

3. 19 RDBMSのクエリ最適化機能に関する教訓 (T 19)

[教訓T19]

リレーショナルデータベース(RDBMS)のクエリ自動最適化機能の適用は慎重に！

問題

A社の営業拠点は全国に約 1,000 ヶ所存在する。この営業支援システムは全国地域を数個のブロックに分けて順次サービスインを行いシステムリソースの増強を行いながら全国規模で順調に稼働していた。

営業拠点の活動は、当日訪問する既得意あるいは新規の顧客に対して提案資料を営業拠点で朝一番にオンライン作成・印刷しパンフレット等と同梱して営業に出発する形態が多い。オンラインシステムは朝7時に起動され、8時から9時30分が利用のピークとなる。ピーク時には約50件/秒のトランザクション処理が発生していた。ある朝、オンラインシステムのタイムアウトが多発し運用監視コンソール端末にトランザクション異常終了のメッセージが連続的に出力された。利用者はシステムからの印刷物が得られず営業活動に出られない状況となった。図3. 19-1にシステム構成の概要を示す。

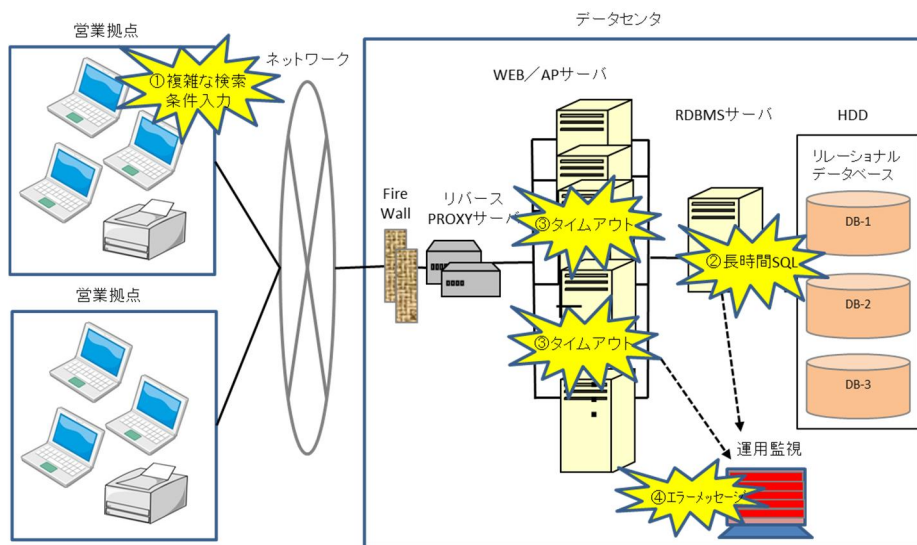


図3. 19-1 A社営業支援システムの概要

障害の発生状況は次のようであった。

当日1件目のトランザクションにおいて利用者から広範囲にわたる検索条件入力が投入された(①)ため、DBサーバで稼働するリレーショナルデータベース (RDBMS) の自動最適化機能により全件フルスキャンを行う SQL 実行計画が適用され RDBMS サーバ内で SQL 処理が長時間実行されたものである(②)。これにより、AP サーバの SQL 監視タイマのタイムアウト(3分)となりアプリケーションが異常終了した(③)。また、以降のトランザクションも同じ SQL 実行計画が適用され続け、タイムアウトが連続的に発生し運用監視コンソール端末に大量のエラーメッセージが出力され続けた(④)。

原因

RDBMS は DB のデータ分布状態の統計情報と起動後最初に行われる SQL の内容により最適と判断した SQL 実行計画を選択する (クエリ最適化機能)。統計情報取得は日次夜間バッチによるデータベース再構築後に毎日実施している。

図 3. 19-2 はデータの変動と RDBMS 性能 (レスポンスに要する時間) の関係を表したものであり、一般的にデータベースのデータ件数増大に伴って徐々に性能 (レスポンスに要する時間) は劣化する。

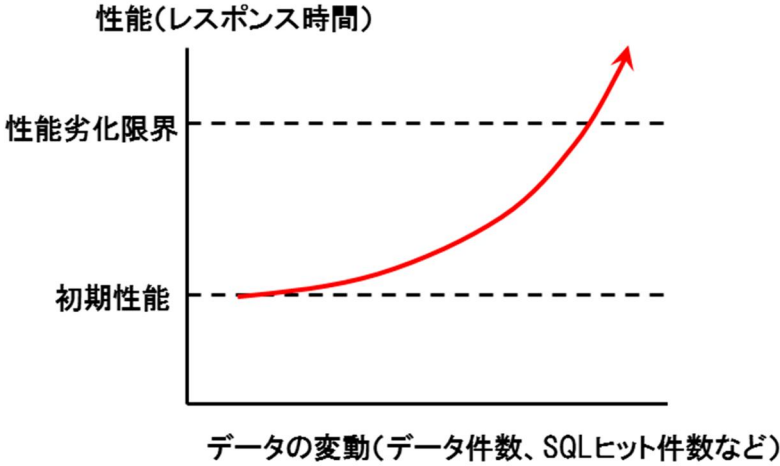
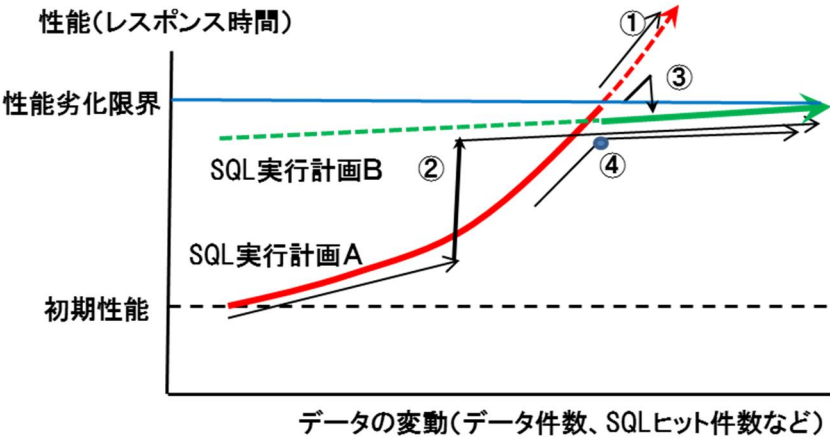


図 3. 19-2 データ変動に伴うレスポンスの劣化

クエリ最適化機能による SQL 実行計画の切替の概要を図 3. 19-3 「データの変動に伴う SQL 実行計画の自動切替」に示す。



- ① SQL 実行計画 A が切り替わらない場合 (徐々に性能劣化限界へ)
- ② 性能劣化限界以前に SQL 実行計画 A から B に切り替わったケース (突然性能劣化発生)
- ③ 性能劣化限界を過ぎた後で SQL 実行計画 A から B に切り替わったケース (徐々に性能劣化限界へ)
- ④ 性能劣化限界で SQL 実行計画 A が B に切り替わったケース (理想)

図 3. 19-3 データの変動に伴う SQL 実行計画の自動切替

RDBMS の SQL 実行計画は最適化機能によりデータの変動に応じて選択されるが、図 3. 19-3 に示すように性能劣化限界点で瞬時に切り替わるものではなく、統計情報と SQL のヒット件数で危険域を予測し事前に切り替わることが多い。今回は、最初のトランザクションの SQL ヒット件数が大量となることにより、フルスキャン型の SQL 実行計画 B が選択されたと推察できる。つまり図 3. 19-3 の②のケースが最初の SQL 実行計画選択時に発生したと考えられる。

その後、2 件目以降のトランザクションにおいても SQL 実行計画 A に戻らず続行されたため、通常のトランザクションもフルスキャン型となり長時間処理、タイムアウト多発に至った。

根本的には RDBMS のクエリ最適化機能の特性と利用者の使い方に関する分析が不足していたことが原因として挙げられるが、十分な分析を事前に行うことは難しく、同様の障害が他のサイトでも発生することが多い。

対策

データセンタでは緊急措置としてオンラインサービスを停止し、RDBMS を再起動して回復したがこの間はサービスが中断となった。

翌日以降の当面の運用対処として早朝にアプリケーション担当者がオンライン起動後の最初のトランザクションを通常のインデックスを使用する検索条件で投入し SQL 実行計画 A が選択されるようにした。

抜本的な対策として、表 3. 19-1 に示すようにインデックスを追加しフルスキャンが発生しないようにする、あるいは SQL 実行計画を固定する（統計情報を取得しない）ということが考えられるが、A 社は更新の負荷増大とクエリ最適化機能を無効化することのメリット・デメリットを比較しインデックスを追加する方策を 1 ヶ月後にリリースした。

表 3. 19-1 SQL 実行計画の変更による性能劣化発生への対策

案	対策案	メリット	デメリット	備考
1	インデックスを追加	クエリ最適化機能は活用できる	更新負荷の増大による性能劣化の恐れあり	データ量の変動が予測される場合
2	SQL 実行計画の固定化(利用率の高い SQL のみ)	案 3 に比べて効率的	利用率が均等だと効果は小さい	システム全体に影響させたくない場合
3	全ての SQL 実行計画を固定化	SQL 実行計画の自動切替は発生しなくなり極端な劣化は発生しない。	データ量の変化に追従しないため高速処理は望めない。	極端に遅くなるのを防止したい場合

効果

抜本的な対策を実施後は、SQL 実行計画 A が常に選択され極端な性能劣化は発生しなくなった。

教訓

RDBMS の最適化機能はデータ変動と SQL のヒット件数を予測しながら SQL 実行計画を選択するため、必ずしも臨機応変に最適な選択が行われるわけではない。ユーザの利用要件を十分に分析し、データベース設計時にインデックスの付け方、SQL 実行計画の選択方法（SQL 実行計画を RDBMS の最適化機能に任せる、あるいは固定化し最適化機能を使用しない方式にするか）、また統計情報取得のタイミングと頻度（サービスイン時のみの取得にするか、あるいは定常運用で行う場合は頻度をどうするか）について検討しリスクを回避することが肝要である。このためにはアプリケーション担当とインフラ担当（データベース担当）の協力が欠かせない。