

ソフトウェア自動分割システム RODS における 履歴解析手法

6M-2

○福田享子 高橋 透 松山 実 横井利彰
武蔵工業大学

1 はじめに

現在我々が開発中のソフトウェア自動分割システム RODS^[1] (Repetitive Optimum Dividing System) は、ネットワーク環境で動作するソフトウェア開発を支援するシステムである。本システムの構成を図1に示す。

RODSで開発したソフトウェアを各コンピュータ上で実行すると、そのソフトウェアの実行履歴がファイルに記録される。履歴解析システムでは、この実行履歴を一定期間ごとに解析し、より確かな負荷の期待値やソフトウェアの分割が適切に行なわれていない部分を修正の必要な箇所として出力する。この解析結果は自動分割システム^{[2][3]}で参照され、ソフトウェアの再分割が行なわれる。そのためRODSでは、ソフトウェアの実行を重ねるごとに、より効果的な分割が行え、ネットワーク環境の変化にも対応することが可能となる。

今回は、履歴の解析手法に、しきい値を用いた場合とファジィを用いた場合について述べる。

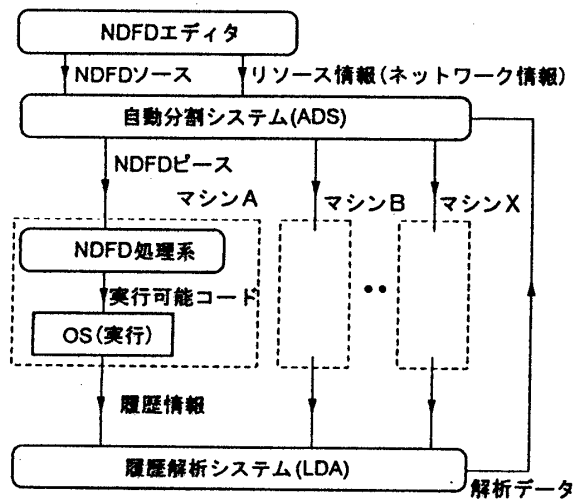


図1: RODSの構成

Analytical method of LDA (Log Data Analyzer) used in RODS
 Michiko FUKUDA, Toru TAKAHASHI, Minoru MATSU-
 YAMA, Toshiaki YOKOI
 Musashi Institute of Technology

2 履歴解析システムの概要

まず、ソフトウェアの実行履歴としてファイルに記録されている履歴内容^[4]と、それらの履歴から解析できることを以下に挙げる。

- データが全て揃うまで及び揃ってから処理が開始されるまでの時間

これらの処理に時間がかかるということは、そのノードで処理が滞っているということである。そこでノードごとに処理の滞る割合を求め、その原因をコンピュータの負荷と共に解析し、以下の解決策に対応するデータを出力する。

- A: 分割の見直し
- B: ノードの共有化分割
- C: コンピュータの性能評価の見直し

- ノードの実行時間

最初から正確な負荷の期待値を用意することは困難である。そこで、各ノードごとに平均実行時間を求め、この値を自動分割システムが分割する際に参照する各ノードの負荷の期待値とする。

- 共有化分割したノードの実行コンピュータ

共有化分割(複数のコンピュータに実行可能な同一ノードが存在するような分割)されたノードには、実行される確率の最も高い代表ノードというノードがあらかじめ設定されている。そこで、そのノードを実際実行した割合をコンピュータごとに求め、代表ノードが適切であるかを解析し、代表ノードが不適切な場合、設定の見直しを求める。

- ネットワークを介したデータ転送量

データの出力を行う際、出力先が同一コンピュータ上であるか、ネットワークを介した他のコンピュータ上であるかによって、転送処理にかかる時間は、

データ量により大きく変わる。処理を円滑に進めるには、ネットワークを介した多量のデータ転送は好ましくない。イーサネットを介したデータの転送量は、1回当たり1024バイトである。そこでこの値を越え、データ転送が複数回行なわれる割合の多いノードを修正必要箇所として出力し、データを出力するノードと入力されるノードが、同一コンピュータに存在するよう分割の見直しを要求する。

- コンピュータの負荷 (実行キュー内のジョブ数)

ネットワークに接続された各コンピュータの平均負荷を求め、負荷の偏りを解析し、著しい偏りがある場合は分割の見直しを求める。

3 履歴の解析手法

履歴解析システムでは、ソフトウェアを一定期間使用した後、履歴の解析を行なう。そこで、しきい値とファジィを用いて同一データを解析し、両者の比較を行なう。

3.1 しきい値を用いた解析手法

履歴をファイルに記録するのは、あらかじめ設定した値を越えたときのみである。そのため、ファイルに記録するデータ量を削減することができるだけでなく、ファイルには修正が必要と思われる可能性の高いデータのみが記録される。

しきい値を用いた解析例を表1に示す。

表1: しきい値を用いた解析例

ノード	滞る割合	負荷	共有化分割	対策
1	33	—	no	—
2	100	1.78	yes	C
3	50	0.52	yes	A
4	100	2.41	no	B
5	0	—	yes	—

3.2 ファジィを用いた解析手法

しきい値を用いた場合、データを削減することはできないが、しきい値を下回ったデータは、解析には直接反映されない。そこでここでは、ファジィ推論を用いてデー

タに重み付けを行なってから解析を行なう。特にしきい値の前後では、重みに幅をもたせ、全てのデータを解析に反映できるようにした。さらに解析結果には、修正の必要な度も合わせて出力する。そのため、修正を行なう優先順位が明確になる。

ファジィを用いた解析例を表2に示す。対策には、修正の必要な度を示す1から3までの数字も表示する。数字は、値が大きくなるほど修正の必要性が増すことを示す。

表2: ファジィを用いた解析例

ノード	滞る割合	負荷	共有化分割	対策
1	39	—	no	—
2	61	1.78	yes	C1
3	61	0.52	yes	A1
4	75	2.41	no	B2
5	35	—	yes	—

4 おわりに

以上の解析により、ファジィを用いた解析の方が、より明確に分割の不具合な箇所を出力することができた。しかし、今回のシミュレーションではデータを削減せずに行なったため、この方法を実践で用いるとデータの保存に問題が生じてくる。そのための解決法として、以下の2つが考えられる。

- ランダムにデータの削減を行なう
- しきい値を低く設定し、その後ファジィを用いる

今後はデータ削減の観点から、上記のどちらの方が解析結果の信頼性を低下させずに、より効果的にデータを削減できるか検討する方針である。

参考文献

- [1] 山下利夫, 松山実, 横井利彰, “ソフトウェア自動分割システム RODS の概要,” 情報処理学会第45回全国大会講演論文集 (分冊5), pp.319-320, 1992.
- [2] 石井浩一, 山下利夫, 松山実, “ネットワーク環境における負荷分散の研究,” 情報処理学会第48回全国大会講演論文集 (分冊5), pp.187-188, 1994.
- [3] 石井浩一, 春木篤幸, 松山実, 横井利彰, “ネットワーク上の分散処理におけるスケジューリング,” 情報処理学会第49回全国大会講演論文集 (分冊1), pp.295-296, 1994.
- [4] 福田享子, 高橋透, 松山実, 横井利彰, “ソフトウェア自動分割システム RODS における履歴解析システムの開発,” 情報処理学会第50回全国大会講演論文集 (分冊5), pp.265-266, 1995.