

分散システムの新技术

マルチベンダ環境での分散システムの実現を目指す

A New Technology of Distributed System

To realize a distributed system in a multi vendor system environment

(電力技術研究所 情報ネットワークG)

情報処理の分野では、インターネットの急速な普及に伴い、インターネットのソフトウェア技術も著しい発展を遂げている。これらの技術の中でも特に有望とされているJava技術は、まだ発展途上ではあるが、現在の分散システムの根本的な問題点を解決する機能を有しており、その適用が期待されている。

そこで、この技術を取り入れたプロトタイプを開発し、機能、レスポンス性能、生産性、保守性等の観点から評価し、業務への適用可能性を検証した。

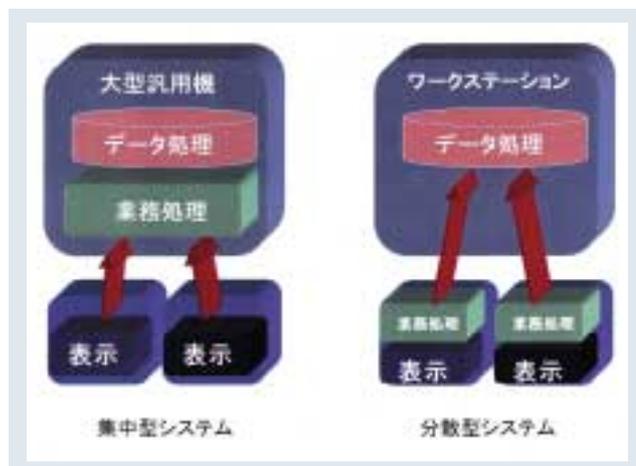
1 研究の背景と目的

当社のシステム開発形態は、従来の大型汎用機中心の集中型システムからワークステーション・パソコン中心の分散型システム（以下、クライアントサーバシステム）が主流となりつつある。（第1図）

このクライアントサーバシステムでは、安価なハードウェアの利用、ユーザに優しい操作性（グラフィカルユーザインタフェース）といったメリットがあり、急速に普及してきている。しかし、クライアントの台数が増えてくると、以下のような運用管理面でのデメリットが問題となっていた。

プログラム開発・変更時に、すべてのパソコンにプログラムを配布しなくてはならない。

プラットフォーム（OS）が変更になるとプログラムをすべて変更しなくてはならない。



第1図 集中と分散

(Information and Network Group, Electric Power Research & Development Center)

In the field of information processing, software technology for the Internet is advancing conspicuously along with the rapid expansion of the Internet itself. The Java technology, which is still being developed, is considered the most promising of such software technologies. This technology is believed to be able to solve the basic problems of current distributed systems, and much hope is being placed on its application.

Thus, we have developed a prototype system incorporating this technology and verified the possibility of its practical application from the viewpoints of function, response performance, productivity, maintainability, etc.

2 研究の概要

本研究では、この問題点を解決する技術として、インターネットで発展してきたオブジェクト指向プログラミング言語であるJavaに着目した。Javaには、プログラミング言語仕様の他に以下の大きな特徴がある。（第2図）

実行時にサーバからプログラムが転送されるため、クライアントにプログラムの配布が不要。

どのプラットフォーム（OS）上でも同じプログラムで動作可能（実行環境でプラットフォームの違いを吸収）であり、プラットフォーム変更時にもプログラムの変更が不要。



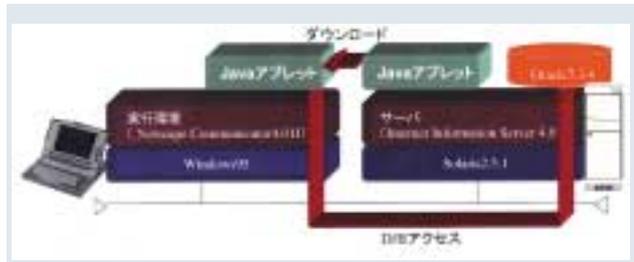
第2図 Javaの特徴

Java技術は、急激に技術革新が進んでいるが、まだ発展途上であるため、導入にあたっては慎重な技術的評価が必要である。そこで、Java技術を使ったプロトタイプを作成し、業務適用時に必要と思われるポイントを中心に、この新しい技術の評価した。

3 プロトタイプの概要と実証試験結果

(1) プロトタイプシステム構成

今回開発したプロトタイプは、研究計画書データの入力を想定しており、システム構成は、下図の通りである。(第3図)



第3図 プロトタイプのシステム構成

(2) 評価結果

画面の見え方

プロトタイプでは、プラットフォームごとに多少画面表示に違いのあることがわかった。(第4図 / 表示されるはずのスクロールバーが表示されない例)

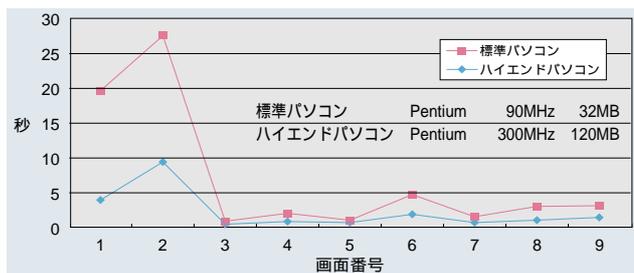
そこで、現在提供されているソフト部品(スクロールバー、テキストエリア等)全てについて画面表示を調査した結果、ソフト部品とプラットフォームの組み合わせによって、見え方の異なることを明確にできた。



第4図 プロトタイプの画面例

レスポンス性能

標準パソコンとハイエンドパソコンを使用し、レスポンス時間の違いを確認した。(第5図)

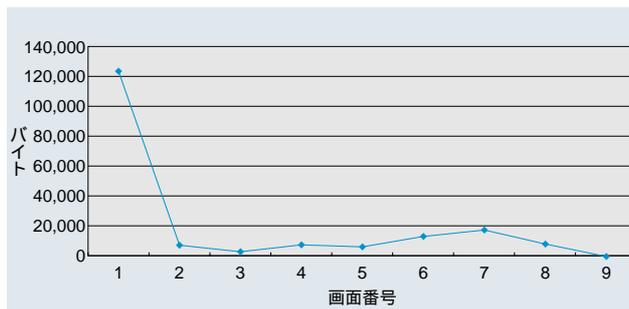


第5図 レスポンス比較表

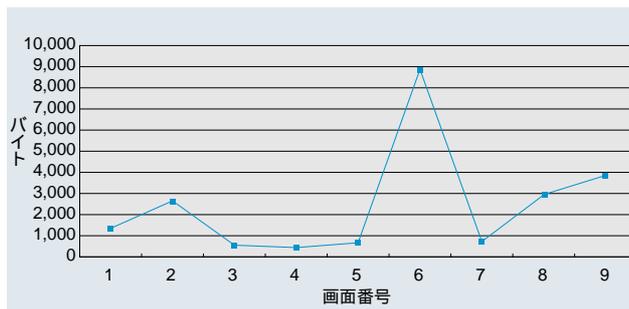
ボトルネックの追跡調査

レスポンス時間の遅い画面については、それがプログラム転送と処理によるものか、あるいはデータ転送によるものかを判別するために、通信トラフィック量を計測した。(第6図、第7図)

この結果、転送されるプログラムのトラフィック量とレスポンス遅延時間との相関関係が明らかになった。また、転送されるデータ量を計測することにより、画面の作りによって転送データ量に違いがある点とその転送データ量がレスポンス性能へ影響することが判明した。



第6図 トラフィック表 (転送されるプログラム)



第7図 トラフィック表 (転送データ量)

4 効果と今後の展開

分散システムの新しい技術であるJavaをプロトタイプで評価して、プログラムの転送と処理によるレスポンス低下への影響、および画面設計時にレスポンス低下を避けるために必要とされる事項を抽出するとともに、Java技術を使うことにより、クライアントへのプログラム配布が不要で、かつ、どのベンダのプラットフォーム上でもプログラムの変更なしで、移行することが可能なことを検証できた。

この研究成果をもとに、ラインではさらに実業務を対象とした分散システム技術の評価が行われ、その結果実業務に適用可能と判断されて、平成10年度開発件名に採用されている。

今後は当社情報システムの標準的な開発技術として取り入れられることになっている。