A-028

プロセス情報と負荷変動傾向を用いた計算機負荷長期予測モデル
Long-term Load Prediction using Process Information and a Time Series Tendency

立見 博史 † 菅谷 至寛 † 阿曽 弘具 †
Hiroshi Tatsumi Yoshihiro Sugaya Hirotomo Aso

1. はじめに

プロセッサ負荷の違いはアプリケーションの実行時間に大きな影響を与える．そのため，グリッドなどの分散処理環境で処理時間全体を短縮するためには効率的な資源配置が必要である．これを実現するためにプロセッサやネットワークの負荷を予測し，スケジューリングに利用する手法 (NWS[1],RIS[3]) が研究されている．

負荷予測の研究に関して，Yang ら [2] は NWS の負荷予測手法を改良し，予測直前の負荷変動傾向を考慮することが予測に有効であると述べている．また，RIS[3] ではプロセッサ負荷の高精度予測手法としてパターン検索法 (高精度予測モジュール) を提案している．パターン検索法は予測を行う時点の負荷変動パターンと類似したパターンを過去のデータベースから検索し，これをもとに予測を行う手法であり，プロセッサ負荷の長期予測を可能にしている．

プロセッサ負荷の長期予測を用いる利点は数ステップ先のプロセッサ負荷を考慮しての資源配置が可能となることである．そこで本研究では平行移動を用いたデータベース検索，さらにはプロセス情報，予測直前の負荷変動傾向などを考慮したプロセッサ負荷の長期予測手法を提案する．プロセッサ負荷予測がより正確になることで，従来の資源配置法を用いてもより適切な資源配置が得られることが期待される．

2. 予測アルゴリズム

時刻 t での実測値を $V(t)$，予測値を $P(t)$ と記す．予測開始時刻を時刻 T とし，予測に用いる過去のデータベースの負荷系列を $V(0)$, $V(1)$, ..., $V(T)$，予測する負荷系列を $P(T+1)$, $P(T+2)$, ..., $P(T+p)$ とする．

2.1 平行移動を用いたデータベース検索

まず各時刻で実窓 $\mathbf{W}_t(i)(=(V(t-D+1), \ldots, V(t)))$ を設定し，各実窓の最終時刻の値が 0 になるように窓 $\mathbf{W}'_t(i)(=\mathbf{W}_t(i)-V(t)\mathbf{1})$ を作成する．D は窓幅を示すパラメータである．次に窓 $\mathbf{W}'_t$ と窓 $\mathbf{W}'_T$ との誤差 $err(t)$ を計算し，$err(t)$ が最小値をとる時刻 s を求める．

$$err(t) = \sum_{i=0}^{D-1} |\mathbf{W}'_t(i) - \mathbf{W}'_T(i)|$$

時刻 T 以後の予測結果は，時刻 s 以後の負荷変動を平行移動したもの，つまり $P(T+i) = V(s+i)-V(s)+V(T)$ とする．

2.2 プロセス名を利用した予測用窓の優先度付け

$err(t)$ を最小にする時刻 s の窓を使った予測が予測値として最適であるとは限らない．$err(t)$ が 2 番目に小さい，または 3 番目に小さい時刻の窓を用いた予測の方が適当である場合も存在する．そこで複数の予測結果の中から最適に近い予測結果を選び出すことを考える．そのためにまず実行プロセス名の情報を用いて複数の予測用窓に優先度をつける．

$err(t)$ の小さい方から k 個の窓を考える．k 個のそれぞれの時刻と時刻 T で実行されていた共通のプロセスの数を求める．例えば，選び出された k 個の窓の中の時刻 u の窓を考える．時刻 u の時点で実行中だったプロセス，時刻 T の時点で実行中だったプロセスが表 1 のようであった時，時刻 u と時刻 T で共通のプロセス数は 3 となる．

最終的に k 個の予測用窓に，共通のプロセス数が多い順，同数の場合は $err(t)$ が小さい順に高い優先度を付ける．

表 1: 時刻 u と時刻 T で実行中のプロセスの例

時刻 u	***predict***, ***predict***, predict, ***perl***, emacs
時刻 T	***predict***, ***predict***, ***perl***, perl, netscape

2.3 CPU 利用率を用いた予測値補正

負荷の増加開始時や減少開始時の予測において，時刻 T での CPU 利用率は時刻 T 以後の負荷変動の目安になると考えられる．そこで時刻 T の直前での負荷変動傾向，時刻 T で実行中のプロセスの CPU 利用率を考慮して予測結果の補正を行う．時刻 t での 1 つの CPU n の CPU 利用率が $x_n(t)(0 \leq x_n(t) \leq 1)$ であった時，負荷収束予測値 $LCV(t) = \sum_n x_n(t)$ とする．

$LCV(T) > V(T)$ である時，負荷が連続増加中 ($V(T-2) < V(T-1) < V(T)$) または大幅に増加している場合 ($V(T) - V(T-1) > 1$)，時刻 T 時点でのプロセスが予測期間中継続して実行されるものと仮定し，すべての i について予測結果を $P(T+i) = LCV(T)$ とする．

$LCV(T) * (const) < V(T)$ である時，負荷が減少傾向 ($V(T-1) > V(T)$ または $V(T-2) > V(T-1)$) にある場合，時刻 T 以後の負荷は時刻 T 時点での CPU 利用率まで減少すると仮定し，すべての i について予測結果を $P(T+i) = LCV(T)$ とする．$const$ とはある定数で，$LCV(T)$ と $V(T)$ との間にある程度の差がある時のみこの補正を行うことを意味する．

2.4 予測直前の負荷変動傾向を考慮した予測結果の選択

2.3 節の CPU 利用率を使った補正が行われなかった時，時刻 T の直前の負荷変動を考慮した予測用窓を選択する．

負荷が連続増加中 ($V(T-2) < V(T-1) < V(T)$)，または大幅に増加している時 ($V(T) - V(T-1) > 1$)，連続増加 ($V(u) < V(u+1) < V(u+2)$) を予測している予測用窓の中から優先度の最も高いものを選択する．その他の負荷増加中の時 ($V(T-1) < V(T)$)，時刻 T 以後の負荷変動は増加する可能性が大きいと考え，k 個の予測結果から過去のデータベース上で連続減少 ($V(u-1) >$

†東北大学大学院工学研究科

$V(u) > V(u+1)$ または $V(u) > V(u+1) > V(u+2)$) するものを除き，それらの中で優先度の最も高い予測結果を選択する．

負荷が連続減少している時 ($V(T-2) < V(T-1) < V(T)$)，または大きく減少している時 ($V(T-1) - V(T) > 1$) も同様に，連続減少 ($V(u) > V(u+1) > V(u+2)$) を予測している予測用窓の中から優先度の最も高いものを選択する．その他の負荷減少中 ($V(T-1) > V(T)$) の時も同様に，k 個の予測結果から過去のデータベース上で連続増加 ($V(u-1) < V(u) < V(u+1)$ または $V(u) < V(u+1) < V(u+2)$) するものを除き，それらの中で優先度の最も高い予測用窓を選択する．

3. 予測実験

本手法を用いてプロセッサ負荷予測の比較実験を行った．比較対象の予測手法はパターン検索法，平行移動の検索を可能にした予測手法 (2.1 節)，プロセスや負荷変動傾向の情報を用いた提案手法 (2.1-2.4 節) の 3 つである．また，今回プロセッサの負荷として使用したものは Linux2.4.x の load average であり，その 5 分間の負荷平均を 5 分おきに抜き出したものを負荷系列として用いた．プロセス名と CPU 利用率は抜き出した負荷系列と同時刻に実行されていたものを ps コマンドを用いて収集した．実験に用いたデータは PentiumIII800MHz のプロセッサ 2 台を搭載した計算機 6 台の負荷を加算したもので，データベース検索の際はある時刻 T から 3ヶ月前までのデータを用いた．各パラメータは，予測長 p=12(60 分後まで)，窓幅 D=12(60 分間のデータ)，予測用窓数 k=10，CPU 利用率補正の際の const=1.5 とした．評価方法は予測結果と実測値との各時間における誤差の和 $PredictErr(T)$ を用いた．

$$PredictErr(T) = \sum_{i=1}^{p} |V(T+i) - P(T+i)|$$

およそ 3000 回の予測実験を行った時の $PredictErr$ の平均を表 2 に示す．平行移動した検索のみを用いてもパターン検索法より良い結果が得られているが，プロセスの情報などを用いた提案手法の方がさらに良い予測結果が得られている．これより平行移動とプロセス情報の利用のいずれも予測に有効であると言える．

表 2: 予測結果の平均 $PredictErr$

パターン検索	0.165
平行移動検索のみ	0.118
提案手法	0.090

実際に予測を行った結果を図 1,2 に示す．横軸は時間軸であり便宜上予測開始時刻 T を 0 としている．$t \leq 0$ の部分は検索に用いた予測用窓の負荷変動，$t > 0$ の部分は予測結果である．

図 1 は CPU 利用率の補正を使った予測結果である．この場合時刻 T での負荷は約 2 を示しているが，実は時刻 T 時点で主に実行されているプロセスはないことがプロセス情報から分かっている．そのことを考慮することによって従来手法では難しかった予測が行えている．

図 2 は負荷変動の傾向を考慮した予測結果である．時刻 T 直前での負荷の増加傾向を考慮した予測であり，こ

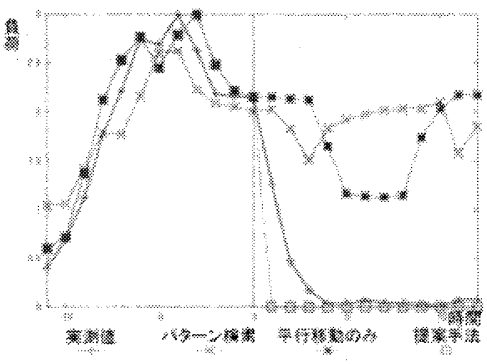

図 1: CPU 利用率の補正を使った予測結果

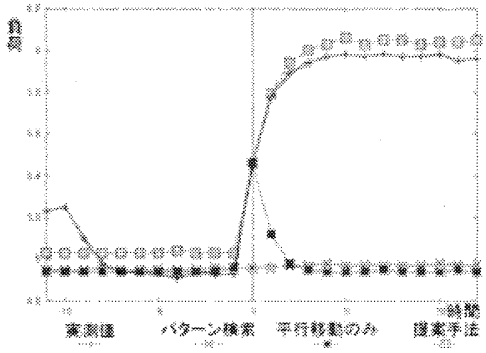

図 2: 負荷変動の傾向を考慮した予測結果

ちらも提案手法では実測値の増加傾向に近い予測結果が得られている．

4. まとめ

本論文では実行プロセスの情報を用いた長期の負荷予測手法を提案した．プロセス名，CPU 利用率，予測直前の負可変動傾向を利用した提案手法において，プロセッサ負荷の長期予測を従来手法よりも高精度に行えることを実験により確認した．

今後はプロセスの実行時間予測の導入，CPU 利用率を用いた予測値補正の改良に取り組み，予測精度の向上を目指していく．

参考文献

[1] R.Wolski et al. “The Network Weather Service:A Distributed Resource Performance Forecasting Service for Metacomputing”, J. of Future Generation Computing System, Vol.15, No.5-6, pp.757-768(1999)

[2] L.Yang et al. “Homeostatic and Tendency-based CPU Load Predictions”, IEEE, Paralell and Distributed Processing Symp. , pp.42b(2003)

[3] 小出，他 “資源情報サーバにおける資源情報予測の評価”，情処論文誌:プログラミング，Vol.42, No.SIG3(PRO10)，pp.65-73(2001)