

分析向けプラットフォームにおける高信頼 ETL 方式の提案 Reliable ETL Method for Analytics Platforms

船矢 祐介[†] 谷川 桂子[†] 愛甲 和秀[†] 上原 敬太郎[†] 永見 明久[‡]
Yusuke Funaya Keiko Tanigawa Kazuhide Aikoh Keitaro Uehara Akihisa Nagami

1. はじめに

近年、IT を活用して顧客との関係を強化し、適切なサービスやコンテンツを提供する、クラウドネイティブな新興企業が出現してきている。そのコアとなっている、顧客ニーズを分析して適切なサービスにつなげる分析システムは Systems of Engagement (SoE) と呼ばれている。分析システムは彼らのビジネスの価値の源泉であり、多くの場合彼らはシステムを自ら構築し管理している。

一方、今までの基幹システムはビジネスで起こる事象の記録に主眼が置かれており、SoE に対して Systems of Record (SoR) と呼ばれている。従来型の現業を持つ企業の多くはこの基幹システムを保持している。しかし近年、このような現業を持つ企業が、分析システムを持つ新興の企業によりビジネスを失う場合がでてきた。そこで、彼らはビジネスを変革し自分たちのビジネスを守るために、分析システムを導入し始めている。

分析システムの特長は、新しいサービスの迅速な立ち上げと、ビジネスの成長と合わせたスケールができることである。そしてこれを実現するのが、例えば Dev/Ops といった文化/開発スタイル、マイクロサービスといったアーキ、クラウド的なリソース利用方法、データの分析、などである。特に、前述のクラウドネイティブな新興企業のシステムを見ると、適切なサービスやコンテンツを適切なユーザに提供するためには、分析が非常に重要である。

このような技術、特に分析、を支えるインフラストラクチャの要件としては、スモールスタート、及びスケールアウトが可能であること、分析のトライアンドエラーが容易に行えることが挙げられる。

現業を持つ企業では、分析の対象となるデータの多くは基幹システム内に蓄積されたデータに含まれているため、基幹システムから分析システムにデータ収集 (Extract/Transform/Load: ETL) する必要がある。しかし、その際基幹システムのデータを誤って破壊しまう可能性がある。さらに、ETL による基幹システムの性能低下は避けられない。これらの問題を防ぐためには管理者との煩雑な調整やシステムの再設計が必要となる。本稿では、この手間を低減しつつ基幹システムのデータ破壊と性能低下を防ぐ高信頼 ETL 方式を提案する。

2. 既存の ETL における課題

分析によって得られる価値がわからない段階で、分析のトライアンドエラーが必要な状況では、コストを抑えることと、本業への影響を抑えながら分析を行えることが望ましい。そのような場合は、業務 DB から必要なデータを抽出して、分析向けにデータウェアハウス(DWH)[1]を設計することとなる。

DWH を利用して分析を実施するために、基幹業務システムが動作しているデータベース(DB)に直接 ETL を実行する場合のシステムを図 1 に示す。ただし、以降はこのシステムを直接 ETL 方式と呼ぶ。左側が基幹業務システムの例であり、右側が分析システムの例である。基幹業務システムには分析対象のデータが格納されている。また、基幹業務管理者によって管理されており、業務へ影響を与えるような操作にはこの管理者との協議が必要となる。分析ユーザは、分析に必要なクエリを投げ、分析システムの ETL ツールによって基幹業務システムからデータを ETL して分析を行う。分析ユーザが投じるクエリの内容によっては、DB から大量のデータを読み出す場合もある。このことにより、以下のような問題を生じる。

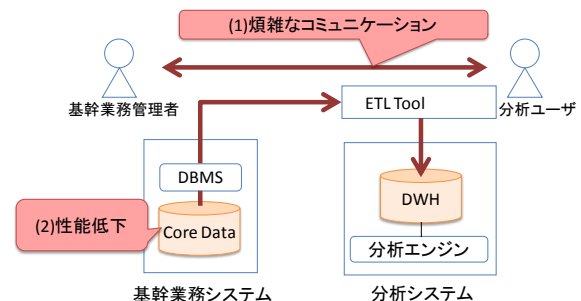


図 1 直接 ETL 方式

まず、誤設定などにより基幹システムのデータを破壊してしまうことである。また、ETL を実行することにより基幹業務システムの性能が低下し業務に影響を与えることである。これらを防ぐためには、分析ユーザは基幹業務管理者と事前に念入りなコミュニケーションをとり、基幹業務システム側の設定をしてもらう必要がある。また基幹業務管理者は、業務に影響の出ないデータ量または時間帯を見極めた上で、分析ユーザに対して制限付きでデータの取得を許可することになる。分析内容が日々変わるトライアンドエラーのフェーズでは、分析ユーザはデータが必要となるたびに基幹業務管理者にデータ提供依頼をするなど、頻繁なコミュニケーションが要求される。

以上より、直接 ETL 方式の課題は、(1)基幹業務管理者との煩雑なコミュニケーションの削減と、(2)ETL による基幹システムの性能低下の抑止の 2 点である。

[†](株)日立製作所 研究開発グループ 情報通信イノベーションセンター, Hitachi, Ltd., Research & Development Group, Center for Technology Innovation - Information and Telecommunications.

[‡](株)日立製作所 ICT 事業統括本部, Hitachi, Ltd., Information and Communication Technology Business Division.

3. 高信頼 ETL 方式の提案

2章で示した課題は、分析ユーザの ETL 処理が基幹業務システムに対して直接影響する点と、分析の内容と時間（つまり基幹業務システムへの影響）が不確定である点に起因する。そこで、ETL によるデータ取得を間接的に行うことを考える。具体的には、基幹業務システムのデータを予め別の場所にコピーしておき、コピーを分析ユーザに使用させることにより、操作ミスや性能影響を基幹業務システムと分離する。さらに、基幹業務システムから別の場所へのコピーを、ストレージの機能などを利用して定期的かつ低負荷で行うことにより、不確定要素を排除し、基幹業務管理者との都度のコミュニケーションを不要にする。

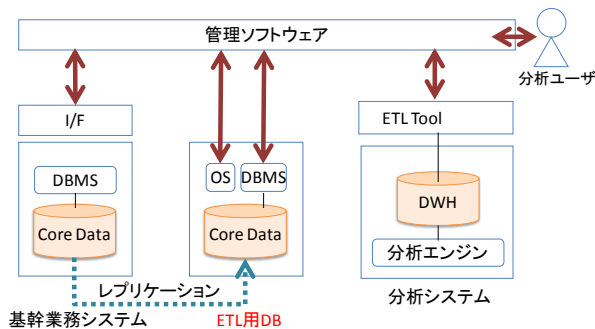


図2 高信頼 ETL 方式

以上を踏まえ、提案する高信頼 ETL 方式のシステムを図2に示す。高信頼 ETL 方式では、基幹業務システムと分析システムに加えて、ETL 用 DB を設置する。また、3つのシステムを管理する管理ソフトウェアを導入する。基幹業務システムと ETL 用 DB の間で事前にレプリケーションの設定を行っておき、分析ユーザが管理ソフトウェアを通じて分析を試みた場合、必要なデータを基幹業務システムではなく ETL 用 DB から分析システムへと ETL する。これにより、ETL による基幹業務システムへの性能影響を防ぐことができ、かつ基幹業務管理者へのデータ取得依頼も不要となる。よって、本システムにより分析ユーザは自由に分析を試行することが可能となる。

4. 性能評価

4.1 評価方法

提案する高信頼 ETL システムを実機を用いて評価する。評価では、業務システムに対して分析システムから ETL を実施することを想定し、その際に実行中の業務への影響を防げるかどうかを評価する。評価環境には、基幹業務システムとして、PC とストレージから構成される擬似業務システムを、分析システムとして ETL ツールを搭載した PC を用いる。高信頼 ETL 方式の実現にあたっては、実際の環境では ETL 用 DB を置くマシンを別途用意して業務システムと完全に分離することが望ましいが、本評価では影響の有無を見ることが目的なので、ストレージ内に ETL 用 DB を想定したコピーを作成する。以上の評価環境の構成、及び機器のスペックを、それぞれ図3及び表1に示す。

表1 評価環境機器のスペック

#	System	Specification
1	擬似業務システム(VM)	CPU : 4vCPU Memory : 8GB
2	擬似業務クライアント(VM)	CPU : 8vCPU Memory : 4GB
3	ETL tool (PC)	CPU : 4-core, 3.4GHz Memory : 16GB
4	ストレージ	Hitachi VSP

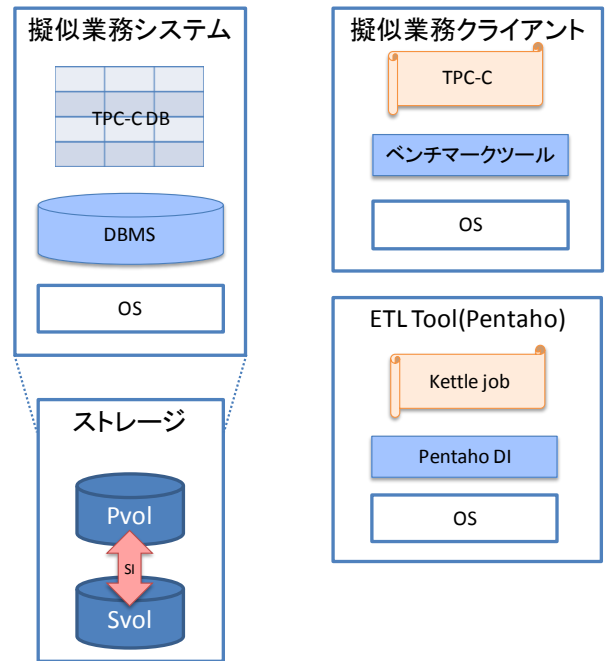


図3 評価環境の構成

次に、直接 ETL 方式及び高信頼 ETL 方式における性能影響の測定の手順を、それぞれ図4及び図5に示す。

直接 ETL 方式の評価手順

1. 擬似業務システムに OLTP を発生させる。具体的には、擬似業務クライアントより擬似業務システムへ TPC-C を実行し OLTP を模擬したリクエストを実行する。
2. OLTP 動作中に ETL 処理を実行する。具体的には、TPC-C 実行中に ETL ツール(Pentaho)[2]サーバから Pentaho DI を利用して ETL を実行する。

高信頼 ETL 方式の評価手順

0. 事前に SI ペア※を組んでおく。DB で利用するボリュームについて、事前に SI でレプリケーションしておく。
1. 擬似業務システムに OLTP を発生させる。具体的には、擬似業務クライアントより擬似業務システムへ TPC-C を実行し OLTP を模擬したリクエストを実行する。
2. TPC-C 実行中に ETL することを想定して、ETL する前に SI のペアをスプリットする。

3. Svol をマウントして ETL を実行する。
 ※ShadowImage: ディスクアレイ内にサーバを経由せず正ボリューム(Pvol)のレプリカ(副ボリューム:Svol)を作成する機能。一時的に同期を停止することをスプリットという。

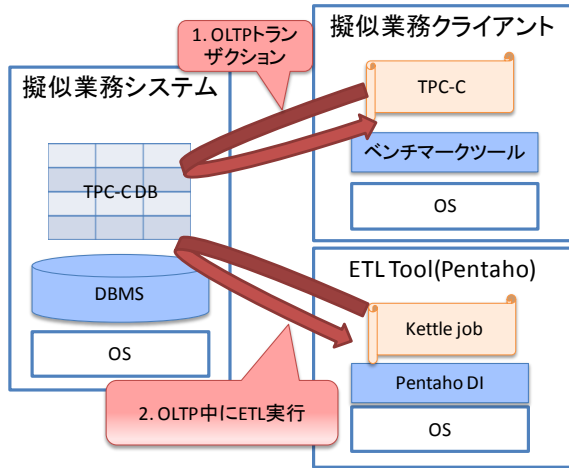


図 4 直接 ETL 方式の評価手順

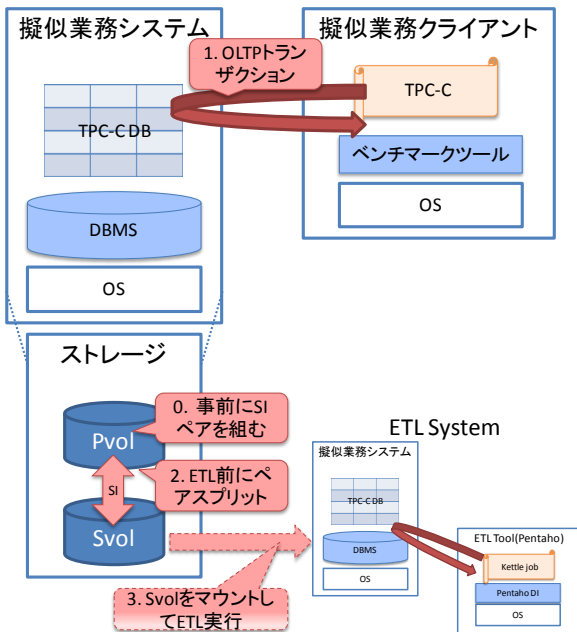


図 5 高信頼 ETL 方式の評価手順

4.2 評価結果と考察

直接 ETL 方式を用いた場合、及び高信頼 ETL 方式を用いた場合の評価結果を、それぞれ図 6 及び図 7 に示す。横軸は経過時間を、縦軸は基幹業務サーバにおける TPC-C のスループットを示している。また、グラフ中に枠で囲われている部分は ETL 中であることを示している。

図 6 より、直接 ETL 方式では、本評価環境において ETL を実施していない時間の性能が平均 652req/sec であったのに対し、枠で示した ETL 中の性能は平均 476req/sec であり、ETL により性能低下 (平均 26%) することが分かっ

た。一方、図 7 より高信頼 ETL 方式では、ETL を実施していない時間の性能が平均 647req/sec であったのに対し、ETL 中 (ETL 前のペアスプリット操作も含む) も性能低下がほとんど見られなかった。また、ETL 中の性能変動を、直接 ETL 方式に比べて 1/8 に抑えられることが分かった。

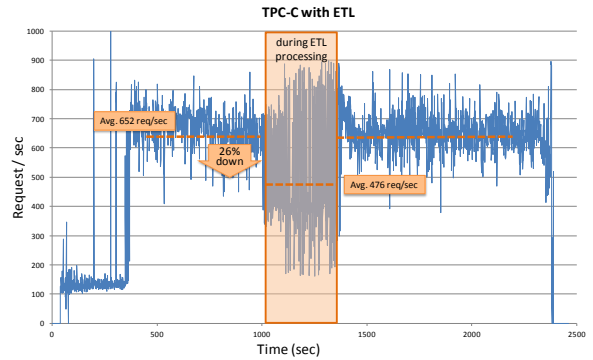


図 6 直接 ETL 方式の評価結果

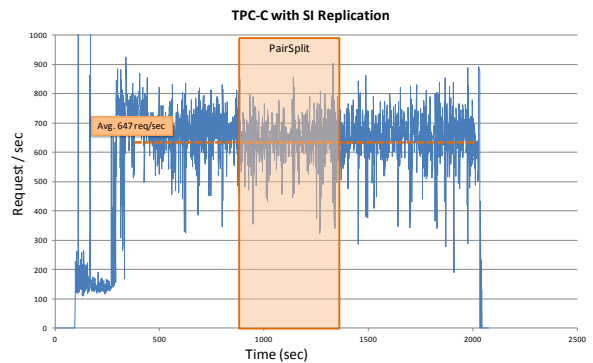


図 7 高信頼 ETL 方式の評価結果

性能影響の大きさは、実際に運用中の業務用 DB の性能設計 (マージンやリソースの割り当て方)、その時の業務の繁忙さ、ETL で要求するクエリやデータ量によって変動すると考えられる。しかし、直接 ETL 方式では ETL 処理によって業務設計の想定以上のリソース要求が発生し業務上のリスクとなり得る事は明らかである。一方、高信頼 ETL 方式では、サーバ、ストレージ、ミドルウェアにまたがって性能を分離して ETL することができるため、このような業務リスクを低減することができる。よって、提案する高信頼 ETL 方式は有効な手段であると言える。

5. 関連研究との比較

昨今、業務にて収集されたデータの分析とその業務へのフィードバックに向けていくつかのメーカーが提供しているソリューションの一つが OLTP で使う DB と OLAP で使う DB を統合化した統合 DB によるシステムである(図 8)。これらは、OLTP に適した Row 型のデータと、OLAP に適した Column 型のデータをともに扱うことができ、また用途に合わせて変換する機能性を持っている[3][4][5]。

分析システムで行うことや、それによって生み出される価値、またその性能要件が明確で、業務システム含めて全

体のシステム設計ができるタイミングでは有効なソリューションと考えられる。また、分析向けのデータ管理が業務向けシステムと同様にエンタープライズハードウェアで行われるので比較的高価になる傾向がある。

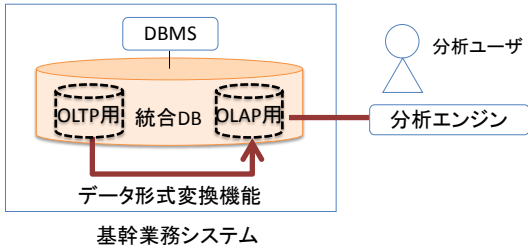


図 8 統合 DB 方式

一方で、分析によってどんな価値が生まれるかわからない状況で、まだ分析のトライアンドエラーが必要な場合や、コスト的に統合 DB 方式は採用できないものの、本業への影響を抑えながら分析を行いたい場合などは、業務 DB から必要なデータを抽出して、分析向けに DWH を設計することとなる。データ鮮度のリアルタイム性が不要でデータへのアクセスを細かく制限したい場合、かつ、業務システムの管理者をたっぷり巻き込んで分析向けシステムの DWH 設計を実施できる場合に有効と考えられる。

以上の方式と高信頼 ETL 方式を比較し、表 2 にまとめる。

(#1) データ鮮度のリアルタイム性：

高信頼 ETL 方式では、業務側での更新が ETL 用 DB に同期され次第 ETL すれば分析可能であり、ETL は性能影響を与えずにオンデマンドに実行できる。統合 DB 方式では、業務側で更新があり次第即時分析に利用できる。直接 ETL 方式では、業務システムが暇な時にバッチ処理にてデータを取得するため、データ鮮度はバッチ間隔に依存する。

(#2) 基幹システムの性能低下：

高信頼 ETL 方式では、性能低下はない。統合 DB 方式では、更新をマージする[6][7][8]ために CPU パワーを使うので、性能低下の可能性がある（ただし影響の大きさは更新量等に依存する）。直接 ETL 方式では、4 節で示したように ETL の影響により 26%性能低下する。

(#3) 分析システム導入時のシステム設計：

高信頼 ETL 方式では、既存業務システムへの影響を最小化（ストレージのコピー設定のみ）できる。統合 DB 方式では、業務システム含めて、全体的にシステム設計をし直す必要がある。直接 ETL 方式では、DWH へ持って行くデータの選定や、データ抽出用バッチの実行タイミングなど、分析ユーザと業務管理者で調整が必要となる。

(#4) システム要件策定時の Try and Error フェーズの運用：

高信頼 ETL 方式では、業務管理者との調停なく、新しいデータの取得が可能である。統合 DB 方式では、統合 DB を利用してテストをすると、本番業務システム側に性能影響が出る場合がある。直接 ETL 方式では、新たなデータが必要になった場合に、分析ユーザと業務管理者間で調整が必要となる。

(#5) 分析結果のフィードバック：

高信頼 ETL 方式、及び直接 ETL 方式では、分析結果は別途アプリケーションレベルでフィードバックする必要が

ある。統合 DB 方式では、分析結果が統合 DB に反映されれば、業務システム側で利用可能となる。

表 2 方式の比較

#	比較項目	高信頼 ETL 方式	統合 DB 方式	直接 ETL 方式
1	データ鮮度	△ ～数分	○ ～数秒	× ～数時間
2	基幹システムの性能低下	○ なし	△ 可能性有	× 26%
3	分析システム導入時のシステム設計	○	×	△
4	システム要件策定時の Try and Error フェーズの運用	○	△	×
5	分析結果のフィードバック	△	○	△

6. おわりに

現業として SoR を有する企業が、自身のビジネスを変革し守るために SoE を導入し、分析を試みようとする場合、基幹業務システムから直接 ETL を実施すると管理者間の協議の手間の削減や性能低下の抑止と言った課題がある。本稿ではこの手間を低減しつつ基幹システムの性能劣化を防ぐ高信頼 ETL 方式を提案した。評価の結果、直接 ETL 方式では平均で 26%性能劣化し性能変動も大きいのに対し、提案手法では性能変動を 1/8 に抑えることに成功した。これにより、少ない手間で基幹系データを分析に活用できる見込みを得た。

今後は、分析ユーザがもっと気軽にかつ的確な分析を実施できるように、例えば分析テンプレートの拡充、分析システムの UI の改良といったことが必要である。

参考文献

- [1] W.H.Inmon, "Building The Data Warehouse", John Wiley & Sons, 1992.
- [2] Pentaho, "http://www.pentaho.com/".
- [3] D. Kernert, F. Köhler, and W. Lehner, "SLACID – sparse linear algebra in a column-oriented in-memory database system", In Proceedings of the 26th International Conference on Scientific and Statistical Database Management (SSDBM'14), ACM, 2014.
- [4] T. Lahiri, S. Chavan, M.Colgan, D. Das, A. Ganesh, M. Gleeson, S. Hase, A. Holloway, J. Kamp, T-H. Lee, J. Loaiza, N. Macnaughton, V. Marwah, N. Mukherjee, A. Mullick, S. Muthulingam, V. Raja, M. Roth, E. Soylemes, and M. Zait, "Oracle Database In-Memory: A Dual Format In-Memory Database", In Proceedings of 2015 IEEE 31st International Conference on Data Engineering (ICDE), Apr 2015.
- [5] D. J. Abadi, S. R. Madden, and N. Hachem, "ColumnStores vs. RowStores: How Different Are They Really?", In Proceedings of ACM SIGMOD'08, Jun 2008.
- [6] M. Colgan, "Oracle Database In-Memory", White Paper, Oracle Inc., Jul 2015.
- [7] W. Endress, "Oracle Database In-Memory: In-Memory Aggregation", White Paper, Oracle Inc., Jan 2015.
- [8] D. Das, J. Yan, M. Zait, S. R. Valluri, N. Vyas, R. Krishnamachari, P. Gaharwar, J. Kamp, and N. Mukherjee, "Query Optimization in Oracle 12c Database In-Memory", In Proceedings of the 41st International Conference on Very Large DataBases (VLDB), Vol.8, No.12, Sep 2015.