

相互情報量と信頼度による予測選択を用いた実行時間予測手法

丹 野 祐 樹[†] 菅 谷 至 寛[†] 阿 曾 弘 具[†]

並列処理環境の利用効率を向上させるためには、効率的なタスクスケジューリングや負荷分散システム等が必要である。それらの中には、システムに投入されるプロセスの実行完了時間が既知であるという前提の基で構築されている手法が多々存在し、それらを実際にシステムに導入する際、投入されるプロセスの実行時間予測が必要となる。過去の実行履歴から予測対象プロセスと類似したプロセスを発見し、その情報を用いることで、実行時間予測が可能である。ここで、プロセス間の類似性をどのように規定するかという問題が生じる。本研究では、相互情報量を用いてプロセス間の類似性を判定するために重要となるプロセス情報を選択し、それによる類似プロセス集合を求め、 t 分布信頼度による予測選択を導入したプロセス実行時間予測法を提案する。予測実験を行った結果、提案手法が高い予測精度を示すことが確認できた。

Run Times Prediction Using Mutual Information and Predictor Selection

YUKI TANNO,[†] YOSHIHIRO SUGAYA[†] and HIROTOMO ASO[†]

Run time prediction, which must be accurate when it is employed in the actual system, is often requested for task scheduling algorithms and load balancing methods. To improve prediction accuracy, historical information of “similar” runs is used. That is because similar applications are more likely to have similar run times than applications that have nothing in common. The problem is how we define similarity between runs. In this paper, we propose a run time prediction method which uses mutual information to define information for similarity measure and predictor selection by a confidence measure. Experimental results indicate that the proposed method can predict run time more accurately than previous methods.

1. はじめに

近年、複数の計算機を高速ネットワークで接続した計算機クラスターや、インターネットを利用したグリッドコンピューティングといった並列分散処理環境の利用がますます拡大している。そして、それらの利用効率向上させるため、タスクスケジューリングや負荷分散システムの研究が行われている。簡単なタスクスケジューリングでは投入されるプロセスを各計算機の負荷に応じて、負荷が公平になるように最適に分配し(図1)、その結果全体としての利用効率の向上を目指す。しかし、提案されている手法の中には、投入されるプロセスの実行完了時間に関する情報を事前に得ていることを前提として構築されているものがある¹⁾。よって、それらを実際に並列分散システムに導入する際、実行時間の情報をどのように用意するかが問題となる。

この問題に対する解決策の1つとして、ユーザーが

実行時間の見積りをシステムに提出するという方法がある。しかし、一般的にユーザーはあまり正確な見積りを行えない傾向にあり、不正確な見積りの利用はシステム利用効率の向上を鈍らせ、あるいは逆に利用効率を悪化させてしまう²⁾。正確なプロセス実行時間の情報を得るために、システム側で実行時間予測を行うことが考えられる。

予測は、予測対象のプロセスと類似したプロセスが過去に実行されていれば、その情報を用いて行うことができる。これは、プロセスの実行時間は、全く共通点のないプロセスの実行時間より、プロセス名等で何か共通点のあるプロセスの実行時間に近い値をとると考えられる³⁾。

ここで、プロセス間の類似性をどのように規定するかということが問題である。類似性はプロセス情報を利用して計ることができる。プロセス情報とは、ユーザー名、プロセス名、使用メモリ量等の、プロセスに付随する情報である。類似性の判断基準が適切ではない場合、本来は類似していないプロセスを類似プロセスと判断することになり、結果として実行時間予測精度が悪化する恐れがある。

本研究では、相互情報量を用いてプロセス情報に關

[†] 東北大学 大学院 工学研究科 電気・通信工学専攻
Department of Electrical and Communication Engineering,
Graduate School of Engineering, Tohoku University

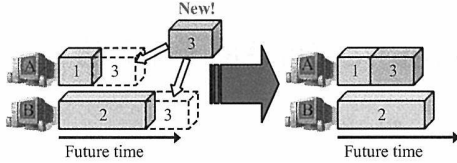


図 1 タスクスケジューリング例

するテンプレートを自動生成し、テンプレートで指定された情報について類似性を判断し、類似プロセス群による仮予測値について信頼度による予測選択を行う、プロセス実行時間予測法を提案する。類似判定用テンプレートを自動生成することで、利用環境に応じた類似性判断基準を定めることができ、様々な環境に対応することが可能となる。また、複数の仮予測を行い、信頼度による予測選択法を導入することで、最終的な実行時間予測精度の向上を図る。

本研究では、プロセス処理のうち CPU 処理の占める割合が一定以上となる計算タスクを予測対象とする。これは、すぐに終了する処理に対する予測は不要なためである。

本論文の構成は以下の通りである。2 章では本研究に関連する研究について述べる。3 章では提案する CPU 実行時間予測手法について述べる。4 章では予測実験結果を示し、6 章ではまとめと今後の課題を述べる。

2. 関連研究

Smith ら³⁾ は、遺伝的アルゴリズム、もしくは Greedy アルゴリズムを用いて類似判定用テンプレートを定義し、それに基づいて実行時間予測を行う手法を提案している。本論文で提案する手法は、Smith らの手法の考えを基礎にしている。

その他、実行時間予測に関する研究はこれまで数多く行われてきたが、特に良い予測精度を示したのは Senger らの手法⁴⁾ である。Senger らは、予測対象のプロセスと過去に実行されたプロセス間の距離を定義し、距離を考慮した重みを用いて実行時間予測値を計算する手法を提案している。以下に手法の概要を示す。

予測対象プロセス x_q と過去のプロセス x 間の距離 $E(x_q, x)$ は、プロセス情報の値から計算される。

$$E(x_q, x) = \sqrt{\sum_{i=1}^n d(a_i(x_q), a_i(x))^2}$$

n はプロセス情報数、 $a_i(x)$ はプロセス x におけるプロセス情報 a_i の値を示し、 $d(a_i(x_q), a_i(x))$ は、 x_i と x におけるプロセス情報 a_i についての値の差を示す。

このプロセス間距離が小さい順に過去のプロセスを k 個取得し、その実行時間 $f(x_i)$ を基にして、距離を考慮した加重平均により予測値 $f'(x_q)$ を計算する。

$$f'(x) = \frac{\sum_{i=1}^k w(x_i) f(x_i)}{\sum_{i=1}^k w(x_i)}$$

$$w(x_i) = \exp \frac{-E(x_q, x_i)}{\sigma^2}$$

σ^2 は定数である。

Senger らの手法は既存の手法よりも予測精度が高いが、全てのプロセス情報を等価に扱っている点について問題が生じる恐れがある。プロセス情報の重要性に差がある場合にその影響が顕著に現れる。非常に重要なプロセス情報であっても、他のあまり重要でないプロセス情報と同等に扱われてしまうため、結果的に真に類似しているプロセスを取得できない可能性がある。

3. CPU 実行時間予測

相互情報量を用いてプロセス情報の重要度・影響力を定め、重要な情報だけを用いて類似性を判定し、 t 分布信頼度による予測選択を導入した CPU 実行時間予測法を提案する。

本手法は、3 ステップに大きく分かれる。

- (1) 類似判定用テンプレートの生成
- (2) 仮予測の実行
- (3) 最適な仮予測の選択

まず、プロセス間の類似性を判断するために重要となるプロセス情報を知るため、類似判定用テンプレートを複数生成し、それを用いて求まる類似プロセスを基に仮予測を実行する。1 つのテンプレートによる類似プロセス群から 1 つの仮予測が生成される。その後、仮予測の中から最適なものを 1 つ選択し、その仮予測値を最終的な予測値として出力する。仮予測選択の基準として、 t 分布から算出した信頼度を用いる。

3.1 類似判定用テンプレートの生成

類似判定用テンプレートとは、プロセス間の類似性を判断するために重要となるプロセス情報の集まりであり、プロセス情報をいくつか組み合わせたものである。ある 2 つのプロセスを比較するとき、テンプレートに含まれる全てのプロセス情報について値が合致しているならば、その 2 つのプロセスは類似していると判断する。

このテンプレートが類似性判定に有効なプロセス情報からなっているならば、類似性判定の結果として取得するプロセスは、比較元のプロセスと非常に類似していると考えられる。有効なプロセス情報は、似た CPU 実行時間値を持つプロセスを集めるのに役立つプロセス情報を意味する。

類似判定用テンプレートを生成するために、まず、実行履歴に記録されているプロセス情報に関して、CPU 実行時間への影響力が高い順に順位付けを行う。影響力の評価、すなわち類似性判定における有効性・重要性の評価は相互情報量を利用する。その後、順位を考慮してテンプレートを生成する。

3.2 相互情報量

相互情報量とは、2つの確率変数の相互依存の尺度を表す量であり、片方の変数を知ることでもう片方の変数をどれだけ推測できるようになるかを示す。確率変数 X と Y の相互情報量 $I(X; Y)$ は次式で表される。

$$I(X; Y) = - \sum_i P(X_i) \log_2 P(X_i) - \left(- \sum_i \sum_j P(X_i, Y_j) \log_2 \frac{P(X_i, Y_j)}{P(Y_j)} \right)$$

i, j はそれぞれ X と Y の状態番号を示し、 $P(X_i)$ は X が X_i となる確率、 $P(X_i, Y_j)$ は X が X_i かつ Y が Y_j となる確率を示す。

右辺第1項は確率変数 X が持つ不確かさを意味し、第2項は Y という条件下で残った X の不確かさを意味する。つまり、相互情報量は、 Y という条件を知ることによって減少した X の不確かさの量を意味し、 X と Y の依存性が強い程、値は大きくなる。

3.3 プロセス情報の重要度順位付け

あるプロセス情報の CPU 実行時間に対する重要度を計算し、それに基づいて順位付けをする。重要度は相互情報量で計る。前述したように、相互情報量は2つの情報の相互依存の度合いを表す量である。よって、プロセス情報と CPU 実行時間との間の相互情報量が計算できれば、その値はそのプロセス情報がどれ程 CPU 実行時間に影響を与えるかを表すことになる。

あるプロセス情報の重要度は、次のように計算される。

- (1) テストデータ中のあるプロセスを基準に残りのプロセスとの比較を行う。CPU 実行時間の値について比較し、値が類似しているプロセスとそうでないプロセスの数を調べ、その生起確率を算出する。各プロセスの CPU 実行時間値は、基準となるプロセスの CPU 実行時間値で正規化しておく。CPU 実行時間値の差が閾値以内であれば類似と見なす。
- (2) 重要度を計算したいプロセス情報を順位決定済みのプロセス情報群に加え、その集合の全てのプロセス情報で値が合致するプロセスとそうでないプロセスの数を調べ、生起確率を算出する。なお、プロセス情報が文字列のものは文字列が一致するか否か、数値のものは正規化した値の差が閾値以内か否かを判定条件とする。
- (3) 得られた2つの生起確率から相互情報量を計算する。
- (4) 基準となるプロセスを変更して同様に相互情報量を計算する。
- (5) 相互情報量の平均値を、プロセス情報の重要度と設定する。

つまり、プロセス情報 a_i の重要度 $Dependency(a_i)$

```

D ← {∅}
A ← {a1, a2, ..., ai, ..., ak}
while A is not empty do
  A' ← A
  while A' is not empty do
    select ai from A'
    calculate Dependency(ai)
    A' ← A' - {ai}
  end while
  select ai with max Dependency(ai)
  D ← D + {ai}
  A ← A - {ai}
end while

```

図2 相互情報量による重要なプロセス情報の選択

は、次式で表される。

$$Dependency(a_i) = \sum_{j=1}^n \frac{I(X_j(D + \{a_i\}); X_j(CT))}{n}$$

ここで、 n はテストデータ中に含まれるプロセス数を指す。 D は既に順位が決定したプロセス情報群、 $X_j(CT)$ はテストデータ中のプロセス j を基準とした時の、CPU 実行時間に関する正規確率を意味し、 $X_j(D)$ はプロセス j を基準とした時の、プロセス情報群 D に関する生起確率を表す。

既に順位が決定しているプロセス情報群も重要度計算に用いているが、これは、順位決定済みのプロセス情報群を補完するプロセス情報の順位を高くするためである。

プロセス情報の順位付けの流れを図2に示す。まず、順位が未決定であるプロセス情報全てに対して重要度計算を行う。そして、その中から最も重要度が高いプロセス情報を1つ選択し、順位を設定する。

以後は重要度計算と順位付けを繰り返す、最終的に全てのプロセス情報について順位を定める。

3.4 類似判定用テンプレートの生成

求めたプロセス情報の重要度順位を考慮し、類似判定用テンプレートを生成する。今回は、4つのテンプレートを生成する。

各テンプレートで使用するプロセス情報を以下に示す。

テンプレート 1 重要度順位上位2つ

テンプレート 2 重要度順位上位3つ

テンプレート 3 重要度順位上位4つ

テンプレート 4 重要度順位上位5つ

今回は、大まかな判断基準を持つ類似判定用テンプレ

レートを基に、徐々に判断基準が厳しくなるように類似判定用テンプレートを生成する。判断基準が厳しいテンプレートによって取得された類似プロセス群は、近い CPU 実行時間値を持つ可能性が高い。しかし、判断基準の厳しさから、そのようなテンプレートからは類似プロセスを全く取得できない場合も考えられる。そういった事態に対処するため、判断基準の厳しさに変化を付けてテンプレートの生成を行うようにする。

4. 類似判定用テンプレートを使用した仮予測

類似判定用テンプレートを利用して、仮予測を行う。それぞれ 1 つのテンプレートは 1 つの仮予測で使用し、仮予測は用意したテンプレートの数だけ行われる。

各テンプレート毎に以下の処理を行う。

- (1) 実行履歴から類似プロセスを取得
- (2) CPU 実行時間仮予測値算出
- (3) 仮予測の信頼度算出

まず、類似判定用テンプレートを使用して予測対象と類似したプロセスを取得する。そして、取得した類似プロセスの CPU 実行時間値から仮予測値を計算する。最後に、 t 分布を用いて仮予測の信頼度を計算する。

4.1 類似プロセスの取得

実行履歴を探索し、予測対象と類似したプロセスを取得する。類似プロセスとは、類似判定用テンプレートに含まれるプロセス情報の全てにおいて、予測対象プロセスと値が一致するものを指す。なお、プロセス情報が一致するかどうかは、重要度計算時と同様に、プロセス情報が文字列であれば完全一致した場合に値が一致とみなし、数値であれば正規化した値の差が閾値以内に収まっていれば一致とみなす。

探索は予測時点から過去方向に行い、予測時点から過去一定期間探索をするか、類似プロセスを規定個数取得したならば探索を終了する。この探索打ち切りは、データの最新性も考慮した類似プロセスの取得のためである。

4.2 仮予測値の算出

初めに、より予測対象プロセスに類似したプロセスのみを仮予測値計算に使用するために、取得した類似プロセス群の絞りこみを行う。取得した類似プロセス間においてプロセス情報の値に差があるならば、その値が予測対象プロセスの値に近い類似プロセスは、予測対象プロセスにより類似していると考えられる。よって、それらのみを使うことで、予測精度の向上が見込まれる。また、類似プロセス間においてプロセス情報の値が等しい場合でも、データの最新性に着目すれば類似プロセスの絞り込みができる。これは、初期の実行時にはデバッグし、後に本動作するようなプロセスは、CPU 実行時間が大きく変化する可能性があり、後期に投入されたプロセスが予測対象プロセスにより類似していると考えられるためである。

絞り込みを行うため、まずは類似プロセスをソートする。類似判定用テンプレートにおいて値が数値のプロセス情報を用いている場合は、テンプレートに使用されている数値プロセス情報の中で、最も順位が高いプロセス情報の値に着目し、その値に基づいて昇順に類似プロセス群をソートする。値が数値でないプロセス情報の場合は、データが最新の順にソートする。ソート後、類似プロセス群の CPU 実行時間において、クラス間分散が最大となるようにプロセス群を 2 分割し、分割された 2 集合のうち前半の集合を選択し、後半の集合を破棄する。

この絞り込みの結果残った類似プロセス群について、その CPU 実行時間平均値を仮予測値とする。

4.3 仮予測の信頼度算出

複数の仮予測から最適なものを選択すれば、予測精度は向上すると期待できる。しかしこの場合、最適性をどのように判断するかが問題となる。本手法では、各仮予測に対してそれぞれ信頼度を求め、その値が最大の仮予測を採用する。信頼度の算出には t 分布を用いる。

4.3.1 t 分布

t 分布は確率分布の一つであり、統計学において標本集団から母集団を検定する場合等に用いられる⁵⁾。

t 分布の確率密度関数 $f(t)$ は次式で与えられる。

$$f(t) = \frac{1}{\sqrt{k}B(k/2, 1/2)} \left(1 + \frac{t^2}{k}\right)^{-\frac{k+1}{2}}$$

k は自由度であり、自由度が高いほど $f(t)$ は正規分布の関数に近づく。 $B(\lambda_1, \lambda_2)$ はベータ関数である。

CPU 実行時間予測は、実行中プロセスの CPU 実行時間は過去に実行された類似プロセスの値にほぼ等しいという考えに基づく手法である。従って、類似プロセス集合を適切に取得できたならば、その標本集団から求められた統計量は、 t 分布に従うと仮定できる。

4.3.2 信頼度の設定と予測選択

取得した類似プロセス集合と仮予測値に対して t 分布を用い、各仮予測の信頼度 *Confidence* を設定する。信頼度は、取得した類似プロセス集合を標本集団、仮予測値を標本平均、求めたい実測値を母平均と見なし、標本平均と母平均の差が標本平均の 10%以内に収まる確率と定める。すなわち、次式で与えられる。

$$Confidence = 2 \int_0^A f(t) \cdot dt$$

$A = \frac{P-0.1}{\sqrt{S/n}}$ であり、 P は算出した仮予測値、 S は取得した類似プロセス群における CPU 実行時間の不偏分散、 n は取得した類似プロセス数である。また、このときの自由度は $k = n - 1$ となる。取得した類似プロセス数が多いほど、またその CPU 実行時間がまとまっているほど信頼度は高くなる。

この信頼度を各仮予測に対して計算する。そして、信頼度が最大となる仮予測を採用し、その仮予測値を

最終的な CPU 実行時間予測結果とする。

5. 予測実験

本手法の有用性を確認するため、CPU 実行時間予測実験を行った。

まず、テストデータから類似判定用テンプレートを4つ生成する。その後、生成したテンプレートを用いて予測実験を繰り返し行う。データベースは LANL CM-5 システムの実行履歴を使用する。この履歴情報から、プロセス情報としてユーザー名、プロセス名、グループ名、引数、メモリ使用量、使用 CPU 数、待ち時間の7つを選んだ。

システムの初期に投入された 1000 個のプロセスをテストデータとし、類似判定用テンプレートを生成する。その後、テストデータを除いたプロセスの中からランダムに選択した 2232 個のプロセスについて、予測実験を行う。また、既存手法として Senger らの手法 (IBL として示す) による予測を行い、予測精度を比較する。

類似プロセスの検索期間は予測点から過去 3 か月とし、提案手法の各仮予測で取得する類似プロセスの最大数は 10 とする。比較対象の IBL 予測に関する条件は、Senger らが設定したものをそのまま用いる。

実行時間の実測値 R_t と予測値 P_t の差を正規化した誤差率 $Error$ は、以下の式で表される。

$$Error = \frac{|R_t - P_t|}{R_t} \cdot 100$$

予測性能は、予測結果の中である範囲の誤差率を示したプロセスがどの程度存在するか、という誤差率分布で評価する。

5.1 実験結果

テストデータから求められたプロセス情報の重要度順位は、重要な順に、グループ名、プロセス名、使用 CPU 数、メモリ使用量、ユーザー名、引数、待ち時間となった。この結果から生成された類似判定用テンプレートは次のとおりである。

テンプレート 1 グループ名、プロセス名

テンプレート 2 グループ名、プロセス名、使用 CPU 数

テンプレート 3 グループ名、プロセス名、使用 CPU 数、メモリ使用量

テンプレート 4 グループ名、プロセス名、使用 CPU 数、メモリ使用量、ユーザー名

この類似判定用テンプレートに基づいて予測を行った結果を表 1 に示す。

予測数は、全 2232 回の予測中、予測が可能であった数を示す。すなわち、類似プロセスが見つかったプロセスの個数である。また、誤差率分布の各項目は、その誤差率範囲の結果となった予測の予測数に対する割合を示す。つまり、誤差率が 0 から 10%、10 から 20%と

表 1 CPU 実行時間予測結果

予測手法	予測数	誤差率分布 [%]			
		0-10%	10-20%	90-100%	100-%
提案手法	2185	63.89	9.43	0.64	8.51
IBL	2232	59.41	11.42	0.85	8.51

低い部分については値が大きい程、90 から 100%や 100%以上と高い部分については値が小さい程予測精度が良いということを示す。

表 1 から、提案手法は IBL よりも予測を行えた数が少ないことが分かる。これは、提案手法は使用するプロセス情報を限定し、さらにプロセス情報の値が合致した場合のみ類似プロセスを取得するため、類似プロセスをうまく取得できない場合があるということを示している。しかし、予測数の減少は少ないため、さほど深刻な事態ではないと考えられる。IBL でのみ予測が行えたプロセスの予測結果を表 2 に示す。

表 2 IBL でのみ予測が行えた CPU 実行時間予測結果

予測数	誤差率分布 [%]			
	0-10%	10-20%	90-100%	100-%
47	31.91	17.02	2.13	14.89

表 1 の予測結果と比較すると、IBL でのみ予測が行えたプロセスは全体的に予測精度が劣っていることが分かる。つまり、提案手法はそのような予測が困難なプロセスを事前に排除できたと言える。

公平な比較を行うため、提案手法と既存手法のどちらでも予測が行えた 2185 個のプロセスについて、その誤差率分布を表 3 に示す。

表 3 CPU 実行時間予測結果

予測手法	誤差率分布 [%]			
	0-10%	10-20%	90-100%	100-%
提案手法	63.89	9.43	0.64	8.51
IBL	60.00	11.30	0.82	8.38

表 3 より、誤差率 10%までは 3.8%、20%までは 1.52%の差となり、大きな誤差率を与えるものも少なく、提案手法が既存手法より予測精度が良いと言える。提案手法の現段階におけるプロセス情報重要度順位付けや類似判定用テンプレート生成は、単純な手法で行われている。よって、その部分について改良を加えれば、提案手法の予測精度はさらに向上すると考えられる。

6. まとめ

本論文では、相互情報量を用いてプロセス間の類似性を判定するためのプロセス情報の重要度を定め、そ

れに基づいて複数の類似プロセス集合を求めて、それぞれで仮予測を行い、 t 分布信頼度により予測選択を行う CPU 実行時間予測手法を提案した。相互情報量を用いることで重要なプロセス情報を抽出でき、また、信頼度による予測選択を行うことで、予測精度の向上が可能となる。

実験の結果、本提案手法が既存の手法よりも予測結果を示すことが確認できた。

今回提案した手法における類似判定用テンプレートは、1 回の重要度順位付けに基づき、上位から順に使用個数を増やしつつ生成される。これは単純な方法と言えるので、今後改良を加える必要がある。また、順位の基となるプロセス情報の重要度は、テストデータに強く影響を受ける傾向が見られる。しかし、今回は 1 つのテストデータのみから重要度の計算を行っているの、その部分についても改良を加える必要がある。具体的な改良方法としては、テストデータを複数用いる方法が考えられる。

また、今後は今回提案した手法と IBL による予測手法を組み合わせた予測法も検討する。

参 考 文 献

- 1) S. C. Kim, S. Lee and J. Hahm : Push-Pull: Deterministic Search-Based DAG Scheduling for Heterogeneous Cluster Systems, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 18, no. 11, pp. 1489–1502 (2007)
- 2) D. Tsafir and D. G. Feitelson : Backfilling Using System-Generated Predictions Rather than User Runtime Estimates, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 18, No. 6, pp. 789–803 (2007).
- 3) W. Smith, V. Taylor and I. Foster : Predicting application run times with historical information, *Journal of Parallel and Distributed Computing*, Vol. 64, pp. 1007–1016 (2004).
- 4) L. J. Senger, M. J. Santana, and R. H. C. Santana : An Instance-based Learning Approach for Predicting Execution Times of Parallel Applications, *Proceedings of the 3rd International Information and Telecommunication Technologies Symposium*, pp. 9–15 (2005).
- 5) 鈴木義也 他: 概説 数理統計, 共立出版株式会社 (1994).