

プロセス情報を用いた計算機負荷長期予測モデル Long-term CPU Load Prediction using Process Properties

立見 博史[†]
Hiroshi Tatsumi

菅谷 至寛[†]
Yoshihiro Sugaya

阿曾 弘具[†]
Hirotomo Aso

1. はじめに

非均質の計算機が多数集まって構成されるグリッド環境では、その性能を十分に生かすためにアプリケーションを自動的かつ最適に配置する負荷分散システムが必要である。CPUやネットワークの負荷は変動が大きく、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、CPUやネットワークの負荷変動を予測することは、有効な負荷分散を実現する上で大きな助けとなる。(NWS[1][2], RIS[4][5])

CPU負荷予測の研究に関して、Yangら[3]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[5]ではCPU負荷の高精度予測手法として類似法を提案している。類似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、CPU負荷の長期予測を可能にしている。CPU負荷の長期予測を行う利点は、数ステップ先のCPU負荷を考慮した資源配置が可能となることである。

本研究では、CPU負荷としてLinuxのload averageを用いる。UNIXやLinuxでは、計算機にほとんど負荷をかけることなく、load averageのデータを収集できるためである。しかし、CPU負荷としてload averageを用いると、従来手法の類似法では予測直前の急激な負荷変動に対応できないという問題点があった。そこで本研究では、データベースからの類似した負荷変動の検索、プロセス名、プロセスの実行時間、CPU利用率などを考慮し、予測直前の急激な負荷変動にも対応可能なCPU負荷の長期予測手法を提案する。これによって、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負荷系列を $V(0), V(1), \dots, V(T)$ 、予測する負荷系列を $P(T+1), P(T+2), \dots, P(T+p)$ とする。

2.1 負荷変動状態の判断

まず観測された負荷系列 $V(t)$ から、1次階差系列 $V'(t)(=V(t)-V(t-1))$ を作成し、各時刻で1次階差系列の窓 $\mathbf{W}_t(i)(=V'(t-D+1), \dots, V'(t))$ を設定する。 D は窓幅を示すパラメータである。

時刻 T の窓 \mathbf{W}_T 内のデータの標準偏差が閾値 sd 以下のとき、負荷は定常状態であるとする。そのときは時刻 T 以後も負荷 $V(T)$ が継続すると予測し($P(T+i)=V(T)$)、予測を終了する。

それ以外の時、つまり負荷が定常状態でないときは2.2節以降の処理を行う。

2.2 プロセス情報の検索

時刻 T で実行されているプロセスと同一のプロセスが過去にも実行されていた場合、時刻 T 以後の負荷変動は過去に同一プロセスが実行されていたときと類似すると考えられる。従って、まず時刻 T で実行されているプロセスと同一のプロセスが実行されていた時刻を、データベース上から検索する。

もしデータベース上に時刻 T と同一のプロセスが見つかった場合、以後、予測時に用いるデータベースとして同一プロセスが実行されていた区間だけを用い、見つからなかった場合は、データベース全体を用いることとする。

2.3 類似した負荷増減値の検索

1次階差系列の窓 \mathbf{W}_t と窓 \mathbf{W}_T との誤差 $err(t)$ を計算する。

$$err(t) = \sum_{i=0}^{D-1} |\mathbf{W}_t(i) - \mathbf{W}_T(i)|$$

各時刻 t の情報を用いて仮予測を行う。仮予測結果は、 $V(T)$ に時刻 t 以後の負荷増減値を加算したもの、つまり $P(T+i) = V(T) + \sum_{j=1}^i V'(t+j)$ とする。

2.4 仮予測結果の選択

誤差 $err(t)$ が小さい時刻の窓を使った予測が予測値として適していないと考えられる場合もある。そこで不適である可能性の高い仮予測は使用しないこととする。

不適である可能性の高い仮予測は、時刻 T で実行されているプロセス以外のプロセスが新たに発生する予測をしているものとした。時刻 T で実行されていない全く新しいプロセスが発生することを予測するのは、相当の規則性がない限り、非常に困難であり現実的でないためである。

残った仮予測の中から、 $err(t)$ が小さい順に k 個の時刻の窓を使った仮予測を平均したものを、ここまでの予測結果とする。

2.5 予測直前の負荷変動傾向を考慮した予測結果の補正

2.4節までの予測手法では、時刻 T 直前のプロセスの発生、消滅による急激な負荷変動に対応できない場合が多い。そこで、そのような急激な負荷変動に対応するため、実行されているプロセスの情報をを用いて予測結果の補正を行う。補正を行う条件は、時刻 T で実行されているプロセス数 $PN(T)$ と、時刻 $T-1$ で実行されているプロセス数 $PN(T-1)$ 、または時刻 $T-2$ で実行されているプロセス数 $PN(T-2)$ が異なるときである。

それ以外のときは補正をせずに、2.4節の結果を予測値とする。

2.5.1 プロセス数減少のとき

時刻 T の直前にプロセスが消滅した場合($PN(T) < PN(T-1), PN(T-2)$)、時刻 T 以後急激に負荷は減少

[†]東北大学大学院工学研究科

することになる。そこで残ったプロセスの CPU 利用率から各プロセスの負荷を推定し、予測値を補正する。

まず各プロセスの CPU 利用率からそれぞれの負荷を推定する。1つの CPU につき、実行されているプロセスが1つであれば、CPU 利用率 80%以上のプロセスは推定負荷 $EL = 1$ 、そうでないプロセスは $EL = 0$ とする。2つ以上のプロセスが実行されているときは、 $\frac{100}{\text{プロセス数}} \times 0.8\%$ 以上のプロセスは $EL = 1$ 、そうでないプロセスは $EL = 0$ とする。

時刻 T で実行されている全てのプロセスの推定負荷 EL を加算した負荷 EL_{total} が、時刻 T 以後も続くものとして予測結果を補正し ($P(T+i) = EL_{total}$)、予測値とする。

2.5.2 プロセス数増加のとき

時刻 T の直前にプロセスが発生した場合 ($PN(T) > PN(T-1), PN(T-2)$)、時刻 T 以後急激に負荷は増加することになる。そこで時刻 T で動いているプロセスの CPU 利用率から負荷の推定を行い、また、そのプロセスの実行時間を予測し、予測値を補正する。

プロセス負荷の推定法は 2.5.1 節と同様である。

プロセスの実行時間予測では、時刻 T で実行されているプロセスと同一プロセスをデータベースから探し出し、過去 5 回の CPU 実行時間の中央値を、そのプロセスの実行時間予測値として用いる。各プロセスは予測された実行時間だけ推定負荷を発生させるものとして予測結果を補正し、予測値とする。

なお、過去に同一プロセスがなかったプロセスは、予測区間中ずっと推定負荷を発生させるものとする。

3. 予測実験

本手法を用いて CPU 負荷予測の性能評価実験を行った。比較対象の予測手法として平均値法、中央値法、最近接過去法、自己回帰法 (AR モデル)、類似法 ([5]) を用い、増減値パターン検索法 (2.1-2.4 節)、プロセス情報を用いた提案手法 (2.1-2.5 節) の性能を評価する。

平均値法、中央値法では、時刻 $T-p+1$ から時刻 T までの負荷の平均値、中央値が予測区間中継続すると予測し、最近接過去法では、時刻 T の負荷 $V(T)$ が予測区間中継続すると予測する手法である。また、自己回帰法ではデータベースの負荷系列を予測長と同じ長さの区間ごとに分け、その区間ごとの平均を求めた新たな時系列に対して自己回帰法を適用することで、予測区間の平均負荷を予測している。そして、その予測された平均負荷が予測区間中継続すると予測する。

今回プロセッサの負荷として使用したものは Linux 2.4.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 Xeon, 2.4GHz のプロセッサ 2 台を搭載した計算機 4 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 $p = 6$ (30 分後まで), 12 (60 分後まで), 窓幅 $D = 3$ (15 分間のデータ), 標準偏差の

表 1: 窓 W_T が定常時の PredictionErr の平均 (AVG) と標準偏差 (STD)

予測手法		$p = 6$	$p = 12$
平均値	AVG	0.0948	0.1732
	STD	0.3198	0.4677
中央値	AVG	0.0873	0.1657
	STD	0.3149	0.4940
最近接過去	AVG	0.0821	0.1344
	STD	0.3097	0.4251
自己回帰	AVG	0.2772	0.4110
	STD	0.4969	0.7328
類似	AVG	0.1554	0.2588
	STD	0.4556	0.6277
パターン検索	AVG	0.0821	0.1344
	STD	0.3097	0.4251
提案手法	AVG	0.0821	0.1344
	STD	0.3097	0.4251

表 2: 窓 W_T が非定常時の PredictionErr の平均 (AVG) と標準偏差 (STD)

予測手法		$p = 6$	$p = 12$
平均値	AVG	1.0784	1.2822
	STD	0.9998	1.2509
中央値	AVG	1.1281	1.3939
	STD	1.0923	1.5081
最近接過去	AVG	0.6537	0.7799
	STD	0.7289	0.8131
自己回帰	AVG	1.4334	1.4376
	STD	1.5118	1.4430
類似	AVG	0.8204	1.0457
	STD	0.8953	1.0262
パターン検索	AVG	0.6925	0.8219
	STD	0.7209	0.7953
提案手法	AVG	0.6171	0.7438
	STD	0.7456	0.8358

閾値 $sd = 0.16$, 2.4 で使用する仮予測用窓数 $k = 5$, 自己回帰法の係数は 10 個とした。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平均 PredictionErr(T) を用いた。

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

およそ 3000 回の予測実験を行った時の PredictionErr の平均、標準偏差を表 1, 2 に示す。窓 W_T が定常であった時は全体の約 9 割で、非定常であった時は全体の約 1 割であった。定常のときの提案手法とパターン検索法は最近接過去法と同等であるので、それぞれの PredictionErr は等しくなる。

表 1 より、窓 W_T が定常のときは、どの予測手法を用いてもかなり正確に予測ができていることが分かる。これより、窓 W_T が定常状態であれば、時刻 T 以後もほとんど大きな負荷変動は起きていなかったと言える。

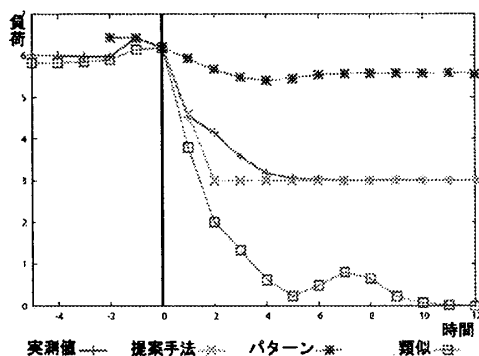


図1: 負荷が急激に減少するときの予測結果

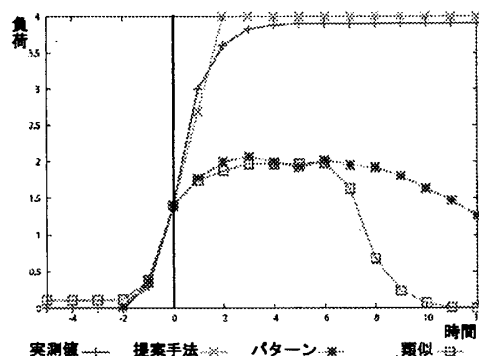


図2: 負荷が急激に増加するときの予測結果

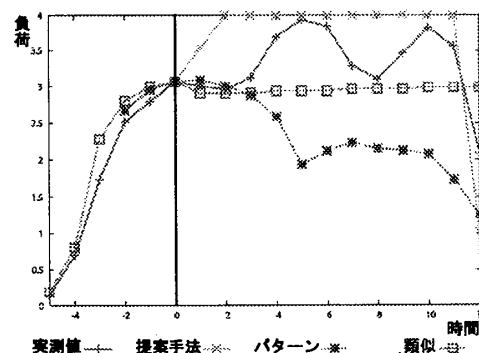


図3: プロセスが途中で終了するときの予測結果

表2より、窓 W_T が非定常のときは提案手法の平均 $PredictionErr$ が最も小さかった。また、提案手法の $PredictionErr$ の標準偏差も比較的小さかったことから、提案手法は他手法と比較して、安定して予測精度が高いと言える。これらの結果から、予測直前のプロセスの変化を考慮した予測によって、従来手法では難しかった負荷の急激な変化にも対応した予測ができたと考えられる。

実際に予測を行った結果を図1,2,3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は負荷が急激に減少するときの予測結果である。

この場合時刻 T での負荷は約6を示しているが、実は時刻 T 時点で実行されているプロセスが減少した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、提案手法では負荷がどこまで減少するのかということも推定できる。

図2は負荷が急激に増加するときの予測結果である。時刻 T 直前で新たなプロセスが投入されたことが分かっている。負荷の増加傾向とプロセス情報を利用することで、提案手法では負荷の急激な増加に対応した予測ができており、また、負荷がどこまで増加するのかということについても推定できている。

図3はプロセスが途中で終了するときの予測結果である。提案手法ではプロセスの実行時間予測も用いているため、負荷の上限値を予測できていると同時に、時刻 $t = 12$ 付近でプロセスが終了することも予測できている。

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、プロセスの実行時間、CPU利用率を利用した提案手法において、CPU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精度向上に取り組み、予測精度の向上を目指していく。また、予測値を実際の負荷分散に利用していくために、予測値の提供方法についても検討していく。さらに、実際の負荷分散で提案手法を利用することで、より効率的な資源配置が可能であることを実証していく。

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] R.Wolski et al. "Dynamically forecasting network performance using the Network Weather Service", J. of Cluster Computing, Vol.3, No.4, pp.265-280(2000)
- [3] L.Yang et al. "Conservative scheduling: Using predicted variance to improve scheduling decisions in dynamic environments", Proc.Supercomputing 2003, 11 15-11, Phoenix, Arizona(2003)
- [4] 小出, 他 "資源情報サーバにおける資源情報予測の評価", 情報処理学会論文誌:プログラミング, Vol.42, No.SIG3(PRO10), pp.65-73(2001)
- [5] 平, 他 "資源情報サーバにおける予測手法である類似法の改良と評価", 情報処理学会研究報告, 2004-HPC-100, pp.25-29(2004)
- [6] 立見, 他 "プロセス情報と負荷変動傾向を用いた計算負荷長期予測モデル", FIT2004 第3回情報科学技術フォーラム, A-028(2004)