イントラネット上の分散並列処理

5U－8

土田　明美　松下システムエンジニアリング株式会社

上田　謙一　松下通信工業株式会社

1. はじめに

イントラネット上には高性能なマシンパワーを持ちながら、実際にはその資源を有効利用されないままになっているサーバが多数存在する。本研究ではそのような資源を利用して並列処理を行うシステムの開発を試みた(図1参照)。このシステムは通常の並列コンピュータと違い、複数の既存のサーバを利用し、それらをネットワークを介して並列化させている。本稿ではこのシステム開発の現況について報告する。

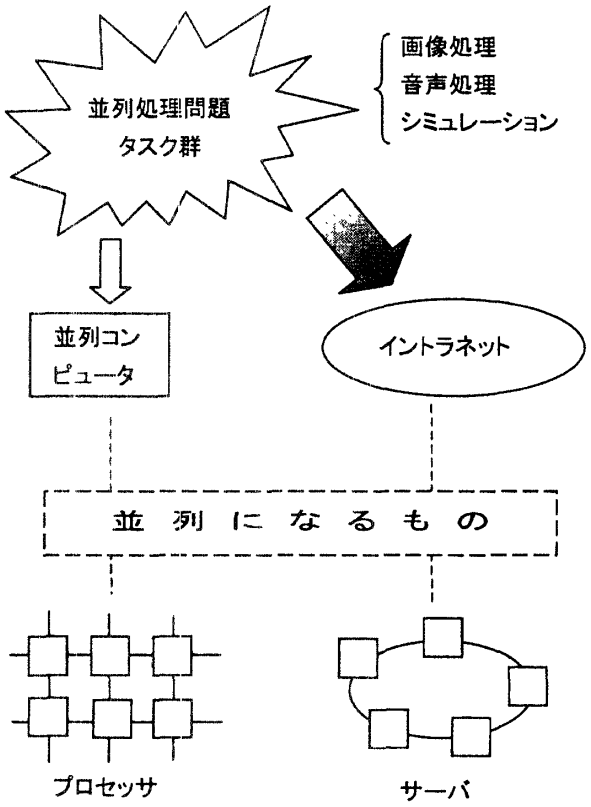

図1　ネットワーク上の並列処理

2. システム概要

このシステムでは、クライアントから既存のサーバにタスクを渡し、サーバからさらに他のサーバへタスクを分配して並列処理し、その最終結果をクライアントに返すという実行の流れになる(図2参照)。これをどうやって実現するかが本研究の課題となった。

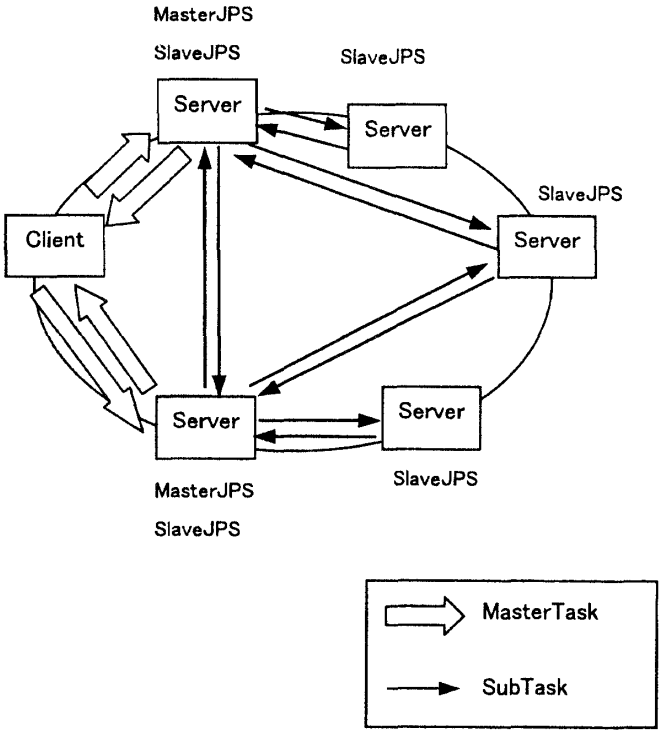

図2 システム構成

3. システム方式

1)マルチプラットフォーム環境

まず、このシステムの開発言語は Java を使用することにした。Java を選んだ理由としては、ネットワーク使用に適した言語であるということ、マルチプラットフォーム言語であるということがあげられる。イントラネット上のサーバにはいろいろな種類のものがあるし、クライアントにも同様のことが言えるので、一つのプログラムがどのマシン上でも実行可能な Java 言語はこのシステム開発に最適であると考えた。

Parallel Processing on Intranet (PPI)
Kenichi Ueda
Matsushita Communication Industrial Co.,Ltd.
Akemi Tsuchida
Matsushita Systems Engineering Co.,Ltd.

2)プログラム配信

クライアント側の GUI には Java Applet を使う。そして、サーバ上で並列処理させるタスク自体も Java プログラムとする。ここで問題になるのは Java プログラムをクライアントからサーバへ渡して実行させるというその方法である。サーバからクライアントへ Java Applet をダウンロードしてブラウザ上で実行させるというのは Java のよくある使用形態であるが、今回の場合はその逆であるので、そのための仕組みが必要になる。そこで我々が考えたのが Java Protocol Server(JPS)である。これはJavaプログラムの実行を司るモジュールであり、これをサーバにインストールしておくことにより、サーバはクライアントから Java プログラムを受け取り、実行し、結果を返すという処理ができるようになる。

3)タスクの分配、実行

さらに、並列処理を行うために、サーバからサーバへの並列タスクの分配をする必要があるので、JPS を MasterJPS と SlaveJPS という2層構造にする。MasterJPS はタスクの受け取り、分配、並列タスクどうしの同期取り、結果をクライアントに返す、といったタスクの管理のみを行い、SlaveJPS は MasterJPS から受け取ったタスクの実行のみを行う。これらの2つの JPS をそれぞれのサーバにインストールしておき、どのサーバとも Master にも Slave にもなれるようにする。

4)タスクの定義

実行タスクの並列化は、それぞれ単独でも実行できるタスク(SubTask)をいくつも組み合わせることによって1つの大きなタスク(MasterTask)にするという方法で実現する。それぞれのタスクはパラメータを持つことができ、タスクが SubTask を持つならば、SubTaskの結果をパラメータとして受け取る。タスク構造例を示す(図3参照)。

5)同期・繰り返し

MasterJPS はタスク(MasterTask)をクライアントから受け取ると、その中から実行可能な SubTask を抽出して SlaveJPS に渡す。SlaveJPS からそれぞれの結果が返ってくれば、それらの同期を取り、さらにその上位タスクの実行が可能になればそのタスクをSlaveJPSに渡す、といったことを繰り返し、最終的な結果が出ればクライアントに返す。また、それぞれのタスクは自分自身の結果をパラメータとして取り、実行を繰り返すこともできる。

6)タスクの割付

クライアントからMasterJPSへのタスクの割付は、使用可能なサーバに均等に行う。MasterJPS から SlaveJPS への割付は、タスクの実行に必要な資源とサーバの資源とを突き合わせて、適切なサーバに割り付ける。

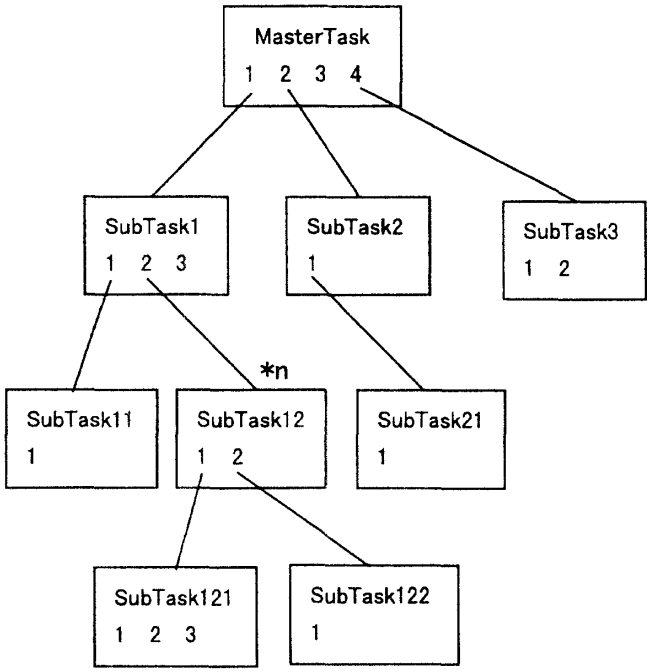

図3 タスク構造

4. まとめ

このシステムにより、イントラネットという環境において、サーバ上の情報を皆で共有する、利用するという方向とは逆に、クライアントから複数サーバを利用して仕事を行わせることが可能であるという証明ができたと考えている。

また Java 言語の持つ特性、すなわちネットワークコンピューティングを初めから想定して開発された言語ゆえの特性を生かしたシステムの開発ができたと考えている。

このシステムに今後も改良を加えていき、実用化できるようになれば、クライアントは時間のかかるプログラムの実行を自分で行わず、イントラネット上のどこかのサーバ上で実行させておくことが可能になる。これは資源の有効活用という意味でも有意義なテーマではないだろうか。