

B-032

システム開発における性能評価手法
Performance Evaluation Method in the System Design

吉村 礼子[†] 魚住 光成[†] 樋口 毅[†]
Ayako Yoshimura Mitsunari Uozumi Tsuyoshi Higuchi

1. はじめに

近年の情報システムでは、扱うデータ量が膨大になり、この影響により性能面での課題が発生する機会が少なくな。しかし、開発段階では本番環境での動作確認が難しく、性能問題の発覚が遅れ開発手戻りが発生し、開発コスト増大や納期遅れにつながる事が多い。そこで、我々は、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った [1]。本研究では、上流から実装および試験の各プロセスで性能を評価し、妥当性を確認して次へ進む開発プロセスと、それぞれの評価で課題を明らかにする性能評価ツールを用いた性能評価手法について提案を行う。

2. 課題

システム性能の一番問題となりやすいデータベース処理では、ディスクアクセスの発生が原因であることが多い。しかし、設計段階ではそこまで意識せず、性能を考慮されないままに開発が進んでいき、問題に気づいた時には原因究明や改修に大きなコストがかかるという課題がある。

3. 従来手法

従来、データベースアクセスの性能は、機能を実装した後、試験段階になって初めて性能を確認し、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。この場合、例えば商用の RDBMS 製品では、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、この方法ではデータベースの機能を熟知していないと有効に利用できず、性能改善の手がかりをつかむことは簡単ではない。

4. 性能評価手法

4.1 目的

これらの課題を解決するため、開発の早い段階から性能を意識させ、開発ステップごとに性能評価を行う開発プロセスと性能評価を行うツールを用いた性能評価手法を提案する。

4.2 性能評価の方針

データベースアクセス応答の処理は、ディスクアクセス時間に大きな割合を占められることから[3]、本研究ではこの処理の効率化を意識して、図 1 に示すような「ディスクアクセス効率化」「キャッシュ利用有効化」「SQL 文実行効率化」の 3 つのポイントに着目し、それぞれに対応する検討項目を明確にして評価を行っていく。

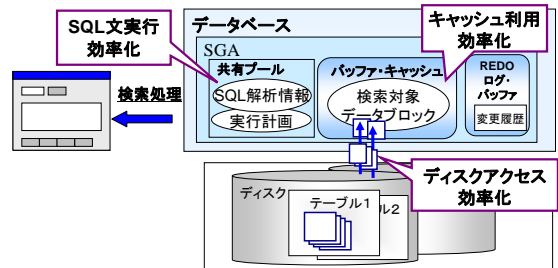


図1 評価の着目ポイント

4.3 開発プロセス

提案する開発プロセスでは、性能に影響する検討項目をステップごとに明確にし、課題が出た場合には必要に応じて前の段階へ戻って再検討を行うプロセスとする。

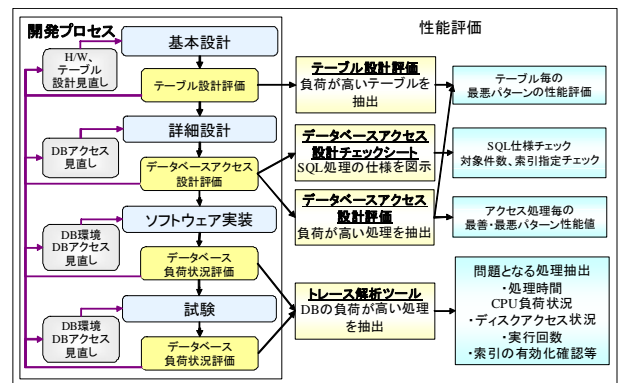


図2 開発プロセス

(1) データベースアクセス設計評価

「SQL 文実行効率化」を目的に設計内容を明確にするため、以下の項目を図示しチェックを行う。

- ・ 処理の流れ
- ・ 検索処理時のデータ絞込み状況
 - 母数と抽出件数
- ・ 検索処理時の条件項目
 - テーブル間、比較項目の関係を図示

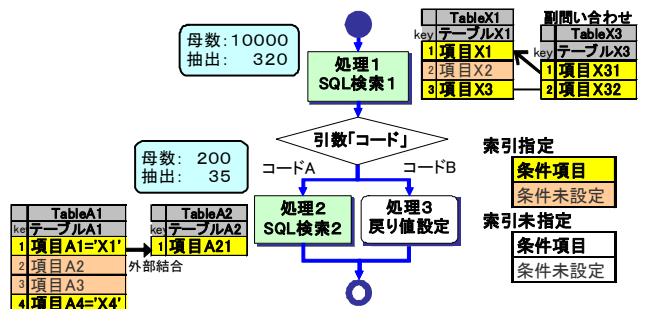


図3 データベースアクセス設計チェック

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

図示することにより、設計者の意識をあげ、さらに、設計のレビューの精度を上げることが可能となる。

(2) データベース負荷状況評価

ソフトウェア実装において動作するようになったシステムに対し、データベースで取得したトレース情報をツールによって自動的に解析し、図 1 に示した 3 つのポイントからデータベース負荷状況が高い処理を抽出する。例えば「ディスクアクセス効率化」はディスクアクセス量、「キャッシュ利用有効化」はヒット率、「SQL 文実行効率化」はスキャン率 (データ取得する件数に対してスキャンするブロック数の割合) の値から負荷状況を判断する。

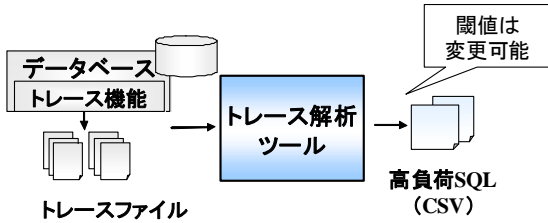


図4 トレース解析ツール

5. 性能評価の適用

今回提案した性能評価手法の評価対象として、設計段階で性能評価を行わず、稼動後に性能問題が発生している作業管理システムを適用した。このシステムはサーバーにてデータベースサーバへアクセスし、大量データに対する検索機能をもつ C/S 型システムである。

5.1 データベース負荷状況評価による課題抽出

検索処理をメインとする 2 つの機能に対し、実装段階に行う「データベース負荷状況評価」を実施し、データベースの負荷が特に高いと思われる項目に対し着目した。

表 1 機能 1 のトレース解析結果

SQL	CPU	時間	ディスクアクセス	読み込み	取得	ヒット率	スキャン率	回数
446	0.27	0.26	0	19310	9	1	2145.56	1
448	0.65	6.14	8771	11953	100	0.2662	119.53	1
481	0.41	7.58	5201	6019	152	0.1359	39.5987	1
154	0.68	1.88	1952	18529	1111	0.8947	16.6778	13507
157	0.18	0.23	0	10059	0	1	0	11241
153	0.16	0.09	0	3762	1170	1	3.21538	13507

機能 1 では、スキャン率が高く検索して 1 件取得するために 2145 ブロック読み込む SQL や、実行回数が 1 万回を超える SQL などが高負荷として抽出され、SQL 文実行が効率的に行われてない。

表 2 機能 2 のトレース解析結果

SQL	CPU	時間	ディスクアクセス	読み込み	取得	ヒット率	スキャン率	回数
94	17.3	18.37	389	219890	631	0.9982	348.479	1
96	12.8	13.27	3912	171544	348	0.9772	492.943	1
33	1.71	55.54	10985	101462	5973	0.8917	16.9868	1
37	1.05	1.01	8064	8171	197	0.0131	414.772	1
113	4.88	4.8	22269	109796	907	0.7972	121.054	1
72	0.2	4.66	2793	6598	3955	0.5767	1.66827	1

機能 2 でもスキャン率が高く、バッファ読み込みや、ディスクアクセス量の多い SQL が抽出されており、SQL 文実行が効率的でなく、ディスクやバッファへのアクセス増大により処理時間も長くなっている。

5.2 データベースアクセス設計評価

これらの課題に対し、詳細設計に戻りデータベースアクセス設計評価を実施した。機能 1 では、実行回数が多い処理は処理の流れを中心に確認し、効率の悪い SQL の繰返しを抽出。スキャン率が高い処理では条件指定された項目があり、データの絞込みを行わず広範囲の検索処理が実行されていることが判明した。機能 2 では、バッファやディスクへのアクセス量の多い処理に対し、索引項目が条件に利用されていないことが判明した。

key	AD3630	key	AD3470	
1	X11352	1	X22593 = A01	索引指定
2	X12008	2	X12008	条件項目
3	X11322	3	X11322	条件未設定
4	X11030	4	X11538	索引未指定
5	X27713 = 'X11'	5	X11026	条件項目
	(X12777 X12778)	6	X29459	条件未設定
	>: A11			

図5 機能 2 のチェック結果

5.3 評価結果に基づいた改善

データベース負荷状況評価の結果を元に詳細設計を見直しデータベースアクセス設計評価を実施。得られた課題へ対応を行った結果、SQL の処理性能で以下の表に示すように機能 1 で 73%、機能 2 で 76%の改善が得られた。

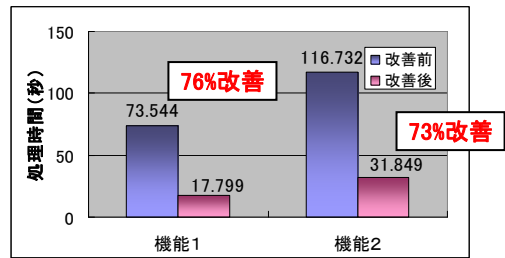


図6 性能改善結果

今回取り上げた事例では、稼動後に発生している性能問題が、詳細設計の性能評価で抽出できる問題であり、実際にこのシステムの設計時に評価を行っていただければ稼動後の性能問題は発生していなかったと思われる。従って、提案した性能評価手法の有効性を確認できた。ただし、索引やキャッシュの効果を利用する際のデータの偏りの影響への評価については解決できておらず、今後の課題として明らかになった。

6. おわりに

本研究では、システム開発の各段階で性能評価を実施して性能問題を早期解決する開発プロセスを提案した。実際の稼動システムに対する性能評価で課題および原因の抽出により性能改善され、開発プロセスの有効性を確認した。今後は、さらにデータの偏りによる影響についても検討を行い、予測精度向上や評価作業効率の向上に取り組んでいく予定である。

参考文献

- [1] 吉村 礼子, 魚住 光成, “上流設計におけるシステム性能予測と評価”, FIT2011(2011).
- [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).