

# トークンを用いた分散資源管理システムについて

4 F - 3

住元 真司  
富士通研究所

1: 始めに

WSとネットワークの普及によって、これらを有効に利用するための分散処理技術が重要になっている。この技術の中でも分散ファイルシステムを頭とするデータの共有と、CPU、メモリなどのもっと幅広い計算機資源の共有を目指した分散資源の共有のための技術は特に重要である。

我々はこれまでに後者の中でもCPU能力の共有について、分散makeを取り上げ、その実現性と効果について確認し、研究を進めてきた。(参考文献1)を参照)

ここではこれを更に進めて、分散環境における資源共有の方式について、高速ネットワーク環境上で、速く、安全に計算機資源を共有可能な資源管理方式の実現を目的として、分散環境上での資源管理の性質と課題を述べ、この課題を解決するために採用したトークンベースの資源管理システムについて説明する。

2: 分散環境の性質と資源管理システムへの課題

広範囲にネットワーク接続された計算機群から構成される分散計算機環境においては、様々な種類の計算機が存在し、様々な位置に計算機が存在する。このような環境において複数のユーザが処理を行なうわけである。

このような複数の計算機を共有する環境においては、ユーザの数や行なう処理によって計算機の状態が時々刻々と動的に変化する。更に、ネットワークで接続されているために、通信遅延の存在が問題になる。

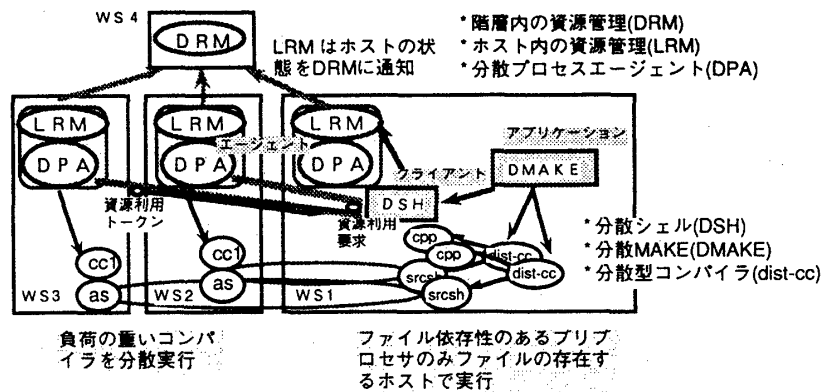
このような環境において、ネットワークに接続された計算機資源を有効に利用するための資源管理システムの実現のための課題として、以下を重要な項目として取りあげた。

- 1)様々な能力を持つ計算機間の能力差をうまく利用するための機構
- 2)分散環境上で資源のローカリティを活かした資源配分を実現する機構
- 3)複数のユーザに対して公平に資源利用を実現するための機構
- 4)計算機の状態の変化に動的に対応可能な機構
- 5)起こるであろうフォールトに対する対策
- 6)通信遅延フリーの機構

3: トークンベースの資源管理システムについて

以上のような課題を解決可能な資源管理システムを検討した結果、図1のモデル上で資源管理システムを構築することにした。

ここでは、主に分散資源管理システムを利用した分散シェルと分散makeについて述べることにする。なお今回のシステムで取り扱っている資源はプロセス資源であるが、原理的には広範囲に適用可能である。



3-1: 分散資源管理システムの構成

全体のモデルは階層構造内の資源管理を行なうDRM(Domain Resource Manager)、ホスト内の資源管理を行なうLRM(Local Resource Manager)、階層構造内のサービスを司るエージェント(ここではDPA(Distributed Processes Agent))と、サービスを受けるクライアント(ここでは分散シェルDSH(Distributed Shell))、分散シェル上のアプリケーションであるDMAKE(分散make)、分散型コンパイラから構成される。

図1 分散資源管理システムのモデル

このモデルの特徴は、コネクションベースの資源割り当て、トークンベースの資源利用機構にある。資源利用の主導を分散先(実際に資源の存在するホスト)とすることにより、動的に変動する資源状態への対応、通信遅延による資源の多重割当ての回避が可能になる。

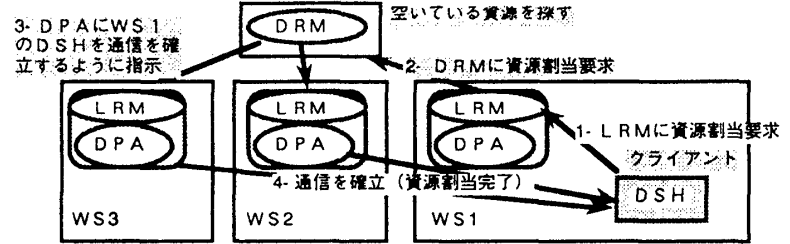
資源利用の動作としては、エージェントとクライアント間でコネクションが確立することにより資源（サービス）が割り当てられ、エージェントが接続されている特定のコネクションの一つにトークンを送出し、クライアントがこのトークンを受けとることにより、実際に資源（サービス）の利用が可能になる。（図2）

このトークンベースの資源利用を効率的に行なうために、分散資源を利用するクライアントでは、1非同期に受信するトークンに呼応した動的なスケジューリングを行なうことを前提にしている。動的なスケジューリングを行なうことによって、動的に変化する資源の状態による影響を最低限に抑え、その時の利用可能資源内での最適な処理が可能になる。

更に、様々な能力を持った計算機を取り扱うためにトークンに資源状態を付加している。クライアントはこの情報も合わせて動的に資源の利用を行なうことになる。

このような構成とすることにより、複数のクライアントで資源を共有する場合にもエージェントにおけるトークンの送出手の制御によって資源利用を抑えることができ、十分に対応が可能である。

◇資源割当時の動作（分散シェル）



◇資源利用時の動作（分散メイク）

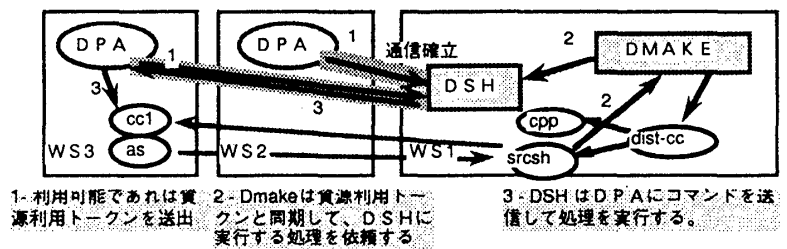


図2 資源割り当てと資源利用

3-2: フォールトへの対策

分散環境では、マシンのダウンなどのフォールトの発生を前提に考える必要がある。我々は2つの面からアプローチを考えている。一つはホストの状態変化の予測によりできるだけ安全な資源を割り当てること、もう一つは状態が変化（ホストダウンなど）した場合にどのようにリカバリを行なうかである。

前者については、我々の考えている分散環境においては、例えばユーザの利用しているWSなどはユーザ利用の仕方により資源利用の比率は著しく異なる。（CADやシミュレーションでは常時100%CPUを利用しているのに対して、メールなどの利用者はほとんどCPUを利用していない。）こういった利用の違いを統計的に集計して予測に反映している。

後者については、分散資源管理システムではエージェントなどのダウンの検出と、遂行されなかった要求をクライアントに対して伝えるのとどめ、処理はクライアントに任せている。例えば分散makeの場合では、その後のリカバリは分散makeコマンドによって行なわれる。

4: 実装

実装は既存のUNIX OS上でソケットを利用してユーザプロセスにて実装している。OSへの改造は一切していない。RPCなどを利用すると利用できるマシンに制限ができるために移植性を考慮してこのようにしている。

現在のところ利用可能なマシンはDS (UXPS/SVR4)、SPARC搭載のSunWS/サーバ (SunOS/4.x.x)、Sun-3 (SunOS/4.x.x)、A-60 (4.3BSD, MACH) である。

サーバ間の処理手順プロトコルを実装する際に制御用のライブラリを作成した。このライブラリにより実装の効率が飛躍的に増加した。

分散シェルの機構は参考文献1)で作成した分散型のコンパイラをベースに拡張して実現し、分散シェルのアプリケーションである分散makeコマンドではトークンと同期して資源利用を行なうために動的なスケジューリングを実装している。

5: 終りに

今回、分散環境上での資源管理システムとしてトークンをベースにした資源管理システムを提案した。今後、更にこの方式の効果、及び、適応性について研究をすすめていく。

最後に、この研究を行なうにあたり、いろいろと助言を頂いている富士通研究所 津田 光コネク ト推進室 システム研究部長 並びに DS上での開発、評価にあたり御協力を頂いている 富士通 オープンシステム事業部 第1開発部の方々に感謝致します。

6: 参考文献

- 1) 住元、湯原、岸本：「分散makeに関する考察」 1991年11月  
B-ISDN時代におけるマルチメディア通信と分散処理シンポジウム
- 2) M.Yuhara, S.Sumimoto :  
Works in progress session, SEDMS III, 1992.