

## 論理関数を用いた

1M-3

## 分散システムへの資源配置アルゴリズムの提案

津久井崇文 土屋達弘 菊野亨

大阪大学大学院基礎工学研究科情報数理系専攻

E-mail: {t-tukui, t-tutiya, kikuno}@ics.es.osaka-u.ac.jp

## 1 まえがき

文献 [1] では、異なる信頼性を持つ PE や通信リンクからなる分散システム上で、複数のファイルにアクセスすることで動作するプログラムの最適な配置を与える手法が提案されている。本研究では、この信頼性最適化問題を取り上げ、BDD を用いた高速なアルゴリズムを提案する。従来法では、分枝限定法を用い、トポロジーグラフ上の操作に基づいて限定操作のための評価値の計算し問題を解いていた。提案法では、評価値の計算に BDD を用いることで高速化を実現している。

## 2 モデル

$n$  台の PE とそれらを結ぶ通信リンクから構成される分散システムを仮定する。PE を節点、リンクを辺とみなした場合、分散システムの形状は無向グラフでモデル化される。それぞれの PE、リンクには信頼性が与えられている。信頼性とはその PE、またはリンクが正しく動作する確率を表す。さらに各 PE には容量と負荷の最大値が設定されている。このようなシステム上で  $m$  個のプログラム  $M_1 \dots M_m$  を動作させることを考える。各プログラムには大きさ、負荷、実行に必要なファイルの集合が設定されている。複数個のプログラムを同一の PE に配置する場合、そのプログラムの合計の大きさ、負荷が PE の容量、最大負荷を越えてはならない。プログラムの実行に必要なファイルは予め各 PE に配置されているものとする。必要な各ファイルに対し、そのファイルを保持する PE までの故障していないパスがプログラムの置かれた PE から存在する時、プログラムは実行可能とする。システムの信頼性を高めるため、同一のファイルが複数の PE に配置されている場合もある。

この様なシステム上でシステムの信頼性を最大にするプログラムの配置法を考える。システムの信頼性と

はシステム上に存在する全てのプログラムが動作する確率と定義する。

## 3 提案アルゴリズム

このアルゴリズムは大きく次の 2 つの部分に分けられる。1 つ目は  $M_1, \dots, M_m$  まで 1 つずつ順番にプログラムを配置していき、探索木を構成していくアルゴリズム、2 つ目は評価値を求めるためのアルゴリズムである。

## 3.1 探索手法

本研究では、分散システムの信頼性を最大にする配置を発見するために、 $A^*$  アルゴリズムを応用した探索を行う。

このアルゴリズムは配置を段階的に行う。まず、システムにプログラムが何も配置されていない状態から、あるプログラムを配置する。プログラムを配置する PE は、容量や最大負荷の制限が満たされていれば、PE の数だけ、すなわち  $n$  通り考えられる。次のプログラムを配置する場合も、配置する PE は最大  $n$  通り考えられる。従って探索空間は各ノードが最大  $n$  個の子を持つ木構造とみなす事ができ、これを探索木と呼ぶ。

プログラムが 1 つも配置されていない状態を表す root ノードを開始点とし、ノードを展開する事によって探索を行う。探索の間、展開によって新しく現れたノードはリストに入れる。このリストは子を未展開のノードを保持する。各ノードに評価値を定め、リストの中で最も評価値の高いノードから順に展開していく。ノードの評価値は、対応する配置から更に探索を進めて得られる完全な配置の信頼性の最大値に対する見積り値とする。この見積り値が巧妙であれば、低い信頼性の配置に対するノードが探索から除外され、短い探索時間で最適解に到達できる。最適解に到達する条件として、見積り値は実際に得られる信頼性以上の値でなければならない。このようにしてノードを展開していき、得られた完全な配置がシステムの信頼性を最大にするプログラム配置である。

“A Boolean-Function-Based Algorithm for Resource Allocation to Distributed Systems”, Takafumi Tsukui, Tatsuhiro Tsuchiya, and Tohru Kikuno, Department of Informatics and Mathematical Science, Osaka University, 1-3 Machikaneyama, Toyonaka, Osaka 560-8531, Japan

### 3.2 評価値の計算方法

評価値はMFST(Minimal File Spanning Tree)と呼ばれる概念に基づいて計算される。MFSTとはシステム上の1つのプログラムが、実行に必要なファイル全てにアクセスするために必要な極小の張る木のことである。従来法では評価値を求めるために複数のプログラムが動作するために必要な木の集合を求めることが必要であった。しかし、このために総当たりにMFSTを組み合わせる手法を用いており、あまり効率的とは言えない。本研究ではBDDを用いて論理関数間の演算によって評価値を計算する。

#### 3.2.1 論理式による表現

$PE_i$ ,  $PE_j$  とその間のリンク  $L_{i,j}$  からなる木が正常であるという事象  $upt_1$  は,  $PE_i$  が正常であるという事象  $x_i$ , と  $L_{i,j}$  が正常であるという事象  $x_{i,j}$  を用いて次のように表せる。

$$upt_1 = x_i \wedge x_j \wedge x_{i,j}$$

このようにしてある木が正常であるという事象を論理式を用いて表すことが出来る。配置済みのあるプログラムについて、少なくとも1つのMFSTが正常であるという事象は

$$mfst = upt_1 \vee upt_2 \cdots \vee upt_k$$

となる。さらに、配置済みの  $i$  個のプログラム全てが動作する事象  $mfst_i$  は、

$$mfst_i = mfst_1 \vee mfst_2 \vee \cdots \vee mfst_i$$

と表すことが出来る。BDDを用いることで、この論理式を効率良く求めることが出来る。評価値はこのようにして得られた論理式が表現する事象が起こる確率として計算される。

#### 3.2.2 評価値の計算

一般に論理式  $U$  は次の式で表現される。

$$U = (x \wedge U_{x=1}) \vee (\bar{x} \wedge U_{x=0})$$

ここで  $x$  は  $U$  の変数,  $U_{x=1}, U_{x=0}$  はそれぞれ  $U$  において  $x$  の値が真または偽である時に導出される論理式を表す。従って  $U$  が成り立つ確率  $Pr(U)$  は次式を用いて再帰的に求められる [2]。

$$Pr(U) = Pr(x) \cdot Pr(U_{x=1}) + Pr(\bar{x}) \cdot Pr(U_{x=0})$$

ここで  $Pr(x)$  は各PE又はリンクの信頼性なので、論理式から直接評価値を計算することが出来る。

## 4 比較実験

提案手法をC言語を用いて実装した。BDDの演算にはカーネギーメロン大学のライブラリを用いた。それぞれのシステム上で  $3 \leq m \leq 5$  個のプログラムの最適な配置を求める問題を与え、実行時間を比較した。得られた結果を表1に示す(単位は秒)。

各システムはそれぞれ6から8種類のファイルを保持し、同一のファイルが1から3個のPEに置かれている。また、配置されるプログラムはそれぞれ3から5種類のファイルを実行に必要とする。

この実験により、システムが小規模であってもBDDを用いた提案法の方が従来法より高速に最適な配置を求められることが分かった。さらにシステムの規模が大きくなると、提案法と従来法との実行時間の差が拡大することが分かった。これは、システムが大きくなるに従って各プログラムのMFSTの数が増加するので、それを総当たりに組み合わせる従来法がより不利になるためと思われる。

表1: 実験結果: 従来法との比較(単位: 秒)

PE数	$m=3$		$m=4$		$m=5$	
	従来法	提案法	従来法	提案法	従来法	提案法
5	0.02	0.02	0.06	0.02	0.15	0.02
6	0.09	0.03	0.12	0.03	0.38	0.06
7	1.47	0.24	3.71	0.35	22.39	0.89
8	26.77	0.88	47.48	1.43	151.67	3.19

## 5 おわりに

近年、2分決定グラフ(BDD)を用いて、分散システムの信頼性を評価する手法が注目を浴びているが、本論文では、BDDを信頼性評価ではなく信頼性最適化への応用について議論した。信頼性を最大化する分散システム上へ最適な資源配置を求める問題は、殆どの場合NP困難であるため、現実的な規模の問題に対して、出来るだけ高速に解を求めるアルゴリズムが必要とされる。本論文では、文献[1]で提案されたモデルについての最適化問題を取り上げ、BDDを用いた高速なアルゴリズムを提案した。評価実験の結果、従来法に比べて、提案法は最高で50分の1程度の時間で最適解を求めることができることが分かった。

## 参考文献

- [1] P. Ajith Tom and C. Siva Ram Murthy, "Algorithms for reliability-oriented module allocation in distributed computing systems," *J. Systems & Software*, Vol.40, pp.125-138, 1998.
- [2] S.-Y. Kuo, S.-K. Lu and F.-M. Yeh, "Determining terminal-pair reliability based on edge expansion diagrams using OBDD," *IEEE Trans. Reliability*, Vol. 48, No. 3, pp.234-246,1999.