

6M-7 関係データベースを用いたシステムでの スループット向上について

西川 和彦
(株) 東芝

1. はじめに

1969年に提唱された関係型データベース(以下RDBと略す)モデルはさまざまな研究開発が実施され1980年代に商用化された。RDBは高度なデータ独立性、簡易なユーザインターフェイス、論理的基盤の確立したデータベースシステムとしてシステムの中心的存在となりつつある。応用分野としては事務データ処理が主であり人事・在庫管理や銀行の勘定システム、座席予約システムがある。

本論文ではオンライン系リアルタイムシステムの実例を紹介し、特に本システムでのスループット向上のための方法について論じる。

2. システム例

ここで紹介するシステムは、日常業務として行われる作業工程を管理するシステムでありその中核にRDBを有している。総テーブル数280、通常使われるレコード件数200000件、時平均約10000程のトランザクションが発生するシステムである。システムの概要を図1に示す。

本システムはTechnical centerで日常工程作業を行っている作業者が進捗状況や作業、勤務予定をLANでデータベースと接続された端末から入力し、データベースに登録する。Officeでは事務員が端末より登録された内容の承認や今後の作業内容、予定をスケジューリングしデータベースに登録、作業者に知らせると言ったものである。

2.1. 本システムの特徴

1) スケジューリング

入力された各工程の日程をスケジューリングする際には、休日や会社の行事予定、工場内工事のための使用不可な建屋、工具の予約状況などの一般的な制約と個人的な休日習得日や夜勤日など個人的制約がある。データベースにはこれら全てが格納されておりこれを検索または更新しながらスケジューリングを行う。

Method for throughput improvement of system
using Relational Database
Kazuhiko Nishikawa
TOSHIBA, Ltd

2) EWSとの負荷分散

作業員、事務員は共にAPL(アプリケーション)を通しデータベースに問い合わせを行うことによって必要なデータのみを取り出し、LANで接続されているEWSに処理させ美しい帳票を得ることができる。データ操作は全てCPU室のミッドレンジコンピュータが処理し、EWSがそれを加工、修飾するといった作業分散形態を取っている。このためミッドレンジコンピュータはデータベース操作に集中でき効率の良い運用が行える。

3) 遊休時間の利用

要求にはリアルタイムで処理しなければならないもの(動的)、システム遊休時間に処理を回しても差し支えないもの(静的)の2種類が存在し、静的なものに対してはタイマーが監視しシステム遊休時間にその処理を後回しにし動的要求の処理速度を落とさないようにしている。また、遊休時間を利用してデータの保守も行っており、使用しなくなったデータを別領域に保存、場合によっては削除を行っている。このことによって不必要なデータにデータベース操作時間を取られることがなくなる。

4) 別システムとのデータ通信

さらに別システムとデータのやり取りも行いより付加価値の高いデータ運用を行っている。

3. スループットの向上

リアルタイムシステムに於いての最も重大な問題は処理速度である。処理速度の向上を妨げる最も困難な原因はOSとDBM(Database Manager)の関係から生じている。商用のDBMはそのほとんどがOS上に構築されており、OSが提供するバッファ管理ルーチン及び入出力ルーチンに依存している。つまりDBMはOS上で動いているタスクである。しかし、この関係ではOS、DBM両者間の機能に不整合が生じ、こ

れに帰因する性能低下が現れている。商用DBMSでは処理速度の速いCPUや、入出力ルーチンを用いる方向でしか逃れる術がないのが現状である。

しかし、OS DBMの機能を調整することである程度の性能向上(スループットの向上)を図ることができる。本章では2章で紹介したシステムでの調整法について述べる。

データベース操作速度を上げるにはデータベースバッファ数を多くしテーブル操作要求に対するヒット率を上げなければならない。しかし、DBMはOS上のタスクにすぎないためスワップアウト、ページフォルトが多発してしまい思うように操作速度の向上は望めない。そこで、OS側のバッファキャッシュ数の増加、ワーキングスペース(仮想記憶を用いている際タスクに割り当てられる主メモリ上のページ数)の拡張、DBMの優先順位を高位にする等の策をこうしてみる。

確かにDBM側のスループットの向上はある程度まで得られるがある境目よりスループットの低下が発生する。それはOSや他のアプリケーションの処理速度がDBMの動作を優先しすぎたばかりに低下したためである。特に今回紹介したシステムで問題となったのはCPUパワーと主メモリをOSとDBMで奪い合い動作が混乱してしまうといったものであった。以上のことにより、システムの処理速度はOSとDBMへのマシン資源の寄与率に関係しているといえる。一言でマシンの資源といってもその項は多く、また各項が相互に関係している形となっており最適値を求めるのは困難である。実際のシステム構築に於いては仕様も考慮にいれ最適値付近を試行錯誤し導き出すといった方式となっている。図2にこの概念図を示す。図2は横軸にDBMへの資源の寄与率を示し、縦軸に処理速度を示している。実線がOSの資源寄与率を処理速度の

関係を示したグラフ、破線がDBMのグラフである。点Pは両者を加えた場合に処理速度が最大となる点であり、これを目標としてシステム調整を行うことが望ましい。

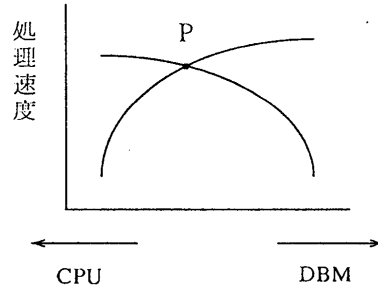


図2資源寄与率と処理速度の関係

4. まとめ

本論文ではオンライン系リアルタイムシステムに於けるRDBの処理速度に関する問題点を事例を挙げて提示した。またこの問題に於ける対処法についても述べた。

本論文を通じて感じられたことは、残念なことにOSとDBMとの差異は大きくDBMとしてのOSまたはその逆といったDBMとOSの強調、融合がこれから必要なのではないかと。空を飛ぶためには鳥にならなくても飛行機に乗れば良いわけで、いままでと異なったDBM型OSが出現する可能性もある。どちらにしろ空を飛ぶという目的は同じな訳でこれからも楽しみである。

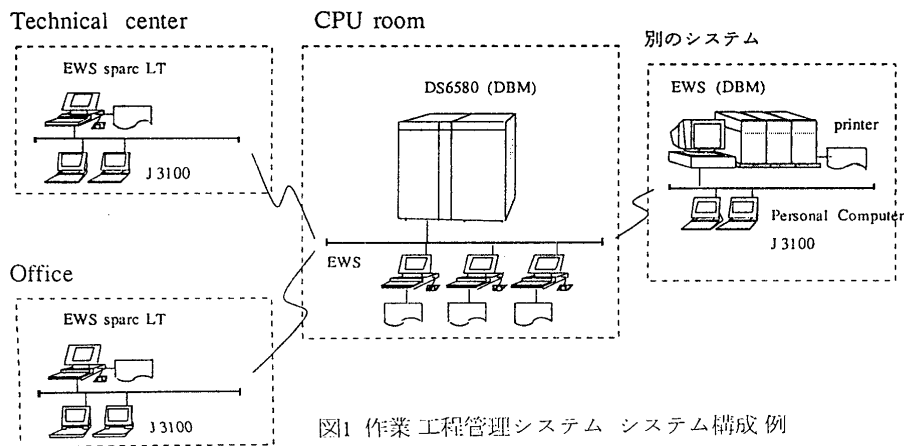


図1 作業工程管理システム システム構成例