

# UNIX ネットワークにおける動的負荷分散への遺伝的操作の導入

## 6U-2

山下貴幸 棟朝雅晴 高井昌彰 佐藤義治  
北海道大学 工学部

### 1 はじめに

複数の計算機を LAN で接続して資源の共有を図る分散システムにおいて、計算機間で負荷の分散を行うことにより、応答時間の短縮や資源利用率の改善など、システム性能の向上を図ることができる。この目的のため、種々の負荷分散方式が提案されてきた [1, 2, 3]。

負荷分散方式は、静的負荷分散方式と動的負荷分散方式に分類することができる。さらに、動的負荷分散は、負荷情報の管理とタスク転送の決定を一台の計算機で行う集中制御型と、各計算機で独立して行う分散制御型に分けられる。

本研究では [3] を基に、マルチキャストによるタスク転送要求の送出先の決定に対して遺伝的アルゴリズムを適用することで、より効率的な負荷情報の収集と利用を図る分散制御型動的負荷分散方式を提案する。また、UNIX ネットワークで構成される分散システム上に実装し、模擬タスクを用いたシミュレーション実験により性能評価を行う。

### 2 動的負荷分散アルゴリズム

はじめに、分散システム上の各計算機に、OS 内部の Round-Robin による待ち行列とは別に、FIFO の待ち行列を外部に設ける。これにより、多数のタスクが同時に実行され計算機の負荷が極端に大きくなる状態を避ける。

次に、 $\{0, 1\}$  からなる長さ  $n$  の文字列を設定し ( $n$  は計算機の台数)、文字列の  $k$  番目の文字が 1 のときに計算機  $k$  にタスク転送の要求を送出するものとする。この文字列が遺伝的操作の対象となる個体である。各計算機ごとに複数の個体からなる集団を用意し、それに対して遺伝的操作を適用する。

タスクの投入から終了までの流れを図 1 に示す。各計算機に投入されたタスクは外部待ち行列に入り、先頭から順に実行、あるいはタスク転送が起動される。

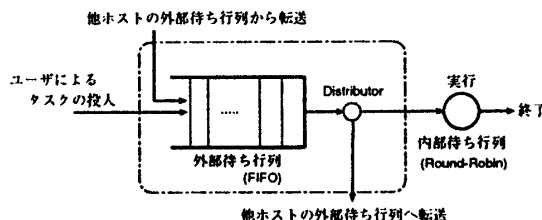


図 1: タスクの投入から終了まで

各計算機では一定間隔で自己の負荷の観測が行われ、負荷が重いと判断された場合にタスク転送手続きが起動される。計算機  $i$  の負荷状態は、プロセッサの絶対速度  $s_i$ 、外部待ち行列の長さ  $L_i$ 、過去 1 分間のロードアベレージ  $l_i$  を用いて次式で評価される。

$$H_i = \frac{aL_i + bl_i}{s_i} \quad (a, b \text{ は正の定数})$$

$H_i$  がある閾値  $H$  以上になった場合、負荷は重いと判断され、外部待ち行列内の先頭タスクに対してタスク転送手続きが適用される。

まず集団から一つの個体とその適合度値に比例した確率で選択され、その内容に従ってタスク転送要求がマルチキャストで送出される。負荷の軽い計算機が一つでも見い出されたなら要求は成功し、タスクが転送される。全ての計算機から要求が拒絶された場合、そのタスクは投入された計算機で実行される。

$k$  回目のタスク転送終了後の各個体の適合度値  $f(k)$  は、過去一定回数のタスク転送要求の送出数  $n(k)$  及び、受け入れられた数  $m(k)$  を用いて次のように与えられる。

$$f(k) = rf(k-1) + \frac{m(k)}{n(k)} \quad (r \text{ は } 0 < r < 1 \text{ なる定数})$$

すなわち、少ないタスク転送要求でより多くの要求が受け入れられた場合に適合度値の増加幅が大きくなり、次回にその個体を選択される確率が高くなる。

また、タスク転送が失敗した場合にのみ、一定確率で selection, crossover, mutation といった遺伝的操作を集団に対して適用する。

### 3 負荷分散システムの実装と評価

#### 3.1 実装

本方式で想定しているタスクは、互いに依存関係がなく独立なものであり、ネットワーク上の各計算機で実行可能なコマンドとする。

本システムは、タスク管理を行うために各計算機で起動されるサーバと、ユーザがサーバに対してタスクを投入するためのクライアントから構成される(図2)。

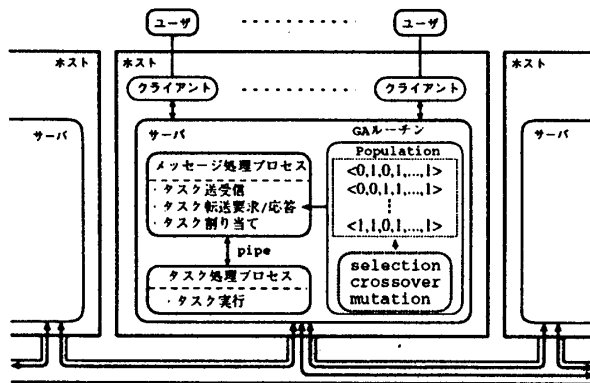


図2: システムの構成

#### 3.2 評価実験

比較対象として、タスク転送要求をランダムに1つだけ送出するユニキャスト方式、全計算機に送出するブロードキャスト方式の2方式を取り上げ、タスクの平均応答時間とタスク転送要求の送出数を比較し、遺伝的操作を導入した動的負荷分散方式の有効性を検証する。

使用したUNIXネットワークは、SPARCclassic×2, SPARCstation 10, SPARCstation IPX, NEWS-3460×2の計6台(順に $H_1 \sim H_6$ とする)を10Mbpsのイーサネットで接続した分散システムである。 $H_1, H_2, H_3$ の3台に単純な浮動小数点演算を繰り返すタスクを生成させ、その平均応答時間を測定した。タスクのループ回数を平均 $10^7$ 回の指数分布、到着間隔を平均 $\lambda$ 秒のポアソン到着とし、それぞれ100個のタスクを生成させた。ループ回数 $10^7$ 回のタスク1個の応答時間は $H_1$ で約6.2秒である。 $\lambda$ を変化させて実験を行った結果を図3,4に示す。

平均応答時間については、ブロードキャストと提案手法の間にはほとんど差は認められない。しかし、タスク転送要求の送出数は、提案手法を用いることで、ブロードキャストの1/2~1/3程度に軽減されている。

従って、分散システムを構成する計算機の台数がさ

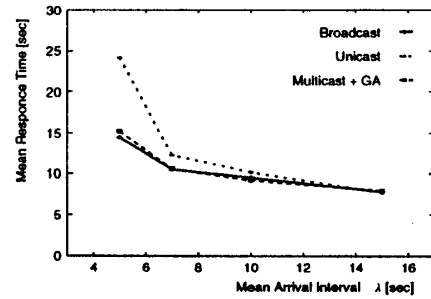


図3: 平均応答時間の比較

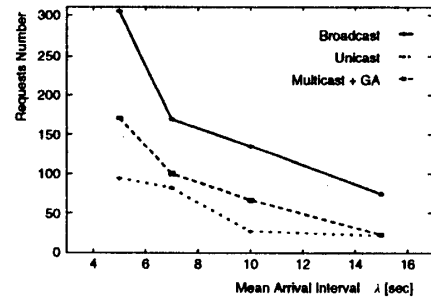


図4: タスク転送要求の送出数

らに多い場合には、平均応答時間の短縮の効果が期待できるものと考えられる。

### 4 まとめ

本研究では、遺伝的操作を導入した分散制御型の動的負荷分散方式を提案し、小規模なUNIXネットワーク上に実装した。より大規模な分散システムでの評価は今後の課題である。

### 参考文献

- [1] N.G.Shivaratri, P.Krueger and M.Singhal: "Load distributing for locally distributed systems", *IEEE COMPUTER*, Vol.25, No.12, pp.33-44, Dec. 1992.
- [2] 棟朝雅晴, 高井昌彰, 佐藤義治: "確率学習オートマトンと遺伝的アルゴリズムによる動的負荷分散", 情報処理学会研究報告(94-AI-93), Vol.94, No.20, pp.95-102, Mar. 1994.
- [3] 棟朝雅晴, 高井昌彰, 佐藤義治: "内部状態モデルに遺伝的操作を導入した動的負荷分散", 情報処理学会研究報告(94-DPS-67), Vol.94, No.105, pp.31-36, Dec. 1994.