

ソフトウェア自動分割システム RODS における
ネットワーク・マネージャの開発

5L-3

○高橋透 福田享子 松山実 横井利彰

武蔵工業大学

1. はじめに

近年、ネットワーク環境が普及してきたが、この環境を効果的に利用するソフトウェアの開発は容易でない。現在、ネットワーク環境で動作するソフトウェアの開発を支援するシステム RODS⁽¹⁾ (Repetitive Optimum Dividing System)の開発を行っている。本報告では、RODS内のネットワークを介した際のデータ転送を受け持つネットワーク・マネージャについて述べる。

2. RODS の構成

RODS上では、ソフトウェアはNDFD(Network Data Flow Diagram)という独自のデータ・フロー・グラムを用いて記述する。NDFD エディタにより NDFD 表記で記述されたソフトウェアは NDFD ソースとして出力される。

自動分割システム (Automatic Dividing System)が NDFD ソースを分割し、ネットワーク上の各マシンへ分配する。自動分割システムが NDFD ソースを分割したものを NDFD ピースと呼ぶ。

各マシンへ配送された NDFD ピースは、NDFD 処理系により実行可能コードに変換される。

各マシン上で実行されたソフトウェアは、その実行履歴が記録される。履歴解析システム(Log Data Analyzer)は、この実行履歴を解析し、解析データを自動分割システムに送る。

自動分割システムは、解析データを基に再分

割・再配送を行う。これを繰り返すことにより、実行時間の短いより最適な分割へと近づけることができる。

RODS の構成を図1に示す。

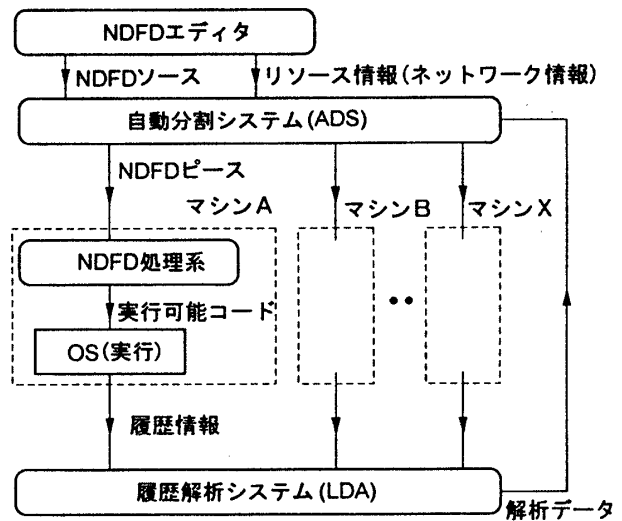


図1 RODS システムの構成

3. NDFD 処理系

NDFD 処理系は NDFD ピースから実行可能コードを生成するものであり、NDFD コンパイラと C 言語処理系により構成される。NDFD コンパイラは NDFD から C 言語ソースへのトランスレータである。

NDFD 処理系は以下の4つのマネージャと呼ばれる。C 言語の関数群を必要とする。

1. メイン・マネージャ

他のマネージャの管理(起動, 初期化, 終了処理)を行う。

Development of Network Manager for RODS

Toru TAKAHASHI, Michiko FUKUDA, Minoru MATSUYAMA, Toshiaki YOKOI

Musashi Institute of Technology

2. データフロー・マネージャ (図2)

ノード間のデータ転送を受け持つ。

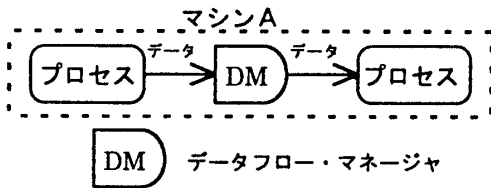


図2 同じマシンのプロセスにデータを送る場合

3. ネットワーク・マネージャ (図3)

メイン・マネージャ、データフロー・マネージャがネットワークを介した際のデータ転送を受け持つ。

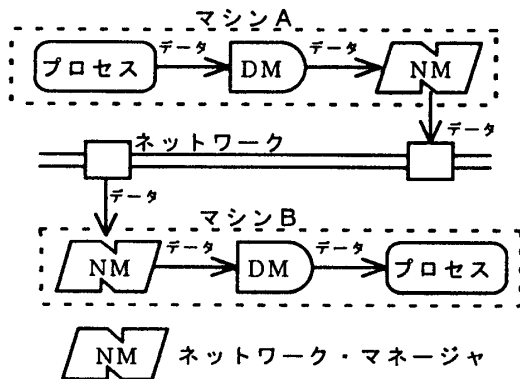


図3 違うマシンのプロセスにデータを送る場合

4. ログ・マネージャ

データフロー・マネージャとネットワーク・マネージャの実行状況を記録する。記録は履歴解析システムにより解析される。

4. ネットワーク・マネージャ

ネットワーク・マネージャは、メイン・マネージャ、データフロー・マネージャの動作のうち、ネットワークに関する処理を受け持つ。

使用者の端末上のメイン・マネージャによる他のマシン上のメイン・マネージャの起動処理、ネットワークアドレス等の情報の取得、端末のメイン・マネージャと他のマシンのメイン・マネージャとのネットワーク接続処理、データフロー・マ

ネージャによるマシン間のデータ転送処理を行う。

さらにネットワーク・マネージャは、ネットワーク上のコンピュータの情報を管理する。データフロー・マネージャがデータを転送する際、実行対象プロセスが2台以上のマシンに存在し、どちらでも実行が可能である時には、その時点で負荷の軽いマシンを選択して、マシン間のデータ転送を行い、ソフトウェア動作の早期終了を図る。

ネットワーク・マネージャは、TCP/IP プロトコル群^[2]とバークレーソケットを用いたプロセス間通信によって実現される。

現在ネットワーク・マネージャは、データグラム型ソケットによってデータ転送を行う。しかしシステムコール sendto/recvfrom を用いるとソケットのバッファの大きさに制限がある。この制限を超えたデータの転送を行うと、過剰部分が消失することがある。ネットワーク・マネージャはこれを回避するため、バッファの大きさを越えるデータに対して、幾つかのブロックに分割して転送し、受信マシン側で連結するという手段をとる。更にコネクションレスのプロトコルは、データの流量制限の機能を持たないため、バッファ不足が起きる可能性もある。ネットワーク・マネージャはこれが起きないようにするための流量制御も行う。

5. まとめ

今後、効率のよいデータ転送方法を得るため、いくつかのプロトコル・パイプを試行すると共に、ネットワーク・マネージャの機能である、データ及びファイルの転送・受信、ネットワークデータの取得、マネージャの起動、プロセスの選択処理を完成する、また他のマネージャと融合を図り、RODSの完成をめざす。

参考文献

[1] 山下利夫, 松山実, 横井利彰, “自動分割システム RODS における NDFD 処理系について”, 情報処理学会第46回全国大会, 講演論文集(分冊5), pp.261-262, 1993

[2] W. リチャード・スティーヴンス/篠田陽一訳, “UNIX ネットワークプログラミング”, トッパン, 1992