

イントラネットシステムにおけるデータ表示の高速化

1P-7

三菱電機株式会社 情報技術総合研究所

五十嵐 史生、青木 裕司

1. はじめに

データ表示システムとして、データサーバ、Webサーバ、表示端末の3階層で構成されるイントラネットシステムが普及しつつある。しかし、図1のようにデータサーバとWebサーバ間が低速回線で接続されている広域分散システムの場合、表示データが多くなるとデータベースの検索と検索結果の転送に時間がかかり、表示が遅くなるという問題がある。この問題を解決するために、Webサーバに表示頻度の高いデータを蓄積して、表示を高速化する方式を試作検討した。

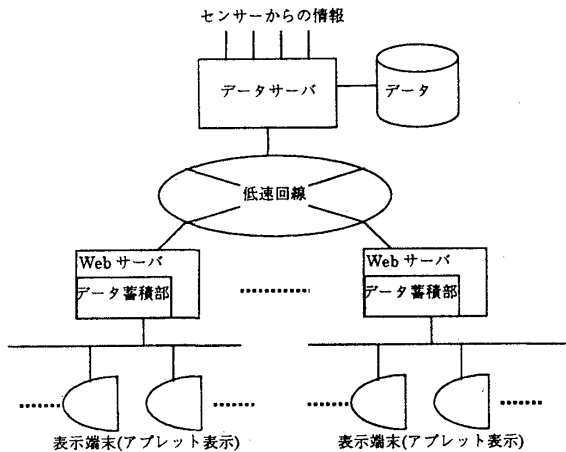


図1 3階層分散型イントラネットシステム

2. データの蓄積方法

表示端末に表示するデータのうち、HTMLファイルなどの固定データはローカルにWebサーバを設置することで高速化できる。しかし、データサーバのデータは一定周期で更新されているので、常に最

新のデータを取得する必要がある。このため表示端末では、Java アプレットが動作してデータを取得表示する。したがって、表示要求のたびにデータサーバから低速回線を経由してデータ取得することになり、表示低速化の原因となる。

そこで、Webサーバ内にデータを蓄積するデータ蓄積部を置き、データサーバのデータのうち、最新データなどの最も頻繁にアクセスされるデータを選択して蓄積することにより、データサーバへのアクセス時間を短縮する方式を検討し試作を行った。

蓄積するデータは一定周期で計測されるデータに限定し、データサーバに最新データが格納された場合は、データサーバからの通知をもとにデータ蓄積部が最新データの取得を行い、データサーバ、データ蓄積部間のデータに差異が生じないようにする。

なお、データ蓄積部は、移植の容易さ、表示端末アプレットとのモジュール共有化を考慮し、Javaで作成した。

3. 性能評価

3.1. 測定環境

この測定を行ったマシン構成は以下の通りである。

- データサーバ： Pentium Pro 200MHz、Windows NT 4.0 Server
- Webサーバ： MMX Pentium 200MHz、Windows NT 4.0 Workstation、jdk1.1.6
- 表示端末： MMX Pentium 200MHz、Windows NT 4.0 Workstation

なお、低速回線には64kbps回線シミュレータを使用した。

3.2. 測定方法

測定方法は以下の通り。

- Java アプレットを起動し、5KBのデータを取

The method of rapid data indication at intranet systems.

Fumio IGARASHI, Yuji AOKI
Information Technology R&D Center
Mitsubishi Electric Corporation

得して表示を完了するまでの時間。

測定は、事前にクライアントブラウザの JavaVM を起動しておき、一旦 Java アプレットを読み込んだ後に行った。

また、データ蓄積部である Java アプリケーションの負荷を測定するため、複数端末から同時に表示要求をした場合の表示速度の変化とデータ蓄積部の CPU 占有時間も測定した。CPU 占有時間は Windows NT パフォーマンスモニタを用いた。

3.3. 測定結果

測定は手動で複数回行い、平均値をとった。

表 1 に表示性能測定結果を示す。

表 1 表示性能測定結果

	蓄積あり	蓄積無し
表示性能	3.3 秒	11.7 秒

表 1 に示すように、データを Web サーバに蓄積することで、3 倍以上の表示性能を見込むことができることが分かった。

また、複数端末同時要求時の負荷測定結果を以下に示す。

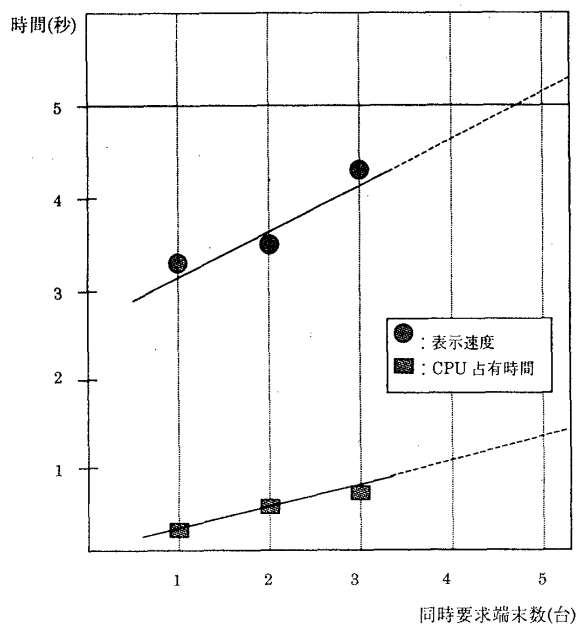


図 2 同時要求時の負荷変化

図 2 に示すように 1 台あたり約 0.5 秒表示端末の表示速度が遅くなり、約 0.2 秒データ蓄積部の処理が増加することが分かった。このことから、Web サーバ上でのデータ蓄積部の負荷は表示端末の負荷より小さいと言える。ただし、多端末時には、データ蓄積部の負荷も無視できなくなる。例えば、測定環境の場合では 20 台同時に表示要求を行った場合にデータ蓄積部の処理時間は 4 秒となる。

4. おわりに

本技術を適用することによって、データ表示システムにおける低速回線というボトルネックを解消し、低速回線を使用しないシステム構成時とほぼ同等の性能を実現できることがわかった。次のボトルネックは、①端末側の表示性能、②Web サーバのデータ蓄積部の負荷と考えられる。以下、各々の対応策を示す。

① 端末側の表示性能

高速な機種を使用することが最も容易であるが、費用面から考えると実現性に欠ける。端末側は、イントラネットシステムの利点を生かすため、標準モジュールのみで構成すべきである。したがって、対応策としては、SuiteSpot などといった最新 Java プラグインモジュールなどの適用や Java アプレットの高速化検討などが考えられる。

② Web サーバのデータ蓄積部の負荷

高速性、信頼性、スケーラビリティの観点から、サーバを多重化する方法が考えられる。その場合、低速回線の負荷を増やさないために、データ蓄積部のデータレプリケーションなどの開発を検討する必要がある。

今後はこれらの課題を検討していく。