

## ソフトウェア自動分割システムRODSの概要

2U-10

山下利夫 松山実 横井利彰

武蔵工業大学 工学部

## 1. はじめに

近年、ネットワーク環境の普及と共に様々なコンピュータが相互に接続され、分散システム上で動作するソフトウェアシステムの需要が高まっている。分散システムは、ネットワーク上に分散配置されたプロセス間で通信を行うことにより、目的の処理を行う[1]。しかし現存のネットワーク環境で分散システムを実現するには、多大な労力が必要である。

ネットワーク上に分散配置される各プロセスの設計は、スタンドアロンで用いられるソフトウェアに比べ、多くの困難を伴う。プロセスは各マシンで並列的に実行され、ネットワークを通じて協調して動作するからである。また効率的な分散システムを設計するには、対象となるネットワークについての詳細な知識が必要になる。例えば、平均した能力を持つマシンで構成されたネットワーク環境に、高速なマシンが1台導入されただけで、最適な分散システムの設計は大幅に変わる。

ソフトウェア自動分割システムRODS (Repetitive Optimum Divising System)は、現存するネットワーク環境で分散システム設計を行う際に生ずる、特有の困難さを解決するためのシステムである。本稿では、RODSの概要について述べる。

## 2. RODSの概要

分散システムを設計する際、目的とする処理をどのようなプロセスに分割するかが問題である。対象とするネットワーク環境に最適で、効率の良い分割は難しい。この分割を自動的に行うことができれば、その困難さが解決されると考えた。

武蔵工業大学では以前から、CASE技術[2]をベースとした、図形を用いた分散システムの記述方式の研究を行ってきた。この研究に自動分割の概念を取り入れたものがRODS

プロジェクトであり、現在3つの部門に分かれて研究を進めている。図形を用いた分散システムの記述方式、自動分割システム、および分割最適化システムである。

## 2.1. 図形を用いた記述方式

分散システムの設計上最も重要なのは、プロセス間を行き来するデータの流れである。そこで処理単位間のデータの流れに主眼をおいたダイアグラム、DFDを拡張して用いている。

拡張されたダイアグラムを、NDFダイアグラム(NDFD)と呼ぶ。コンピュータ上での編集を考慮した表記の変更[3]、幾つかの制御構造の付加、移植性を保証するためのライブラリのレベル付け等の変更を行なった。

また、簡単なNDFD処理系を試作し、NDFDから自動的にC言語のソースコードが生成可能なことを確かめた。

現在、自動分割を考慮した表記の見直しを行なっている。それと同時に、Xウィンドウ上で動作するNDFDエディタの製作を進めている。

NDFDの例として、Xウィンドウ上で動作するxloadの、最上位のダイアグラム(コンテキスト・ダイアグラム)を図1に示す。xloadは、マシンの負荷状態をグラフ化して表示するツールである。

## 2.2. 自動分割システム

自動分割システムは、対象とするネットワーク環境に関する情報を記述したファイルが必要とする。そしてNDFDで記述されたソフトウェアを解析し、それぞれのマシン用のNDFDに分割する。分割されたNDFDはそれぞれのマシン用のC言語のソースに変換され、コンパイルされる。

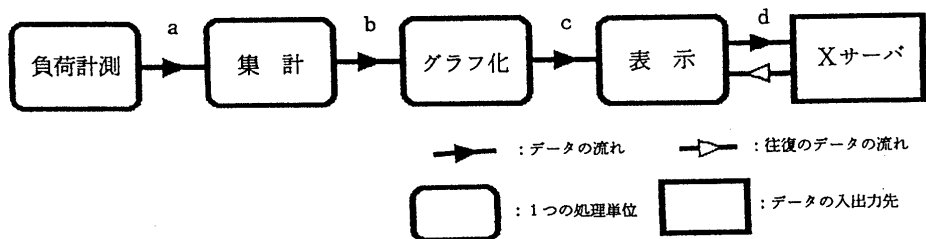


図1. xloadの構成

図1のxloadを用いて、ネットワーク上の他のマシンの負荷状態を知りたいとする。流れるデータ量やネットワークの転送速度を考えると、aの位置で分割するのが適切である。図1においては、右にいくほど送られるデータ量は増大しているからである。ところがxloadでは、dの位置でデータの転送が行なわれており、最も効率の低い設計である。

また分割時に1つの処理を、複数のプロセスに重複して含ませることができる。重複した処理は実行時に最も処理が軽いプロセスが引き受け、動的なスケジューリングが行なわれる。重複の割合は任意に決定でき、重複を禁止すれば完全な静的スケジューリングとなり、無制限にすれば完全な動的スケジューリングとなる。また全ての処理が必ず2つ以上のプロセスに存在する設定もあり、この場合はどのマシンがダウンしても処理の継続が可能となり、フォールトトレランスが備わる。

### 2.3. 分割最適化システム

自動分割システムが必要とするネットワーク環境に関する情報は、正確なものを用意するのは難しい。また、各マシンの負荷等はあくまで平均値でしかなく、NDFDの解析結果も、期待値でしかない。またネットワークの構成や負荷状態も、変化する。

そこで必要となるのが、分割最適化システムである。最初はまず不完全な情報のみで、自動分割を行ない、実際に使用していく間に、実行履歴を保存しておく。一定期間後に履歴



図2. NDFDの分割過程

を集計し、その情報を基にソースであるNDFDを再分割する。この過程を図2に示す。

一定期間ごとに履歴のチェックと再分割を行なうことにより、より最適な分割を行なう。これにより、ネットワーク環境の変化にも対応できる。

### 2.4. まとめ

以上3つのシステムにより構成されるRODSにより、分散システム上で動作するソフトウェアシステムが容易に開発できると考えられる。対象とするネットワークの環境に左右されず、目的とする処理のみを記述することでソフトウェアを作成することができる。またNDFDで記述されたソフトウェアはそれぞれのネットワーク環境に合わせて自動分割され、定期的に最適化されることで、効率的に実行される。

### 3. おわりに

現在、Xウィンドウ用のNDFDエディタを開発中であり、また自動分割のためのNDFD解析方法を研究中である。

### 参考文献

- [1] 早田弘一, 安在弘幸, 山之上卓, “図を用いた分散システム記述言語と分散システム自動生成システム”, 情報処理学会第44回全国大会講演論文集(分冊5), pp.81-82, 1992
- [2] A. S. Fisher, “CASE”, 共立出版, 平成2年
- [3] J. マーチン, “自動システム設計のための標準ダイアグラム作成技法”, 近代科学社, 平成3年