

OMCSのPSAシステムモデル

相澤正俊[†] 富山卓二^{††} 齋川幸貴^{†††}

OMCS のシステム事例としてモバイルコンピューティング時代のビルディングシステムや株価情報の検索等のそれまでのメインフレーム (MF) やオープンシステムでは不可能であった数万件/秒以上のスーパーOLTP の基盤として PSA (Parallel Stream Architecture) を開発した。PSA はメモリ処理を基幹システムの OLTP として実現した先駆的なシステムモデルでもある。PSA システムモデルのアーキテクチャをそれまでの OLTP との比較で説明し、今後の重要な OLTP 基盤となることを本論文で述べている。

PSA System Model of OMCS

MASATOSHI AIWZAWA[†] TAKUJI TOMIYAMA^{††}
KOKI SAIKAWA^{†††}

This paper describes PSA (Parallel Stream Architecture) developed as a leading case of OMCS. PSA is adapted to billing system of mobile computing age or stock information searching, in which tens of thousand transactions capability is mandatory. Namely, PSA is a architecture for super OLTP that expand capability of Mainframe or previous Open System. PSA system model is a advanced system model which uses memory processing as the enterprise OLTP system

1. はじめに

都市銀行の勘定系システム、航空券予約システムにおいて、OLTP のトランザクション性能の最大は秒間数千件程度となっている。TP モニタと DB との組み合わせによる OLTP には性能的な限界点がある。クライアントからの要求を処理するために、A P サーバを多重化して並列実行するが、DB サーバへのアクセスが集中し DB がボトルネックとなる。DB のボトルネックを解消しようとする、システム構築のコストもかなり大きなものになる。

[†] 国際社会経済研究所 理事長 (前NEC副社長) Institute for International Socio-Economic Studies

^{††} NEC システムテクノロジー株式会社 社長 NEC System Technologies

^{†††} 日本電気株式会社 第二OMCS事業部 事業部長 NEC Corporation

ところが、携帯電話の料金計算をリアルタイムにて処理する場合や株価情報をリアルタイムに配信する場合には、数万件/秒という性能が必要とされる。従前の OLTP の処理能力を 1 桁以上上回る処理性能を実現するためには、通常の OLTP にてボトルネックとなる DB に依存しないような、新たなアーキテクチャによるスーパーOLTP が必要となった。インターネットのユビキタスコンピューティング時代にはこういったタイプの OLTP の増加が予想されるが、その先駆けとなるシステム事例である。

スーパーOLTP として処理すべきデータは、「イベント」として大量に発生し、データセンター側にリアルタイム処理すべきデータとして大量に集まる。1 問 1 答型のトランザクション処理とは異なり、クライアントから一方的に大量のイベントをセンタ側に送信するタイプ (所謂、集信型) のトランザクション処理である。スーパーOLTP が対象とする業務処理の領域は、図 1 に示す通りとなる。

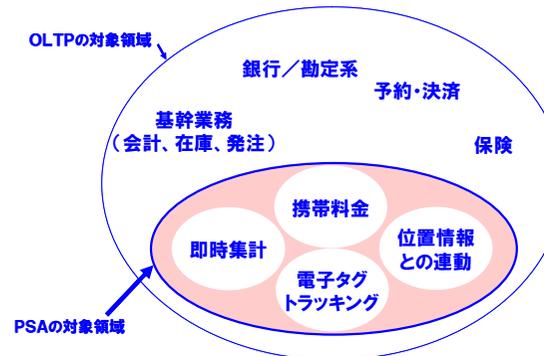


図 1 PSA (スーパーOLTP) の適用領域

2. PSA システムモデル

スーパーOLTP として処理すべきデータは、「イベント」としてリアルタイムに大量に発生し、データセンター側にリアルタイム処理すべきデータとして大量に集まる。その処理量は従来の OLTP の処理能力を 1 桁以上上回る秒間数万件～十数万件に達し、これを処理するためのスーパーOLTP が PSA (Parallel Stream Architecture) であり、それをシステムモデル化[1]したものが、PSA システムモデルである。以下、PSA システムモデルが対象とするシステム要件を述べ、具体的なアーキテクチャと特徴を述べる。

2.1 要件と課題

2.1.1 システム要件

対象とする業務モデルは、大量イベントを入力として、リアルタイムに高スループットなトランザクション処理を行うモデルである。例えば、携帯電話の料金計算を、

個々の通話の CDR(Call Data Record)がイベントとして発生する毎に、即時に料金計算を行うようなシステムを対象としている。従って、一般の OLTP に比べ、比較的処理が単純で、DB へのアクセスを局所化できる処理を対象としている。

大量イベントの発生によるスループットは数万件～十数万件/秒レベルである。性能以外の非機能要件では、システム構築のコストに制限のある商用システムを構築するために、以下に示すような経済性(コストパフォーマンス)、高可用性、拡張性などが必要となる。

- 性能: イベント処理性能 数万件/秒
- 信頼性・高可用性: ハードウェアやソフトウェア(OS、PP)の故障に対して、高速切替えを行い、データの紛失や重複したイベント送信や追い越し処理がないこと
- 経済性: 1 件のイベント処理当たりのオーバーヘッドを極小化し (1/10 以下に抑制)、相応のシステムコスト削減効果を生み出すこと
- 拡張性: トラフィックの伸びに対し、AP の修正なく、システムを拡張できること。

2.1.2 課題

秒間数万件というレベルのトランザクション処理を実現するためには、リアルタイム性を担保しつつ、オーバーヘッドを最小にできることが最も重要な要件である。従来のオンライントランザクションシステムを構成するアーキテクチャ(方式)では、下記のような課題がある。

(1) 1 件あたりのオーバーヘッドが大きい

- 従来の OLTP/DB による処理では、TP モニタによるメッセージ送受信、DBMS による I/O・ロック・バッファ管理、コミット処理といった様々な AP 処理以外のオーバーヘッドが大きい。

(2) データベースの構築限界

- トランザクション処理は同時に多数のスレッドまたはプロセスが並列実行し、それぞれのトランザクションから共有された DB にアクセスされる。このため、集中的な共有資源競合(索引、ログ、バッファなど)が発生し、DB の処理限界がシステム構成を決定づける。

上記を前提として、数万件/秒レベルの性能を出すためには、多数の AP サーバと多数の DB サーバによる実装が必要となるが、サーバ台数やソフトウェアのライセンス数が膨大となり、OLTP を用いた商用システムの構築は設備および費用という点で、現実的でない。

2.2 PSA のアーキテクチャ

2.2.1 狙い

バッチ処理では、大量のイベントを一括処理することで、コミット時の I/O や CPU 使用量を効率化でき、1 つのイベントのオーバーヘッドを極小化できる。一方、リアルタイムに処理するためには OLTP にて 1 イベントずつ処理を行うので、オーバーヘ

ッドは大きくなる。

商用のシステム規模を実現可能なレベルに抑えるため、低コスト性を重視し、かつ、リアルタイム性を兼ね備える必要があった。図 2 のように、バッチ処理の低コスト性と OLTP のリアルタイム性を併せ持つような処理特性となる。PSA では、複数イベントをまとめてコミットすることによって(エラー時には複数イベントをまとめてロールバックする)、この特性を実現している。

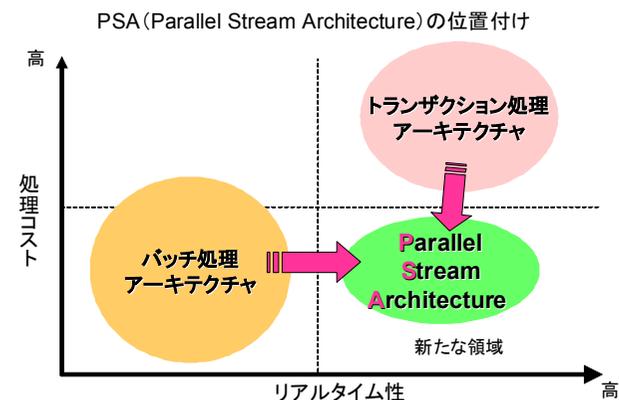


図 2 PSA の特性

2.2.2 PSA の構成

業務処理を多段の処理ステージに分割し、更に処理ステージの中を並列実行できるように分割し(分割した実行単位をストリームと定義)、メモリ DB を使用して高速な並列処理を実現する。このアーキテクチャを PSA と呼び、具体的な処理の流れの概要を図 3 に示す。

大量の入力イベントを、バッチのように全て一括処理するのではなく、また、従来の OLTP の様に 1 件ずつ個別に処理するのでもなく、リアルタイム性を損なわない範囲のイベント件数をまとめて処理する。この実行単位を 1 つのストリームとする。高スループット性を実現するために、複数のストリームにて大量の入力イベントを分担して並列実行することで実現する。1 つのストリームはイベントを処理した後に、後段のストリームに対してイベントを出力することができ、イベントの受け渡しにより、多段のストリームが連鎖的に動作を行う。

PSA を実現するために、超並列処理エンジン(ミドルウェア)を製品化した。

ストリームの実行制御、イベントスケジュールのためには、PSM^a(Parallel Stream

^a 製品の正式名称は、WebOTX Parallel Stream Monitor である

Monitor)を開発した。PSMはストリームへのイベントの配信、ストリーム上のコミット制御、後段のストリームへのイベント配信を行う。

ストリームはスレッド上で動作する(1スレッドに複数のストリームを定義可能な実装となっている)。PSMはイベントを入力すると、APにイベントを渡し、APはメモリDBのアクセスを行い業務処理を実行した後、後段ストリームの宛先を指定してイベント出力をPSMに依頼する。

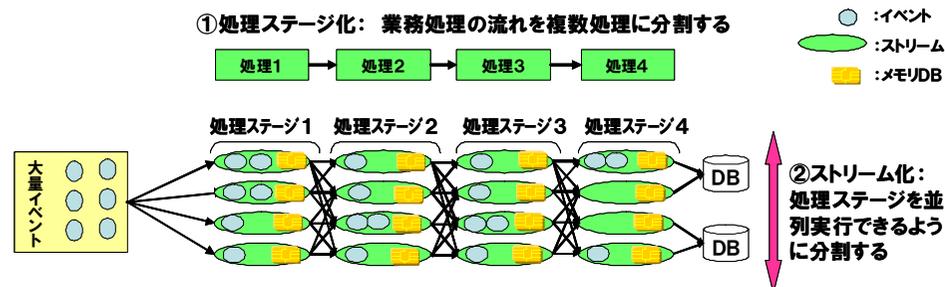


図3 PSAの処理の流れ

メモリDBとしてTAM^b(Table Access Method)を開発した。TAMはインデックスによる検索更新が可能であり、アクセス用の各種APIをもつ。メモリDBのメモリテーブルは、システム起動時、外部記憶装置から読み出され、システム終了時、外部記憶装置に書き出される。メモリテーブルの内容をコミットする際にシステム障害時の復旧用ログが採取されること以外、メモリテーブルへのアクセスを契機として外部記憶装置へのアクセスが発生することはない。

PSAの構成要素は図4に示す通りであり、具体的には以下の構成となる。

①ストリームとデータ分割

1つの処理ステージは、複数のストリームを多重実行させる。高スループットを実現させるためには、ストリーム間にて競合が発生しないことが必要であり、メモリDBの1つのメモリテーブルは、特定の1つのストリームからのアクセスしか許さないことを基本とする。複数のストリームから参照されるような共有テーブルは実装可能であるが、更新アクセスは特定のスレッドからのみ可能となっている。

各ストリームが並列実行できるように、各ストリームが扱う更新用データ(メモリDBのデータ)はストリーム間で重複しないようにデータ分割して割り当てる。データ分割の方法は、業務処理依存であり、メモリテーブルの主キーに対して、レンジ分割やハッシュやキー値の一部(下N桁)などを使って分割する。この分割によって、

^b 製品の正式名称は、InfoFrame Table Access Method である

各ストリームへのイベントの行き先が決定するため、ある特定のストリームにイベントが偏らないように、各ストリームへのイベント流入量が均等になるようなデータ分割を行う様、業務設計することが望ましい。この様なデータ分割を前提としている点がPSAの大きな特徴である。

②コンテナ転送機構

複数件のイベント情報はコンテナに格納され、ストリーム間で転送される。サーバ間をまたがる場合はネットワーク転送となるが、システム障害時等のコンテナの消失を防ぐために、メモリテーブルの内容をコミットする際に、外部記憶装置に格納する。

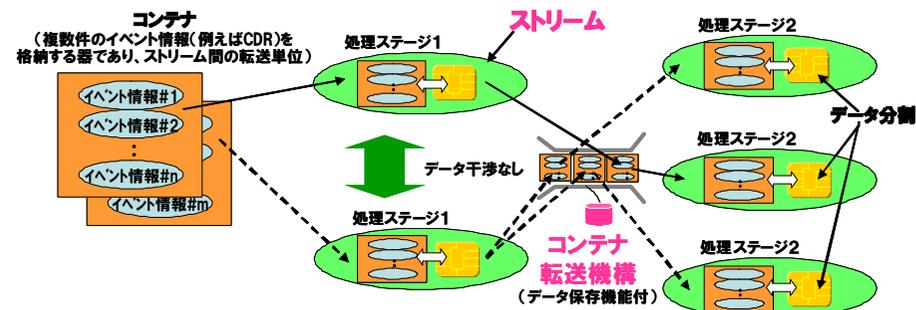


図4 PSAの構成要素

3. PSAの特徴

PSAの特徴について、OLTPとの比較もふまえて以下に述べる。

3.1 メモリテーブルによる検索・更新処理

OLTPではSQLによってDBアクセスを行うが、SQLによるCPUリソースの消費、DBサーバとの通信、DBのストレージへのIOといったオーバーヘッドが大きい。

OLTPを用いて多段の業務処理を行った場合、トランザクションを分割し、それぞれのトランザクションの中でDBアクセスを行うことになる。

数万件/秒という性能をOLTPで実現するためのシステム構成は、APサーバとDBサーバをスケールアウトし、かつ、多段構成でトランザクションを組む必要があるため、莫大なサーバ台数となり商用システムとして実現が困難である。

PSAでは、メモリテーブルアクセスを使用することで、DBアクセスのオーバーヘッドやDBサーバのボトルネックによる影響がなく、高速なデータ処理が可能となる。

最終的な処理結果をDBMSへストアする必要がある場合には、処理結果をイベントとしてデータストア用のサーバへPSMによって送り、リアルタイム処理とは非同期に結果をDBMSに書き込むことができる。この最終処理は非同期に行うことになるので、

この処理自身がネックになることはない。

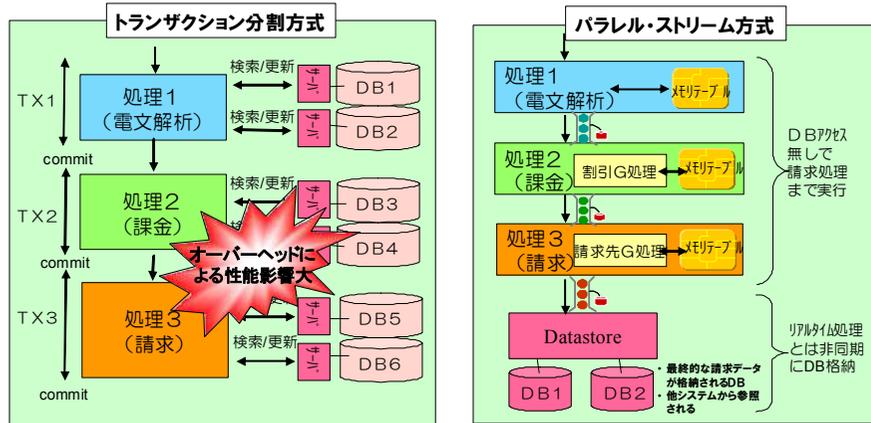


図5 DBとPSA上のメモリテーブルの比較

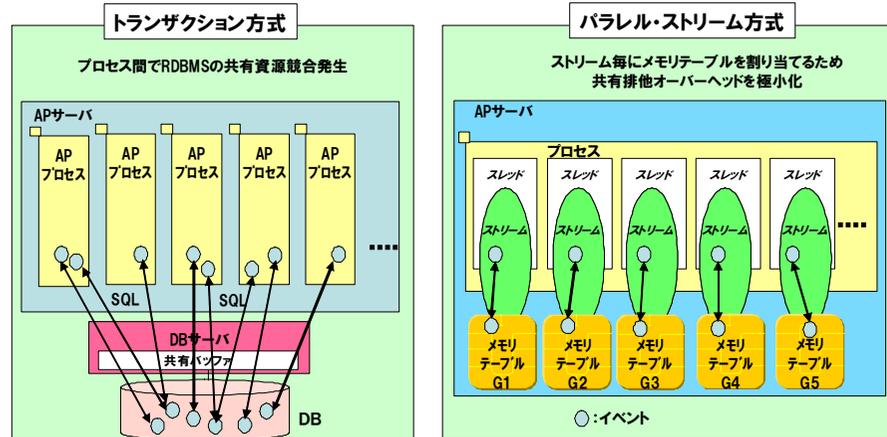


図6 OLTPのロック競合との比較

3.2 OLTPのロック競合との比較

トランザクションモニタ (マルチスレッド) とメモリ DB を組み合わせた OLTP では、スレッド間のロック競合が発生し、スループットに影響が出る。図6のように、APサーバ上のAPプロセスから発行したSQLは、DBサーバ上のDBMSの共有バッファやストレージに対して競合が発生するため、DBMS内でのロックが発生する。こ

のロックによる待ち合わせにより、APサーバ上のプロセスは wait 状態となり、サーバのCPUリソースを使い切ることが出来ず、期待されたスループットを出すことができない。

PSAではメモリDBのメモリテーブルをストリーム毎に括り付け、ストリームは1つのスレッドに割り当てられるため、複数のスレッドからのテーブルやレコードへのロック競合が発生しない。このため、高スループットな性能を出すことが可能となる。

3.3 高拡張性

スループットの急増により、CPUリソースが不足した場合の拡張性について、OLTPとPSAの比較を行う。

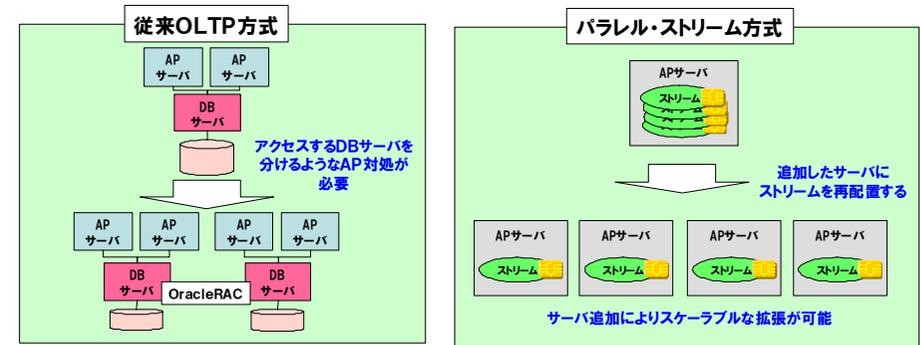


図7 拡張性

OLTPでは、スループット増加となるとDB性能がボトルネックとなる。DBサーバのスケールアップやスケールアウトにより対応することになる。スケールアウトではOracleのRACのようにDBサーバ台数を増やして対応するが、増設できるDBサーバ台数には限界がある。また、APのDBアクセスの仕方もキャッシュフュージョンが発生しないようにアクセス先のDBサーバを分けるようなAPロジックを組み込む必要がある。

PSAにてスループット増加となったときには、サーバ追加を行いストリームの再配置をPSMに指示してストリームを別のサーバに移動させる事により、スケラブルに処理能力を拡張することができる。

c Real Application Clusters

d 2001年リリースのOracle 9iで最大32。2003年の10gで63、2007年の11gで100。

e 複数のサーバ間で同一のデータがアクセスされた場合に、サーバ間でキャッシュを通信し合い、データの整合性を守る仕組み。通常通信のオーバーヘッドがかかるので、性能劣化の原因となることがある

f PSMに対するコマンド入力により指示する

3.4 高可用性

トランザクションのしかかり中に（1コミット単位となる複数のイベントを処理中に）サーバダウンが発生した場合、サーバ切替え後に、処理途中であった入力コンテナの復旧や出力途中のコンテナの削除を行い、旧現用系で出力されていたログを用いて、待機系（新現用系）でメモリテーブルが復旧され、再度、入力コンテナからイベントデータを読み込んで処理を再開する（図8参照）。これらの処理は、PSMとTAMが連携して行うので、イベントデータの紛失、重複したイベント処理や重複したイベント送信は行われない。また、PSMではコンテナの中のイベントの処理順序は変わらないため、イベント処理の追い越しも発生しない。

この方式の場合には、待機系（新現用系）でのメモリテーブルの復旧は、系の切替え時に行われるので、系の切替えが発生するまで待機系で事前に処理を行っておく必要がなく、例えば現用系4台に対し待機系2台といった柔軟な構成を取ることが可能であり、システム構築コストを抑えることができるが、メモリテーブルの復旧には時間がかかる。

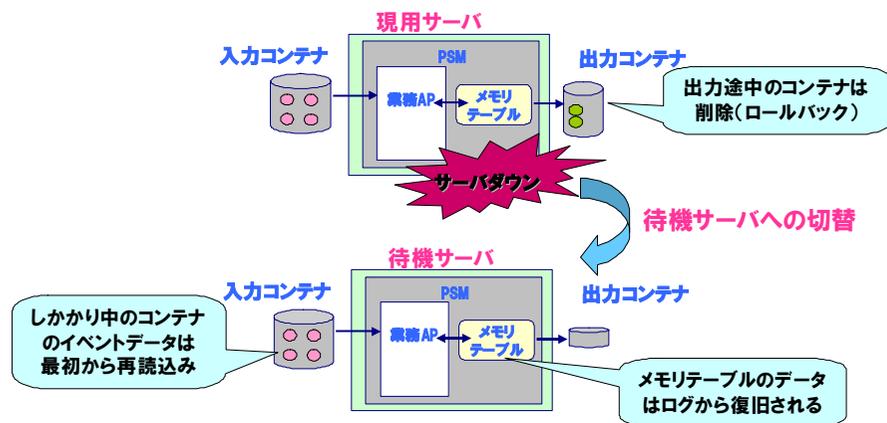


図8 サーバ切替え動作

メモリテーブルの復旧の高速化を目的として、現用サーバのメモリテーブルのデータを待機サーバにレプリケーションしておくオプションも用意されている（図9参照）。レプリケーションは通信によって行い、複数のLANを束ねて送信することが可能である。LAN障害についても考慮しており、1つのLANに障害が発生しても残りのLANにより通信を行い、レプリケーションを継続することができる。

この方式の場合には、待機系に予めメモリテーブルのデータを送っておくことになるので、1つの現用系に対して必ず1つの待機系を必要とする。このため、メモリテ

ーブルの復旧の時間は短縮されるが、システム構築コストが高くなることには注意が必要である。

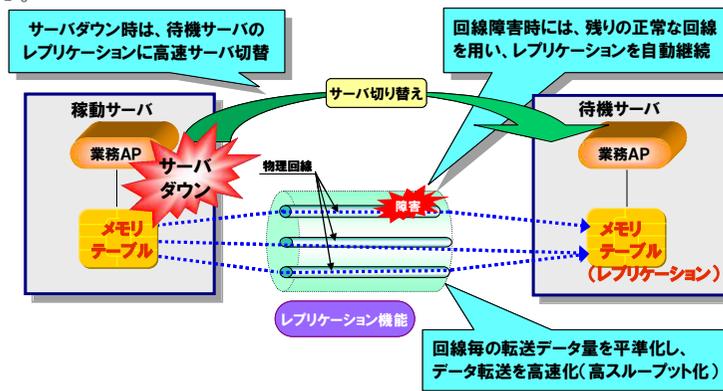


図9 高速サーバ切替

4. 性能評価

4.1 メモリテーブル性能

TAMはDBMSの15倍スループットが良く、大幅な性能差となった。DBMSへのアクセスのオーバーヘッドが非常に大きいことがわかる。（図10）

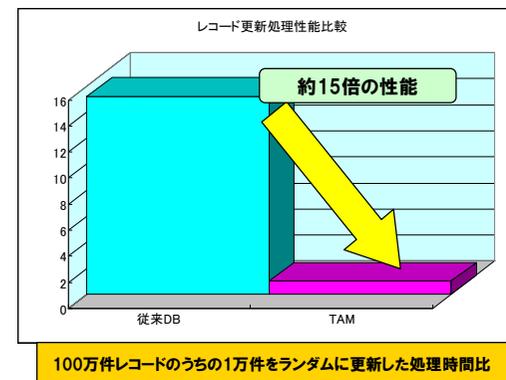


図10 TAMの高速性能

g システム傷害に巻き込まれた高々数件のレプリケーションをログから復旧する程度。これは通常数秒であるが、実際のシステム切替えには、他の処理も含めて20～30秒程度かかる。

TAM の性能に関しては、共有リソースへの排他制御がないため、実行多重度を上げてもスケラブルに性能が向上することを確認できた。実測では、CPU 数を 2 倍に増加したときの TAM へのアクセス回数は、1.99 倍に増加した。(図 11)

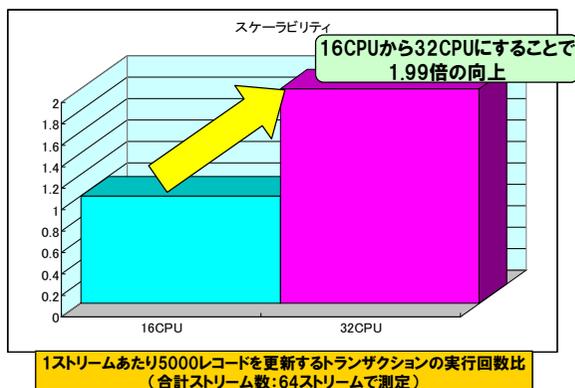


図 1 1 TAM のスケラビリティ

4.2 トランザクションオーバーヘッドの削減効果

PSM の性能に関しては、複数のイベントの処理を一括で処理するため、コミット処理やデータ入力のオーバーヘッドを小さくすることができる。実測では、1 トランザクションデータあたりのコミットのオーバーヘッドを 1/10 まで低減することを確認した。

この測定では、従来 OLTP と PSM とのミドルウェア性能の比較であり、OLTP と PSM 上での業務 AP 処理の差分 (SQL vs メモリアクセス) は含まれていない。

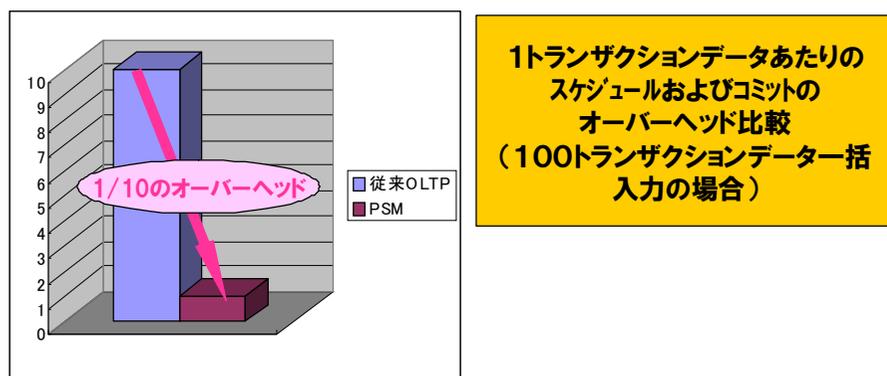


図 1 2 PSM によるオーバーヘッド削減

5. PSA で実現したシステム事例

5.1 ドコモ MoBills

PSA はドコモの料金システム (MoBills) をターゲットとして、2003 年 12 月から検討を開始し、2007 年に世界初のリアルタイム課金を実現した。携帯電話の通信明細データ (CDR) を受信して、その CDR をもとに通話料や割引額、請求額をリアルタイムに計算する。CDR は 10 億呼/日、ピーク性能は 5 万件/S となった[2]

ドコモ料金システムは、5000 万人を超える利用者の料金処理を行う、社会的に重要な大規模システムであり、PSA 上に大規模な AP が開発された。このため、OMCS^h[3] としてのシステム要件が求められた。すなわち、メモリ処理による高性能に加えて、サーバ故障時にデータの紛失や重複した処理がなく、かつ、24H365日連続運転するための高可用性・高運用性、負荷の増大に柔軟に対応できる高拡張性が求められた。

PSA適用事例：リアルタイム料金システム

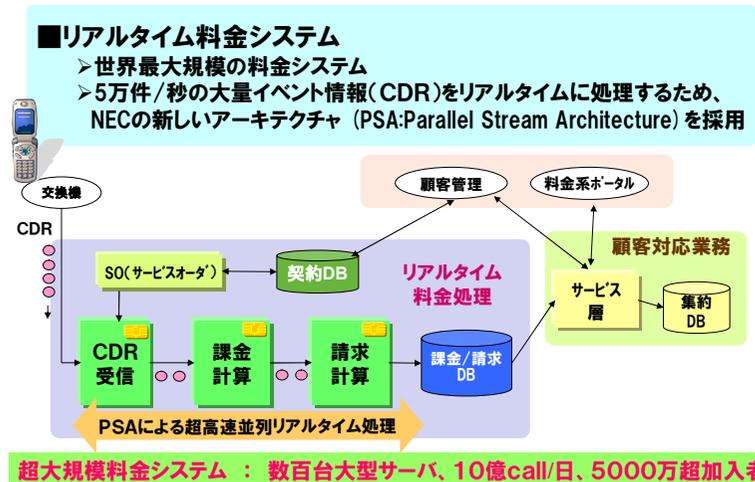


図 1 3 ドコモ様の料金システム事例

5.2 QUICK

TAM は、国内最大の金融情報提供会社のシステムである QUICK の次世代情報配信

^h Open Mission Critical System

基盤に適用された。2011年6月から商用稼働を開始している。[4]

この配信基盤は、東証などから株式に関する大量データをリアルタイムに高速処理する。東証からの情報量増大に対応するため、システムアーキテクチャを刷新し、TAMを適用した。大量の情報を受信して、それぞれを数十台のサーバへ斉配信する処理をミリ秒オーダーで実現した。

低コストで OMCS を実現するために、Linux ベースに OMCS 技術（高信頼サーバ、Enterprise Linux with Dependable Support）を採用している。

PSA適用事例： 株価配信システム

株式の時価情報をリアルタイムに配信するシステム
大量かつ高速化(低レイテンシー:msオーダー)を実現するため、
PSA(TAM)を使用した処理方式を適用、2011年6月サービスインし稼働中

技術的ブレークスルー： 魅力的な価格で、超高性能を提供するため

- ① 新たなPFへの製品対応 ⇒ TAMのLinux対応
- ② 高信頼で安価なオープン製品 ⇒ Express5800/スケラブルHAサーバ、Enterprise Linux with Dependable Supportの採用
- ③ 高速通信 ⇒ 10ギガLAN(マルチキャスト)

・マルチキャスト:ネットワークで、決められた複数のノードに対して、同時にパケット送信する方法。
cf.ユニキャスト:決められた1つのノードに対して、パケットを送信する方法
ブロードキャスト:不特定多数に対して、パケットを送信する方法

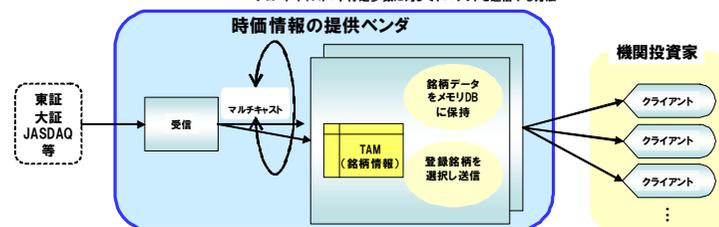


図14 QUICK様の株価高速配信の事例

6. PSAの新たな応用領域

ブロードバンドによる通信能力の大幅な向上とモバイルの進展による多数のクライアントの出現により、ネットワークを介したサービスがあらゆるビジネスで拡大している。Market to Marketを含むサービス拡大に伴い、今後、大量イベント処理へのPSAの適用が拡大すると予想しており、以下のような客先提案が行われている。

6.1 位置情報

GPSによる位置情報と連携し、新たなサービスを提供することが可能となる。

携帯電話や自動車などからGPSを用いた位置情報を大量のイベントとして入力し、利用者の趣味指向と提供情報（レストランやドライブインなどの近くにある店舗情報、渋滞情報など）とのマッチングを行い、タイムリーに利用者に情報を通知する。大量イベントの処理にはPSMを適用し、位置情報とのマッチングにはTAMによって高速化を行うことができる。

6.2 RFID (Radio Frequency Identification)

RFIDは製造業などで部品や製品に付与され、サプライチェーンにより各種企業間を製品が流通する状況をトレースするために用いられる。RFIDのトレース情報は莫大な量となるので、PSAによる大量イベント処理の効果を期待できる。

広義のRFIDの一種としては、非接触ICカードがある。Suicaなどの乗車カード、Edyなどの電子マネーがあり、利用者と購買行動の情報はマーケティングに活用することができる。

6.3 スマートメーター

欧米や中国のみならず、日本でも震災後の停電の影響もあり、スマートメーターへのニーズが高まっている[5][6]

スマートメーターから電力消費量をリアルタイムに送信して、個別の電力消費量から使用時間帯による割引計算などの課金処理、各地域の電力消費量を計算し電力供給量の監視や制御、法人に関しては特定な法人に属するスマートメーターからの消費電力総和を求め法人への請求明細の作成や法人としての電力使用量の上限監視などを行うシステムが今後のスマートグリッドに求められる。

7. 世のメモリ処理におけるPSAの意義

PSAはスーパーOLTPの開発が着想ポイントであるが、近年のメモリ処理はDWHのリアルタイム処理など大量のDBの並列処理でのメモリ処理の研究が行われている。

2003年にスタンフォード大学にて発表されたSTREAM[7]は、リアルタイムな大量データに対して検索の問い合わせを行うもので、その後、リアルタイムな大量データをメモリ上で行う処理について様々なタイプの研究活動や製品化が行われた[8][9]。

PSAがメモリ処理にもたらした意義を整理すると、以下の通りである。

- ① 大規模メモリ処理の実現
当時(2003年)のサーバは、512GBのメモリを搭載可能であり、この大容量メモリを使用して実用に耐えうるメモリDB(TAM)を実現した。
- ② 新しいOLTP(スーパーOLTP)
スーパーOLTPとして扱う大量イベントをリアルタイムに処理するための新たなアーキテクチャの幕開けとなった。
- ③ 世界最大級の実績

世界最大規模の料金システム（ドコモ）、大規模な株価配信システム（QUICK）の実績により、OMCSが必要とされる大規模システム領域に、スーパーOLTPとしてPSAを適用することができた [2] [4]。

8. まとめ

ユビキタス時代において必要とされる、それまでのOLTPモニタ、DBシステムによるOLTPに比べ1～2桁以上高性能なスーパーOLTPを、メモリ処理を駆使して実現したPSAのアーキテクチャの具体的構成と特徴を述べた。

また、PSAのシステムモデルは世界最先端のモバイルビルディングシステム等に適用し、その実効性が確認され、世界的にも見てもこの様な大規模システムで実現された例は少ないと考える。

今後ユビキタスコンピューティングの時代には、更に重要性が増すと考え、今後の新たな応用例（GPS、RFID、スマートメーター）にも触れた。OMCSの適用範囲も既存のOLTPの世界からPSAを活用した新しいOLTP（スーパーOLTP）への発展が期待され、PSAの重要性は増すものと考えている。

謝辞

OMCSのシステム開発においてPSAシステムモデルの必要性をご教示戴いたNTTドコモの皆様とPSA上のAP開発に取り組まれたNTTデータの皆様の絶大なご支援がなければPSAシステムモデルの開発は不可能であったので、深く感謝申し上げます。また、PSAの具体的開発とそれを適用したシステムの開発に参画された皆様にも重ねて感謝申し上げます。

参考文献

- 1) 相澤正俊他：「OMCSのシステムモデルによるプラットフォーム構築」
- 2) 宍戸周夫：「経営の見える化を追求--NTTドコモが取り組むリアルタイムマネジメントシステムの実態」、ZD Net Japan、http://japan.zdnet.com/cio/sp_09oow/20392427/ 2009年04月27日
- 3) 相澤正俊他：「OMCS構築技術の研究」
- 4) ソフトバンクビジネス+IT：「金融情報ベンダーQUICK、次世代情報配信基盤にNECを採用」 <http://www.sbbit.jp/article/cont1/23723> 2011年08月19日
- 5) 日本経済新聞：「スマートメーター、年2億台の大市場へ」、
<http://www.nikkei.com/tech/ecology/article/> 2011/6/6
- 6) 宮内直人：「スマートグリッド監視・管理システム」、情報処理学会報告IPJS SIG Technical
- 7) A. Arasu, et al.: STREAM: The Stanford Stream Data Manager, IEEE Data Engineering Bulletin, 26(1)
- 8) 森 有一 他：「情報爆発時代の到来に向けた大量高速データ処理技術への取り組み」、日

立評論、2008.07

9) 松浦紘也、鈴木豊太郎：「データストリーム処理とバッチ処理における動的負荷分散」 電子情報通信学会技術研究報告. DE, データ工学 110(107), 69-74, 2010-06-21

10) 富山卓二：「次世代ネットワーク時代におけるバックエンドシステムのアーキテクチャ」N E C 技報 Vol.59 No.2 P.45-48

商標について

OMCS、BankingWeb、PSA、ECOCENTERは、NECの日本国内における登録商標です。OpenDIOSA、PARALLELSTREAMMONITOR、SIGMABLADE、InfoFrame、WebOTXは、NECの日本国内およびその他の国における登録商標です。

Windows、SQL Serverは、米国Microsoft Corporationの米国およびその他の国における登録商標です。UNIXは、X/Open Company Ltd.がライセンスしている米国ならびに他の国における登録商標です。HP-UXは、米国およびその他諸国におけるHewlett-Packard Companyの登録商標です。Oracle、Java、WebLogic、TUXEDOはOracle Corporationおよびその関連企業の登録商標です。SAPはドイツ及びその他の国におけるSAP AGの商標または登録商標です。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。その他の名称は、それぞれの所有者の商標または登録商標です。