

LA-006

プロセス情報を利用した実行時間予測と信頼度による予測選択手法 Runtime Predictions Using Process Properties and Predictor Selection

丹野 祐樹[†]
Yuki Tanno

菅谷 至寛[†]
Yoshihiro Sugaya

阿曾 弘具[†]
Hiroto Aso

1. はじめに

並列分散処理環境の効率的な利用には、実行中プロセスの実行時間予測が有用である。計算機クラスタにおいて、クラスタを構成する各マシンで実行中にあるプロセスがどの程度の時間で完了するか知ることにより、新規プロセスの投入先やスケジューリングを決めることができ、全体としての処理完了時間を改善できる。

実行中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、ある程度正しい実行時間予測が可能となる [2]。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よりは、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推測されるためである。

本論文では、より多くのプロセス情報を利用してより正確な実行時間予測を実現する手法と、複数の予測手法による予測結果からより適切な予測を選択する手法を提案し、これらの手法の精度を検証する。

2. CPU 実行時間予測

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の予測は不要なためである。使用するプロセス情報は、Linux の ps コマンドを 60 秒毎に実行することで取得し蓄積する。ここで取得している情報は、ユーザー名、プロセス名、経過実実行時間、経過 CPU 時間、使用メモリ量、引数である。

本研究で提案する実行時間予測手法は、過去の類似プロセスから CPU 実行時間を予測する部分と、実行中プロセスの状態に合わせて予測値を補正する実実行時間補正部分の 2 つに分かれている。CPU 実行時間予測については、4 つの手法によってそれぞれ CPU 実行時間の仮予測値を求め、その中から最も適していると思われる値を選択する手法を示す。

2.1 類似プロセスの選択

予測時点から過去一定期間内に実行されたプロセスを検索し、指定したプロセス情報を基に類似プロセスを選択する。ただし、その CPU 実行時間が実行中プロセスの経過 CPU 実行時間以下となっているものについては選択しない。検索時に使用するプロセス情報の指定方法により 4 手法を考える。各手法で使用するプロセス情報を表 1 に示す。各手法での仮予測値算出法を以下に示す。

表 1: CPU 実行時間予測手法

予測手法	使用するプロセス情報
Normal	ユーザー名, プロセス名
Args	ユーザー名, プロセス名, 引数
Mem	ユーザー名, プロセス名, 実メモリ使用量
Mem+Args	ユーザー名, プロセス名, 引数, 実メモリ使用量

2.1.1 Normal, Args での仮予測値算出

Normal ではユーザー名とプロセス名, Args ではさらに引数を考えて、それらが一致した場合に類似プロセスとして選択する。

選択した類似プロセス群を実行開始時刻順に並べ、CPU 時間のクラス間分散が最大となるところで 2 分割をする。分割された 2 集合のうち実行開始時刻が新しい側を選択し、その CPU 実行時間平均値を仮予測値とする。

この分割により、初期の実行時にはデバッグし、後にプログラムを本動作させるといったような、CPU 実行時間が大きく変化する状況に対応することが可能となる。

2.1.2 Mem, Mem + Args 時の仮予測値算出

Mem ではユーザー名とプロセス名, Mem+Args ではさらに引数が一致するプロセスを類似候補として検出する。この候補プロセスと実行中プロセスとの実メモリ使用量を予測点 T から過去 5 状態に渡って比較し、累積誤差 $CumError$ を計算する。

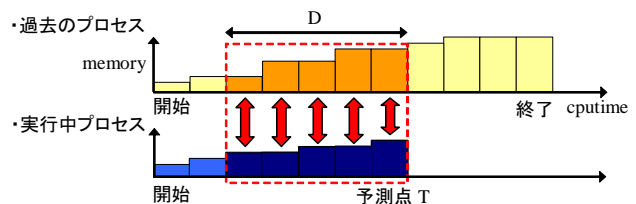


図 1: 実メモリ使用量の累積誤差算出範囲

$$CumError = \sum_{k=0}^{D-1} |M(T-k) - M'_{close}|$$

ただし、D は窓の範囲で $D = 5$ とし、 $M(T-k)$ は実行中プロセスの予測点 T から過去 k 状態目の実メモリ使用量、 M'_{close} は候補プロセスの対応する状態における実メモリ使用量である。この累積誤差が実行中プロセスの窓内実メモリ使用量合計値の 10% 以内ならば、類似プロセスとして選択する。

選択した類似プロセス群について累積誤差が小さい順に並びかえを行なう。CPU 実行時間のクラス間分散が最大となるところで、集合を 2 分割し、累積誤差が小さい側の集合の CPU 実行時間平均値を仮予測値とする。

2.2 信頼度による予測手法の選択

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、どのようにして仮予測を選択するかが問題となる。本手法では、各

[†] 東北大学大学院工学研究科

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。

2.2.1 t 分布

t 分布は統計学における確率分布の一つであり、標本集団から母集団を検定する場合等に用いられる [3]。

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 実行時間予測法は、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。

2.2.2 信頼度の設定

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。

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

$A = \frac{P \cdot 0.1}{\sqrt{S/n}}$ 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。

3. 実実行時間の推定

実行中の計算タスク数が搭載 CPU 数より多い場合、CPU 資源を分け合うことになるため、終了するまでの実実行時間がその分長くなる。そこで、[1] と同様の手法により、CPU 実行時間予測値を延長補正した値を実実行時間予測値とする。

CPU 利用率が $\frac{\text{搭載 CPU 数} \cdot 80}{\text{実行中プロセス数}} [\%]$ 以上となる実行中プロセスをフル稼働タスク、そうでないものを余裕タスクとする。実行中のフル稼働タスク数が搭載 CPU 数より多い場合、フル稼働タスクは、フル稼働タスク数 a が搭載 CPU 数より多い区間 $I(a)$ を $I'(a) = I(a) \cdot \frac{a}{\text{搭載 CPU 数}}$ とすることで延長補正を行う。余裕タスク、または、実行

中の計算タスク数が搭載 CPU 数以下の場合、CPU 実行時間予測値に CPU 利用率を除算した値を実実行時間予測値とする。

4. 予測実験

本手法の有効性を確認するために実行時間予測実験を行った。対象は Xeon 2.4 GHz プロセッサを 2 台搭載した Linux マシン 1 台で、研究室のメンバーで共用し、主に科学技術計算に用いている。本手法では CPU 処理の割合が低いプロセスは対象としていないため、対話的シェルや CPU 実行時間が 60 秒未満のプロセス、CPU 利用率が 10% 未満のプロセスについては予測を行わない。また、類似プロセスが全く存在しないプロセスは除外する。Normal 予測のみ有効で取得類似プロセス数が 1 しかないプロセスも、類似プロセスを適切に取得できなかったと見なし、評価対象から除外する。類似プロセス検索期間は最大 3 ヶ月、取得類似プロセス最大数は 10 とする。予測精度は、実測値 R_t と予測値 P_t の誤差率 *Error* で評価する。

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

4.1 人工プロセスによる予測実験

2007 年 2 月から 3 月まで上記のマシンでデータ収集を行ったが、性能を検証するために十分な量のプロセスが記録されなかったため、実験用に作成したプロセスを計算機に投入することで、メモリ使用量の変動を考慮した場合の予測実験結果を示す。作成したプロセスは以下の 4 つである。

task-360 CPU 実行時間が約 6 分かかり、引数として整数を渡すと CPU 実行時間が入力整数倍になる

task-600 CPU 実行時間が約 10 分もしくは 20 分かかり、それぞれ使用する実メモリ量が異なる

task-m-incre3 実メモリ使用量が増加し、処理の半分を終えたところでメモリを一度解放し、また徐々に増加する

task-m-step 実メモリ使用量が task-m-incre3 と同様の变化をするが、その変動幅が大きい

投入するプロセスとその引数はランダムに決定した。

ランダムに予測点を定め 50 回予測実験を行なったところ、予測対象プロセス総数は 93 個となった。このうち、各手法で類似プロセスが存在したものを、予測が有効なプロセスと呼ぶ。

CPU 実行時間を予測した結果を図 2 に示す。図は予測結果の誤差率に対して、その誤差率をもつ予測結果の割合を表しており、4 つの予測手法それぞれと、その中から信頼度が最も高いものを選択した場合の結果を示している。予測が有効であったプロセス数は、信頼度による選択を行なった場合は 61、Mem+Args 予測のみの場合は 52、続いて Mem が 55、Args が 57、Normal が 60 となった。プロセス情報として引数を使用すると予測精度が向上し、メモリ使用量を使用するとさらに予測精度が向上

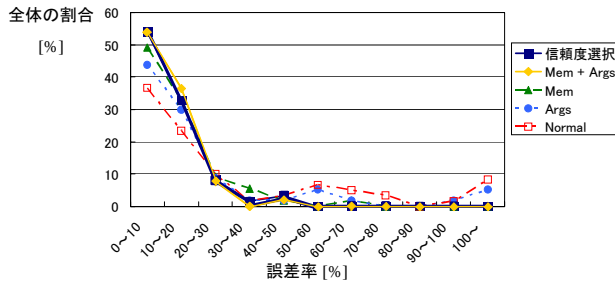


図 2: CPU 実行時間予測結果

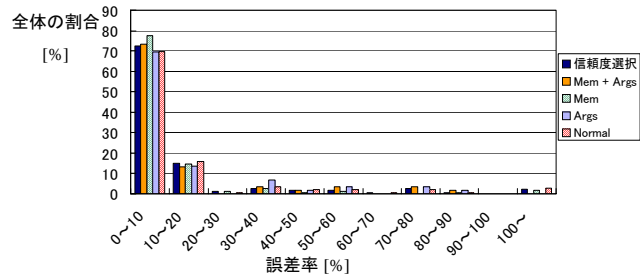


図 4: CPU 実行時間予測結果

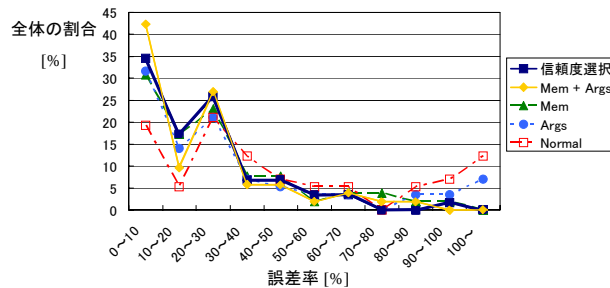


図 3: 実実行時間予測結果

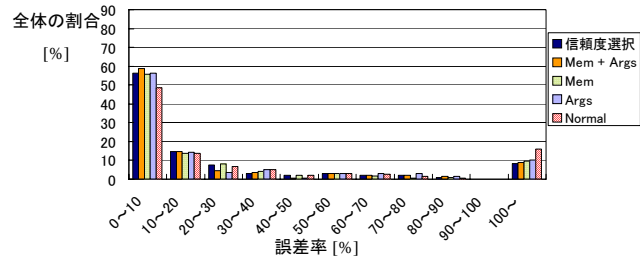


図 5: 信頼度 0.9 以上の CPU 実行時間予測結果

した。また信頼度による選択を行った場合、誤差率の割合は Mem+Args とほぼ等しいが、予測が有効だった個数が、52 に対して 61 と増えているため、最も良い予測結果と言える。よって、信頼度による手法選択が予測に有用であると考えられる。

CPU 実行時間予測結果に補正を加え、実実行時間を予測した結果を図 3 に示す。予測が有効となったプロセス数は、信頼度選択時が 58, Mem+Args, Mem が 52, Args, Normal が 55 となった。実実行時間予測結果からも、メモリ使用量を考慮すると予測精度は向上し、また信頼度による手法選択が有用であるということが言える。

4.2 実データによる予測実験

信頼度の有効性を確認するため、過去に取得した 2002 年 12 月から 2006 年 10 月までの実データを用いて予測実験を行った。この過去のデータベースはメモリ使用量の経過状態が記録されておらず、終了時の値のみ利用可能であった。しかし、プロセスには開始から終了まで使用メモリ量が変化しないプロセスも存在するため、それらのデータベースと見なすことができ、信頼度の有効性を議論することが可能だと考えられる。

ランダムに予測点を定め、300 回予測実験を行った。予測対象プロセス総数 501 個のうち CPU 実行時間予測が有効となったプロセス数は、信頼度選択時が 332, Mem+Args が 136, Mem が 326, Args が 140, Normal が 340 となった。t 分布信頼度の有効性を確認するため、全予測結果での誤差率に対する割合を図 4 に、信頼度 0.9 以上のものの割合を図 5 に示す。

信頼度が 0.9 以上となる場合、全体の結果よりも誤差率が 100% を越える割合が低く、誤差率が 20% 以内に収まる割合は高いという結果が得られた。これより、信頼度を予測精度を表す指標と見なすことができ、t 分布による信頼度の有効性が確認できたと考えられる。

5. まとめ

本論文では、実行中プロセスの情報を利用した実行時間予測手法を提案した。プロセス情報を使用した予測を複数行い、その中から適切な予測を選択することで予測精度が向上することを確認し、また、選択基準として t 分布による信頼度を用いることの有効性を示した。

今回、予測のため実メモリ使用量を用いたが、これは予備実験において仮想メモリ使用量を用いた場合よりも予測精度が良かったためである。しかし、実メモリ使用量はその時の計算機の状態に左右されるため、使用するメモリ情報を仮想メモリ使用量にするか、もしくは両方使用し信頼度による選択を行う、といった変更を行う予定である。また、実験用に作成したプロセスでなく、実際に計算機に投入されるプロセスについて予測を行い、本提案手法の有効性について判断する必要がある。

謝辞

本研究の一部は、科学研究費補助金 (課題番号:19700052) による補助のもとで行われた。

参考文献

- [1] 立見博史 他: “プロセス情報と実行時間予測を利用した統合的計算機負荷長期予測手法”, 電気情報通信学会論文誌, Vol.J89-D No.11, Nov. 2006.
- [2] Warren Smith, et al.: “Predicting application run times with historical information”, J. Parallel Distrib. Comput, Vol.64, pp. 1007-1016, 2004.
- [3] 鈴木義也 他: “概説 数理統計”, 共立出版株式会社, Dec. 1994.