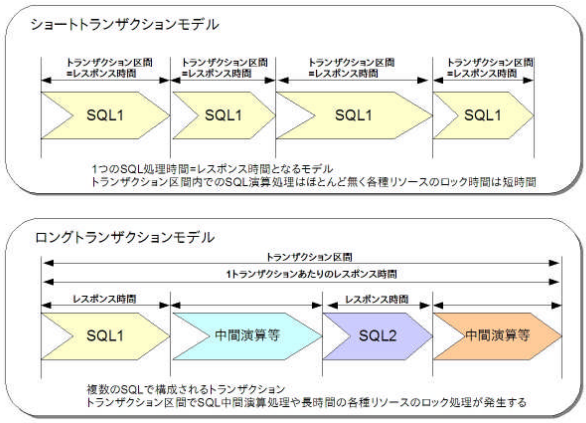

図 2. トランザクションモデル概念図

今回の評価検証では，ストレージの性能に着目しているため，DML である参照系・更新系に関しての性能評価を行う．

3. 評価方法及び評価項目

評価用の DB を選定する際に，レスポンス性能としての評価と同時に，ストレージ部の処理時間コストのような，機能ブロック毎の処理時間コストが計測できることが望ましいと考えた．このため，機能ブロック毎に所要時間を計測するために，ソースコードを修正可能である OSS（Open Source Software）を選定することとした．

In-Memory DB としては，OSS として適当なものが無かったため，独自開発された In-Memory DB を選定した．Disk-Based DB としては， OSS である PostgreSQL[3]を評価基準 DB として選定した．ベンチマークプログラムとの接続方法に関しては，接続インタフェースの違いにより，In-Memory DB ではライブラリ接続を行い，PostgreSQL では，ソケット接続で計測を行った．

3.1. 評価環境構成

評価環境として使用した評価機の構成は以下である.

OS：RedHat Enterprise Linux AS4(64bit)
Kernel： 2.6.9-5
CPU：Intel Xeon(TM) CPU 3.40GHz×2
Memory：1GByte×8
Disk：250Gbyte(RPM：7200 Cache：8MB S-ATA)
PostgreSQL：PostgreSQL-8.1.5

カーネルパラメータ等の変更に関しては，DB の性能に関する影響を考慮する必要があるため，共有メモリの最大値以外の変更は行わない.

3.2. ショートトランザクション性能評価

ショートトランザクションでのレスポンス応答性能を評価するにあたり，VoIP（Voice Over IP）等で適用例の多い，SIP（Session Initiation Protocol）による参照・更新をモデル化したベンチマークテスト用のアプリケーションを作成した．使用した DB 条件は以下のものである.

テーブル数：1
列数：11
列タイプ：固定長文字列×5，整数×6
列サイズ：728byte（固定長）
行数：100,000

計測モデルは，インデックスを使用する実行計画（PLAN）を使用し，参照系及び更新系共に，全ての列の SELECT 及び UPDATE を行なうモデルとし，1 タプルのみの参照，更新を行なった．取得するタプルが同一のブロック（ページ）に存在することによる共有メモリ上での同一のページでの折り返し処理を極力避けるために，WHERE 句で指定するインデックスキー値となる値はランダムに生成し，同一の SQL を 1 万回試行した上での平均値を採用した．以下に関して性能評価を実施した.

1）In-Memory DB/Disk-Based DB 性能比較

In-Memory DB と Disk-Based DB の 1 つの SQL 処理時間としての，レスポンス性能の評価を行う．性能差が大きい場合には，機能ブロック毎の処理時間コストの評価を行う．評価項目としては以下の分類を行った.

a）参照系性能評価
b）更新系性能評価

2）Disk-Based DB の共有バッファ性能評価

検索対象 Table の全データを共有バッファ上にキャッシュすることにより，ディスクへのアクセスを抑止することで，PostgreSQL の共有バッファでクエリ処理を折り返し，レスポンス性能向上の評価を行う．ストレージとしての，メモリとディスクの性能のみを評価したい意図があり，更新系ではディスクへの書込みを完全には排除出来ないため，評価項目としては以下の分類のみを行なった.

a）参照系性能評価

1）により，レスポンス性能の相対的な比較を行い，同時に機能ブロック毎の処理時間コストを測定することにより，ストレージ部での処理時間コストがレスポンス時間全体のどの程度の割合を占めているのかの評価を行う．2）により Disk-Based DB のキャッシュの性能を評価し，ストレージ部を，仮想的に単純にメモリに置き換えた場合に SIP 等で使用される場合のレスポンス要求性能を満たせるのかどうかの考察を行う.

3.3. ロングトランザクション性能評価

ロングトランザクションでのレスポンス及びスループット性能の測定では，ベンチマークソフトとして OSS 化されている OSDL-DBT2 を使用した. DBT2 は，商取引をシュミレーションするモデルであり，参照・更新系の比率を変更して計測することが可能である．使用した DB 条件は以下のものである.

テーブル数：9
列数：3～17
列タイプ：可変長文字列，整数，小数，時刻等
列サイズ：10～675byte（可変長列含む）
行数：100～1,000,000
ウェアハウス数：10
接続クライアント数：10
試験時間：30 分間

計測モデルは，参照系は DBT2 の READ-ONLY モデルである STOCK-LEVEL のみを使用し，更新系は全てのモデルをデフォルト比率で使用した．評価値としては，DBT2 自身の計測するレスポンス時間とスループット数を採用した. 以下に関して性能評価を実施した.

1）In-Memory DB/Disk-Based DB 応答時間比較

In-Memory DB と Disk-Based DB のレスポンス及びスループット性能の評価を行う．評価項目としては以下の分類を行った.

a）参照系性能評価
b）参照・更新系性能評価

2）Disk-Based DB の共有バッファ性能

PostgreSQL の持つ共有バッファ量を操作することにより，対象とするテーブルを全て共有バッファ上に配置した場合と，PostgreSQL の起動に最小限必要な共有バッファ量のみを確保した場合での，共有バッファによるストレージ部とのアクセス処理コストのスループット性能への影響に関して評価を行う．評価項目としては以下の分類を行った．

a）参照系性能評価
b）参照・更新系性能評価

4. 評価結果

各評価項目に基づく評価結果に関して以下に示す．

4.1. ショートトランザクション性能評価結果

ショートトランザクションでのレスポンス性能評価結果に関して以下に示す．

4.1.1. In-Memory DB/Disk-Based DB 性能比較

a）参照系性能評価結果

SIP モデルでのショートトランザクションでの参照系の性能に関しては，大きな性能差が認められた．この計測では，PostgreSQL の共有バッファでの折り返し処理を抑止するためのバッファ量の調整と，ディスクからバッファにページを読み上げる関数を呼び出していることの確認を行なっている．表 1 に計測結果に関して示す．

表 1. 参照系レスポンス時間

DBMS 種別	レスポンス時間（μsec）
PostgreSQL	200
In-MemoryDB	22

機能部毎の，処理時間コストを計測し，ストレージ部の処理時間コストとレスポンス時間全体との比較を行った．ここでのストレージ部の処理時間コストは，Access 処理部以下の経過時間としている．表 2 に計測結果に関して示す．

表 2. ストレージ部処理時間

DBMS 種別	ストレージ処理時間（μsec）	応答時間全体への占有率（%）
PostgreSQL	63	31.5
In-MemoryDB	20	90.9

表 2 より In-Memory DB は，ストレージからのデータの読み出し時間がレスポンス時間全体の支配時間となるが，Disk-Based DB として選定した PostgreSQL では必ずしも支配時間とはならない．ストレージからの読み出し時間に In-Memory DB と PostgreSQL にそれ程大きな性能差が見られないのは，ディスクキャッシュ，Linux のカーネルキャッシュの効果もあるものと思われる．表 2 のデータは，PostgreSQL の共有バッファを使わない構成でのデータである．共有バッファの性能に対する影響に関しては次節 4.1.2 に評価結果を示す．

計測結果より，PostgreSQL ではストレージ部の処理時間コスト以上に，中間演算処理コスト，プロセス間通信コスト等での処理時間コストが大きいことが確認された．機能ブロック毎の処理時間コストのプロファイルを行った結果を，図 3 に示す．

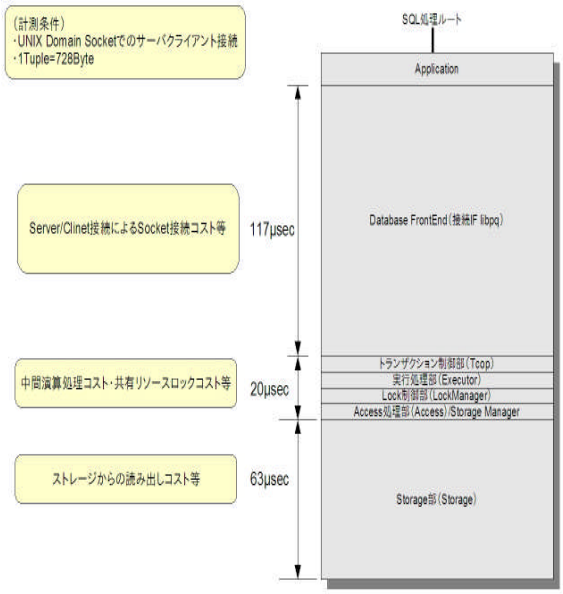

図 3. PostgreSQL 処理時間プロファイル

b）更新系性能評価結果

更新系の性能評価では，参照系程の性能差は見られなかった．表 3 に計測結果に関して示す．

表 3. 更新系レスポンス時間

DBMS 種別	レスポンス時間（μsec）
PostgreSQL	9,186
In-MemoryDB	5,002（69）※

※：（）内のデータは更新 Log の非同期書込時

参照系ほどの性能差が見られない理由としては，今回使用した，In-Memory DB は更新ログの書込み

をディスクに対して行っており，更新系の処理手順としてはそもそも DB としての大きな差分が無く，ディスクへの書込み時間が処理性能を支配することにあると思われる．更新系においては，実装アーキテクチャとして，非同期更新ログ書込み等を行うことで，大きな性能差分を生じさせることは可能であるが，アーキテクチャの違いによるものであり，ストレージ性能の評価という観点からは外れるため，ここでは詳細な評価を行わないこととした．

4.1.2. PostgreSQL の共有バッファ性能結果

PostgreSQL の共有バッファを使い，ストレージ部をメモリ化することによるレスポンス性能に関しての評価を行った．使用可能な共有バッファ値を調整し，キャッシュヒット率を 100%にした状態で，ディスクからバッファに読み上げる関数を呼び出していないことを確認した．参照系での計測結果を表 4 に示す．

表 4. 共有バッファ状態毎のレスポンス時間

共有バッファ状態	レスポンス時間（μ sec）
バッファ無	200
バッファ有	163

表 4 より，参照系ではバッファの効果による性能効果が認められる．

DB がデータを含んだページをディスクから共有バッファに読み上げる時間を Page Fetch Cost とすると，表 4 より 1 ページ当りの Page Fetch Cost は以下のように想定される．2 で除しているのは今回の場合には，2 ページを Page Fetch していたためである．

$$\text{Page Fetch Cost} = (200-163)/2$$
$$=18.5\ \mu\text{ sec}$$

また，PostgreSQL 内部で，実際にディスクからバッファにページを読み上げる関数の処理時間を計測し，同程度の計測結果を得た．なお，この値は当然ながら，評価機の各種デバイス等の性能，OS のカーネルキャッシュ[9]等の影響に依存する．

実装のアーキテクチャとして，参照に関しては，共有メモリとメモリストレージに大きな差は無いため，参照系の場合には，Disk-Based DB においても，ストレージをメモリへ置き換えることにより，1 ショートトランザクションあたり，18.5μ sec ～程度のレスポンス性能改善を見込むことは可能だと思われる．ただし，PostgreSQL では，全体の処理時間コストに対するストレージの比率は支配的では無いため，ストレージをメモリ化しても，In-Memory DB のようなレスポンス性能は期待出来ないものと思われる．

4.2. ロングトランザクション性能評価結果

ロングトランザクションでのレスポンス・スループット性能評価結果に関して以下に示す．

4.2.1. In-Memory DB/Disk-Based DB 性能比較

ロングトランザクションモデルで，ストレージ性能が全体性能に与える影響に関しての評価を行った．評価結果に関して以下に示す．

a）参照系性能評価結果

表 5 に計測結果に関して示す．ロングトランザクションでの参照系の性能に関しては，当初 In-Memory DB の方が高い性能を示すのではないかと想定していたが，逆に Disk-Based DB の方が高い性能を示した．

表 5. 参照系ロングトランザクション性能

DBMS 種別	レスポンス時間（sec）	スループット（数）※
PostgreSQL	0.003	89,703
In-MemoryDB	0.052	87,098

※：30 分間で処理を行ったスループット総数

In-Memory DB/Disk-Based DB のレスポンス時間が処理時間コストとして，どのように構成されているのか検証するために，トランザクションを構成する SQL から，実行計画（PLAN）の構成を分解し，ストレージ処理を伴う処理時間と，DB 内部での中間演算処理を伴う部分を比較した．図 4 に処理時間構成比率に関して示す．

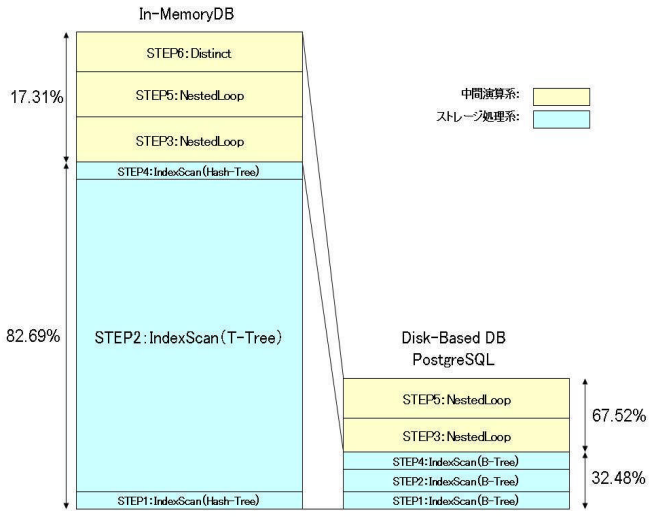

図 4. 処理時間比率プロファイル

PostgreSQL では，中間演算等を行なう内部処理時間がストレージ処理を伴うインデックススキャン時間よりも全体の処理時間の構成として支配的となる．今回使用した In-Memory DB では，T-Tree を使ったインデックススキャン時間が全体の処理時間として支配的である．T-Tree のスキャン時間が大きいのは，マルチカラムインデックスのスキャンキーの選定ロジックに問題があり，非効率なストレージアクセスを行なっているためであり，インデックス（アクセス処理部）の実装の問題である．ただし，T-Tree のスキャン時間を改善しても，ストレージに依存しない NestedLoop 等の内部処理時間が支配的となるため，ショートトランザクションの場合のように PostgreSQL に対して，10 倍程度の性能差を生み出すことは困難だと思われる．このため，DB 内部での中間処理等を伴うロングトランザクションでは，ストレージ性能の寄与が小さくなってしまう．スループット性能に大きな差分が見られないのは，ベンチマークソフトの DBT2 内で SQL を発行する際に，ms 単位の処理待ち時間が発生するため，DB での処理に余裕があるためだと思われる．

b）更新系性能評価結果

表 6 に計測結果に関して示す．In-Memory DB は，更新ログの設定を非同期に設定した．更新系の性能に関しても参照系と同じく，PostgreSQL が高い性能を示した．

表 6. 更新系ロングトランザクション性能

DBMS 種別	レスポンス時間（sec）	スループット（数）※
PostgreSQL	0.49	8,366
In-MemoryDB	1.25	8,088

※：30 分間で処理を行ったスループット総数

DBT2 のベンチマークモデルでは，更新 SQL のみが発行される訳ではなく，参照系と同じ SQL 及び副問い合わせ等と組み合わされた更新系 SQL が発行されるため，参照系の性能が大きく寄与してしまい，更新系に特徴的なものは見られなかった．ストレージへの書き込み時間そのものは，In-Memory DB では更新ログを非同期で書き込んでいることもありμsec の単位で高速であるが，トランザクション全体の処理時間が msec 単位となる場合に，その性能があまり寄与せず，DB での中間演算処理性能やロック性能に全体性能が支配されている．

4.2.2. Disk-Based DB の共有バッファ性能結果

PostgreSQL の共有バッファ量を変動させることにより，ロングトランザクションモデルでどのような性能差分が発生するのかに関して，十分に共有バッファを確保した場合と，PostgreSQL の起動に最小限必要な共有バッファ量のみを確保した場合での，評価を行った．評価結果に関して以下に示す．

a）参照系性能評価結果

表 7 に計測結果に関して示す．参照系でのバッファ効果に関しては，レスポンス性能に影響することが示された．

表 7. 参照系での共有バッファ状態毎の性能

共有バッファ状態	レスポンス時間（sec）	スループット（数）※
バッファ無	0.009	267,997
バッファ有	0.004	268,714

※　：30 分間で処理を行ったスループット総数

b）更新系性能評価結果

表 8 に計測結果に関して示す．更新系でのバッファ効果に関しては，レスポンス性能に殆ど影響しないことが示された．

表 8. 更新系での共有バッファ状態毎の性能

共有バッファ状態	レスポンス時間（sec）	スループット（数）※
バッファ無	0.809	21,517
バッファ有	0.793	21,579

※：30 分間で処理を行ったスループット総数

更新系では，PostgreSQL は同期ディスク書込みを行うため，レスポンス時間はディスク書き込み時間に支配される．参照系では共有バッファ効果があることから，実装アーキテクチャを，ディスクへの非同期書込みとすることで，性能向上を見込むことは可能だと思われる．

5. 適用分野に関しての考察

今回の評価結果より，In-Memory DB のストレージ性能の優位性の適用分野に関して考察する．

5.1. ショートトランザクション性能分野への適用

ショートトランザクション性能を要求される分野では，要求されるレスポンス時間が絶対的に小さいこともあり，参照系では，ストレージ処理時間コストが小さい In-Memory DB の優位

性が確認された．同時に，PostgreSQL では，ストレージ処理時間コスト以上に，プロセス間通信処理時間コストや，中間演算処理コストが大きいことが確認された．これは，DB として多機能なことによる，実装の複雑さも影響していると思われる．更新系では更新ログがディスク同期書込みの場合には，それ程大きな優位性は確認できなかった．同時に，更新ログをディスク非同期書込みに変更することで，大きな性能改善がみられるため，更新系の性能向上の方向としては，更新ログ処理のアーキテクチャに依存すると思われる．

PostgreSQL をこの分野に適用するためには，ストレージ性能以外にも次の処理コストの低減が必要であると思われる．

- 中間演算処理コスト
  実行処理部の SQL 中間演算コスト
  共有リソースロック処理コスト
- サーバ・クライアント間通信コスト
  Socket 接続によるプロセス間通信コスト

5.2. ロングトランザクション性能分野への適用

ロングトランザクション性能を要求される分野では，処理時間コスト全体に占めるストレージ処理時間コストが比較的小さく，DB 内部処理性能が大きく影響していることが，確認された．このため，単純に In-Memory DB を導入しても，意図した性能の改善が期待できるとは限らないと思われる．この分野への適用形態としては，ショートトランザクションでの優位性を生かして，Disk-Based DB でレスポンス性能の見込めない SQL の処理を分担するような，Disk-Based DB との連携形態での適用性があると思われる．

In-Memory DB をこの分野に適用するためには，次の処理コストの低減が必要であると思われる．

- 中間処理コスト
  実行処理部の SQL 中間演算コスト
  共有リソースロック処理コスト

6. まとめ

本稿では， In-Memory DB の適用分野に関して，ストレージ性能が DB 全体としての DML の性能にどのように影響するのかという観点から評価検証及び考察を行った．5 章で考察したように，応答性能に影響する機能部は SQL による中間演算処理コストやトランザクションパタンにも依存し，単純には決定できないため，Disk-Based DB を In-Memory DB に置き換えれば，常に性能改善が期待出来るわけではないことを確認した．

最近では，OS のバッファキャッシュ[9]や，ディスクの性能改善[5]もあり，ストレージとしての性能は，メモリと縮まりつつあるが，μsec 単位でのショートトランザクション応答性能が要求される場合には，ストレージとしてメモリを使用すべきであると思われる．同時に，プロセス間通信コストが許容可能なのかどうかを検討する必要がある．

また，msec 単位程度のロングトランザクション応答性能が要求される場合には，SQL により取得する行数にも依存するが，ストレージよりも，中間演算処理性能，ロック性能等を重視すべきであると思われる．

文　献

[1] Oracle, http://www.oracle.com

[2] SQLServer, http://www.microsoft.com/japan/sql/default.mspx

[3] PostgreSQL, http://www.postgresql.org

[4] Stefan Manegold, “Understanding, Modeling, and Improving Main-Memory Database Performance”, SIKS Dissertation Series No 2002-17.

[5] AN-I ANDY WANG,” The *Conquest* file system: Better performance through a disk/persistent-RAM hybrid design” ACM Transactions on Storage (TOS), Volume 2 , Issue 3 (August 2006), Pages: 309 - 348

[6] Mark Wong at el, “OSDL Database Test 2 (DBT-2),” http://www.osdl.org/lab_activities/kernel_testing/osdl_database_test_suite/osdl_dbt-2/

[7] 日本 OSS 推進フォーラム サーバー部会 技術評価 TF, “OSDL DBT-1 による PostgreSQL のチューニング”,

http://www.ipa.go.jp/software/open/forum/development/download/060529/DBT1_on_%20PostgreSQL_report.pdf

[8] Peter Wai Yee Wong, Ric Hendrickson, Haider Rizvi, Steve Pratt, “Performance evaluation of linux file systems for data warehousing workloads”, ACM International Conference Proceeding Series; Vol. 152

[9] Chuck Silvers, “UBC: An Efficient Unified I/O and Memory Caching Subsystem for NetBSD”, Freenix 2000 USENIX Annual Technical Conference Paper

[10] Juchang Lee, Kihong Kim, Sang K. Cha “Differential Logging: A Commutative and Associative Logging Scheme for Highly Parallel Main Memory Database”, 17th International Conference on Data Engineering (ICDE'01) p. 0173

[11] Hongjun Lu Yuet Yeung Ng Zengping Tian,” T-tree or B-tree: main memory database index structure revisited”, Database Conference, 2000. ADC 2000. Proceedings. 11th Australasian