B-010

上流設計におけるシステム性能予測と評価
Performance Estimation and Evaluation in the System Design

吉村　礼子†　　魚住　光成†
Ayako Yoshimura　　Mitsunari Uozumi

1.　はじめに

近年、ソフトウェアの開発工程は短縮される傾向にあり、開発のリスクは増加する一方である。そのため、開発プロセスの導入により、開発作業の統一を図り、品質の均一化および作業効率の向上を行うことが重要である。しかし、開発プロセスに従っていても、上流設計における問題が、実装後に発覚する場合も少なくない。システム性能については、原因究明に手間取り、再設計が必要となるなど開発の手戻りが大きな負担となる[1]。そこで、本研究では、開発の手戻りをなくし開発のトータルコスト削減を目指し、上流設計時に性能設計の妥当性を評価するシステム性能予測手法の提案を行う。

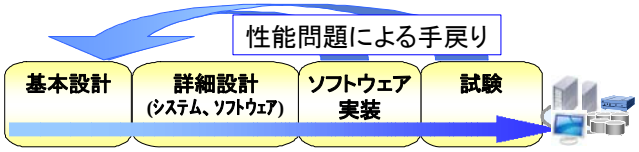

図1　システム開発の流れ

2.　課題

情報システムの性能を悪化させる原因は様々であるが、ディスクアクセスが発生するデータベースの処理に原因があることが少なくない。この原因を上流設計時に発見できるほど、手戻り工数は削減される。しかし、機能実現が優先され、データベースの性能については DBMS に依存して後回しにされることもある。

3.　従来手法

従来、データベースアクセスの性能は、システム開発を行って動作させてから、性能問題が発生した箇所に対してチューニング作業により解決を行うことが多い。例えば Oracle では、統計情報などのパフォーマンス情報を提供し、S/W 完成後の性能向上を支援している[2]。しかし、この方法ではテーブル構造やアクセス方法に見直しが必要になった場合に、大きな手戻りを伴うことになり抜本的な解決にはなっていない。

4.　データベースアクセス性能予測手法

本研究では、このデータベースアクセス応答性能をターゲットとし、上流設計時に設計情報から性能予測する作業を開発プロセスに組み込み、問題となる設計を早期に抽出することを目的とする。

4.1　データベースアクセス性能予測の方針

データベースアクセス応答の処理は、ディスク I/O 時間に大きな割合を占められることから[3]、本研究では「ディスク I/O」に着目して問題となる設計の抽出を行う。

† 三菱電機（株）情報技術総合研究所
Information Technology R&D Center, Mitsubishi Electric Corporation

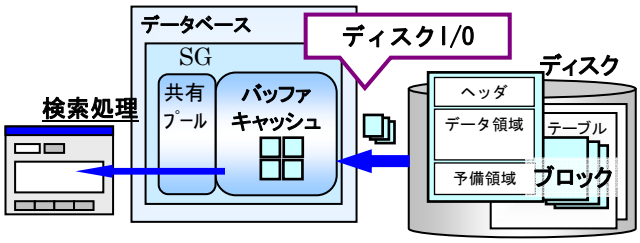

図2　データベースアクセス処理

4.1.1　バッファキャッシュへの占有率

バッファキャッシュに読み込まれたデータが全体の 2%を超えた場合、キャッシュから外される可能性が高くなり[2]、ディスク I/O 発生を引き起こすことになる。そこで、テーブル毎にデータサイズを見積もってバッファキャッシュ占有率を算出し、データサイズの大きいテーブルを性能を悪化させる要因として抽出する。

4.1.2　ディスク I/O 処理時間

テーブル設計情報から、全表スキャンが発生しディスク I/O が発生した場合の処理時間を最悪パターンとして算出し、目標を超える処理について抽出する。

4.2　設計情報に基づいた性能予測

システム設計時には上流から段階的に性能が決定する要素が設計されており、この決定要素に着目して性能評価を行う。上流設計時の取得項目は以下のものがある。

4.2.1　基本設計で取得できる情報

H/W 情報、データベース設定情報、テーブル基本設計（ER 図など）から、テーブル毎のデータサイズを求め、以下の項目を算出する。

- データバッファの占有率
- I/O 回数を算出後、最悪パターン処理時間

4.2.2　詳細設計で取得できる情報

アクセス処理の性能に影響する要素[4]を、テーブル詳細設計（テーブル構成、データ件数）やデータベースアクセス処理情報（索引項目、データ分布）から取得し、テーブルを全表スキャンした場合の最悪パターンと、索引指定が有効に効いた場合の最善パターンの処理時間を算出する。

- アクセス処理毎の最善／最悪パターンの処理時間

5.　評価方法

5.1　開発プロセスへの組み込み

性能予測評価の作業の統一と予測精度向上を目的に、テーブル設計とデータベースアクセス設計に対する評価を組み込んだ開発プロセスを設定した。基本設計では「H/W やテーブル構造の設計が妥当であるか」、詳細設計では「データベースアクセス処理設計が妥当であるか」の評価を行

い、各設計段階での性能評価に基づいて必要に応じて再設計を実施するプロセスとした[5]。

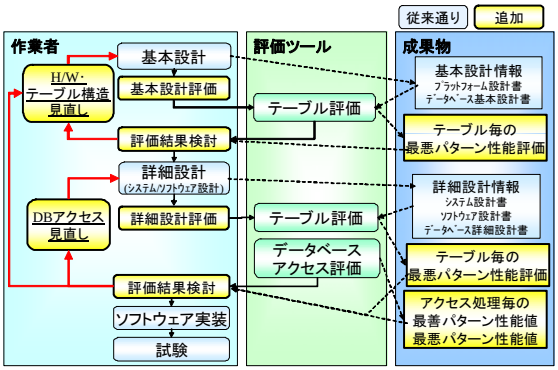

図3　開発プロセスにおける性能予測

5.2　性能評価ツール

性能評価ツールとしてテーブル評価シート、データベースアクセス評価シートを用意した。

5.2.1　テーブル評価シート

データベース設定情報・テーブル構造・データ件数を入力することにより、バッファへの占有率、処理時間を算出し、目標超えのテーブルを抽出する。

5.2.2　データベースアクセス評価シート

対象項目・索引指定・データ分布などを入力情報として以下の項目を算出し、目標性能を超える処理を問題ありとして抽出する。

- 索引スキャン時の処理時間
  - ユニーク索引検索／索引検索（対象件数：少）
- 全表スキャン時の I/O アクセス処理時間
  - 全表スキャン検索／索引検索（対象件数：多）

6.　実システムへの性能予測評価

今回、実際に稼動している 2 つのシステムに対し、今回設定した開発プロセスに従って、設計情報から性能予測評価を実施し、予測結果と実測値との比較を行った。

6.1　基本設計に対する評価

設計書から取得した情報を入力してバッファ占有率と I/O アクセス処理時間を算出し、目標値を越えた 2 つのテーブルを抽出。今回は再設計なしで次の評価へ進んだが、実際には、再設計を行うかあるいは詳細設計へ進み、要注意テーブルとして設計を行う。

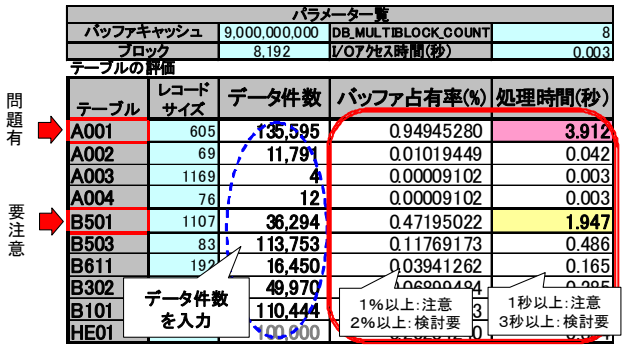

図4　テーブル評価結果の例

6.2　詳細設計に対する評価

再度詳細設計に基づいて行ったテーブル評価結果とアクセス情報を元に処理時間を算出し、目標性能との比較を行った。その結果、テーブル評価でも要注意であったテーブルに対し、索引指定を行っているものの該当データが多いと思われる処理に対して、全表スキャンの可能性が高く、要注意の処理として抽出した。もう 1 件の要注意のテーブル利用の処理では、該当データが少なく索引が有効に作用し、問題なしと評価した。

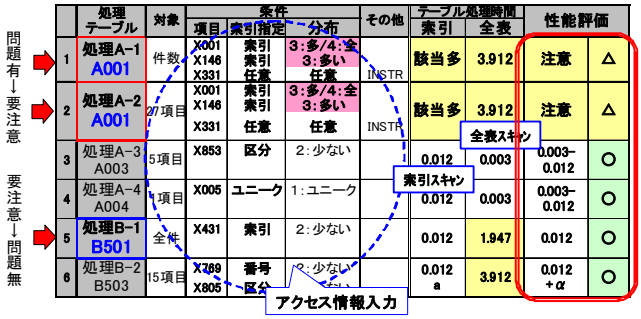

図5　データベースアクセス評価結果の例

6.3　評価結果

設計情報から 18 件の性能予測評価を行い、「16 件問題なし、2 件を見直し要」に対して、実測値と比較した結果は「17 件問題なし、1 件問題あり」であった。この問題ありの処理は、該当するデータ件数に比例して処理時間が増加しており、問題ありと指摘していた処理のうちの 1 件と一致した。また、基本設計評価で要注意とした後、詳細設計評価で問題なしとした処理については、評価通り問題なしであった。

このことから、今回取り上げた事例では、稼動後に問題となる処理に対して設計段階での抽出が行えており、性能向上のための再検討すべき項目の有効性と開発プロセスへの組み込みの有効性が確認できた。

7.　おわりに

本研究では、「基本設計からソフトウェア設計まで設計の各段階で、どこまで性能が予測できるかということ」を目的とし、設計情報から問題のある設計の抽出が可能であることを確認できた。また、設計段階ごとに設計の妥当性を評価し、再設計を実施、あるいは評価情報を次の設計へ追加情報として利用する性能予測評価の作業を開発プロセスに組み込むことの有効性を確認できた。今後は、その他の CPU やネットワークによる処理性能などにも取り組んでいく予定である。

参考文献

[1]岡崎公洋, “プロマネとしての品質マネージメントに関する教訓について”, プロジェクトマネジメント学会(2006).
[2]Oracle, “Oracle Database パフォーマンス・チューニング・ガイド 11g リリース 2（11.2）” (2010)
[3]C.J.Date, “An Introduction to Database Systems ”, ADDLSON WESLEY, Vol.1 (1990).
[4]五十嵐 建平, 大塚 信男 , 小田 圭二 , 鈴木 博貴, 村方 仁 , “門外不出の Oracle 現場ワザ”, 翔泳社 (2005).
[5]Fuggetta,A.,et al., “ソフトウェアのトレンド”,海文堂(19970)