

## DB 負荷テストの CSS 環境への適用

4 A C - 8

森田 靖<sup>1</sup>（日立エンジニアリング（株）） 高木 勝則（日立エンジニアリング（株））

矢野 理（日立エンジニアリング（株）） 中野 泰秀（日立エンジニアリング（株））

### 1. はじめに

CSS環境のパフォーマンスは、DBのレスポンスに大きな比重が占められる事が多い。特にクライアント数の増減、DBの更新系トランザクションの発行数により、そのレスポンスが左右される。

一方で、一般に公開されている性能値などは、特定モデルをベースとしたチャンピオンデータが多く、色々なケースを想定した実システムへの性能設計に対して活用が難しい状況にある。このため、CSS開発の現場では、現実のシステムに近い環境での事前評価が課題となっている。

今回、CSS環境において、状態の発生数、クライアント数、トランザクション数の遷移、トランザクションデータをDB負荷ファクタとした、CSSの応答性の評価環境を構築した。このDB負荷テスト結果により、CSS環境におけるパフォーマンスを事前に検討でき、実システムの性能事前検討が可能となる見通しを得たので報告する。

### 2. CSSの事前評価の問題点

CSS環境での性能は、DBのレスポンスが大きな比重を占めると考えられるが、各DBベンダーの出す、基準を定めての性能評価には、“トリック”を用いた（最高）性能値が用いられているとの報告もある。[1]

このため、実システムに近い形での評価が求められていた。しかし、評価環境がない、あるいは評価に手間がかかるといった理由から、事前評価を実施しないといった事も多い。この結果、受入検査直前になって応答性などの問題が顕在化した場合、工程混乱、ソフトの作り直し等による工数増大などの問題を誘発する

事になる。

また、最近よく言われている「小さく生んで大きく育てる」式の段階的に開発するケースでは、初期のシステム構成である程度の性能評価を実施できる。しかし、初期システムでの応答性は満足したものの、増設・改造によりクライアント数やレコード数等が増加する事により、パフォーマンスを左右するパラメータ（配列のサイズ、カーソル数など）がしきい値を越えてしまい、応答性が極端に低下するなどの問題が潜在している。

一方、シミュレーションによる評価では、ツールなどの準備が大変、あるいは専用の言語を用いてわざわざプログラミングしなければならないなど、本来の開発以外の開発を強いる事になる。

CSS開発の現場では、上記問題の現実的な対応が望まれており、開発システムの性能評価に関しては、「ある程度の精度でOKだが、スキルに関わらず簡単に行えるようにして、事前評価を必ず実施する事」が課題となっている。

### 3. 問題解決のアプローチ

CSS環境でのDBアクセスのパフォーマンスは、スキーマや利用するSQL文をはじめ通信オーバーヘッド、アクセスプロセスで消費するメモリ等種々の要因で決定されるが、最終的には、トランザクションの数とクライアント数（定常時及びピーク時）が重点ポイントと考える。

このため、事前評価としては、適用するDBMSに対し適用システムの代表的なスキーマをでDB構築し、もっとも多用されるSQL文を用い、トランザクシ

<sup>1</sup> Application of DB(DataBase) Load Test to CSS(Client Server System) Environment.

Yasushi Morita, Hitachi Engineering Co., Ltd.

ン数及びクライアント数他をパラメタとして、実際のCSS環境で性能測定を行ってみる必要があると考えた。これにより、かなり粗い精度ではあるが、実システムの性能を事前に評価可能である。

#### 4. DBパフォーマンス測定ツール

前記方針により、容易に性能評価できるような形の社内ツール、CSS向けデータベースベンチマーク測定ツールを開発した。

本ツールは、適用システムの標準的スキーマを持つDB（サーバー）に対して、ネットワークを経由し、クライアント側より、もっとも多用されるSQL文を繰り返し発行するものである。

評価のためのパラメータは、

- (1) 計測回数,
- (2) クライアント数,
- (3) 処理件数 (初期値と繰り返し数)

等がある。

これらの評価結果は、処理件数に対する処理時間として報告される。この報告は、汎用のスプレッドシートツールのファイルとなっているため、これを基に、応答時間の計算や事前検討報告書の自動生成にも役立てることができる。

図1は、本ツールの出力結果の1部である。

#### 5. 評価

本ツールを、生産管理システムで用いる標準モデルスキーマに適用し、評価を実施した。この結果次の効果が得られた。(本評価システムは、社内OA用LANを用い、かつDB操作は、DBへの単純登録・更新・削除・検索の4つのボタンであり、単位時間当たりの対象クライアント数、DBアクセス数を増減させるものとした。)

##### (1) アクセス方法の違いによる評価

性能問題を起こすポテンシャルが高い(ノウハウを活用しない)DB操作を本ツールにより、事前検出できた。

##### (2) 多重クライアントの評価

1サーバー当たりのクライアントの最適数(安全数)を検証でき、クライアント数からの分散DB設計(サーバー数)の指針となった。

このほか、作成したSQL文をそのままテストする機能をツールに盛り込む事を予定している。本機能を利用する事で、プログラマが作成したSQL文に対し、短時間かつ容易に性能の評価を行う事ができる。本ツールの適用は、初級プログラマからベテランと、幅広い構成スタッフを抱える大規模ソフト開発の現場においては、利用を100%義務づける事で性能に関する問題の削減が期待できる。

以上から、CSSパフォーマンスの各種検討に関して、DB負荷テストの有効性を確認する事ができたと考える。

#### 6. 終わりに

以上、CSS環境におけるDB負荷テストについて考察した。今後上記ツールを、実システム開発に対して適用し、有用性の本格的な検証を行う。

#### 7. 参考文献

- [1]岸田 一：クライアントサーバー型基幹業務システム設計の評価方法，情報処理 Vol.36,No.6,pp493-499(1995)

検索処理時間-1 (全件検索)

| 処理件数   | 平均処理時間 | 最速処理時間 | 最遅処理時間 |
|--------|--------|--------|--------|
| 4,000  | 10.92  | 8.33   | 13.55  |
| 8,000  | 19.87  | 16.5   | 24.41  |
| 12,000 | 31.05  | 29.73  | 36.62  |
| 16,000 | 40.68  | 37.29  | 44.48  |
| 20,000 | 49.31  | 46.06  | 53.77  |

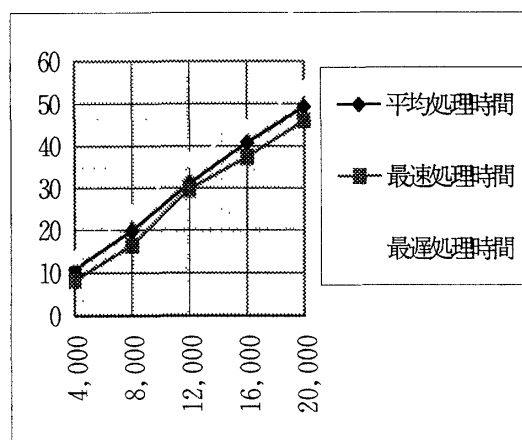


図1 DBベンチマーク測定ツールの出力結果