B-013

GPU サイクル共有システムのための遊休時間予測手法の比較
Comparison of Idle Time Prediction Methods for GPU-Cycle Sharing Systems

南 翔太† Shota Minami
伊野 文彦† Fumihiko Ino
萩原 兼一† Kenichi Hagihara

1. はじめに

サイクル共有システム[1]とは，ネットワーク上の計算機における遊休サイクルを活用し，科学計算を高速化するシステムである．その計算資源として，従来のCPUのみならず，グラフィクス処理用のアクセラレータGraphics Processing Unit（GPU）[2]が注目されている[3]．GPUは，CPUと比べてメモリ帯域幅ならびに浮動小数点演算性能が高く，少ない計算資源で大きな加速を期待できる．

GPUを活用するサイクル共有システム[3]の特徴として，数秒程度で実行を完了できるジョブを大量に実行できる点が挙げられる．一般に，CPUに基づくシステムは数時間程度の実行時間を要するジョブを対象としてきたため，短い遊休時間であっても積極的にジョブを投入する点はGPUを活用するシステムにおける特徴の1つである．したがって，サイクル共有システムの実行効率を高めるためには，長い遊休時間だけでなく，秒単位の短い遊休時間を活用する必要がある．

サイクル共有システムの実行効率を高めるために，Ren ら[4]はCPUにおける分単位の遊休時間を予測している．この予測により，より長く遊休するであろう計算資源に対してジョブを割り当てることができる．彼らは，将来における計算資源の状態が現在の状態およびその経過時間に依存すると考え，準マルコフ過程（SMP：Semi-Markov Process）に基づく予測手法を提案している．しかし，この予測手法は1分以上の長い遊休時間を対象としていて，秒単位の短い遊休時間に対する有用性は定かではない．

そこで，本研究はGPUを活用するサイクル共有システムを対象として，秒単位の短い遊休時間の長さを予測するために，SMPを含む複数の予測手法を比較する．比較対象は，SMPモデル，自己回帰モデルおよび直前p個の平均に基づく予測手法である．これらの予測精度を，実際のサイクル共有システムで得た実行ログを基に調べる．

2. 関連研究

サイクル共有システムにおける計算資源の選択を支援するために，Ren[4]らは，CPU 使用率やメモリ使用率を測定し，その分析結果に基づいて SMP を用いた予測手法を提案している．この手法は，分刻みに離散化された遊休時間を扱う．実験では，SMP による予測手法および RPS toolkit[5]が提供する時系列データに対するいくつかの線形予測手法を比較し，SMP による予測手法は予測誤差が他の手法よりも低いことを示している．さらに，Condor[1]の資源選択手法[6]と比較して，実行時間の長いタスクについてタスクの失敗が少なく，高い予測精度を発揮することをシミュレーションで示している．

† 大阪大学大学院情報科学研究科 Graduate School of Information Science and Technology, Osaka University

Rood ら[7]は，直近数日または数時間の使用状況の履歴を基に予測する．また，状態遷移の確率を求める際に，新しい履歴もしくは 1 日の中での時間帯に基づいて，履歴中の状態遷移回数に重み付けする．これにより，予測精度の向上を図る．結果として，SMP に基づく手法に比べて，精度を最大で 4.6%向上させた．スケジューラへの適用実験において，全ジョブを完了させるまでの時間を既存の予測手法に比べて 20%削減した．

3. 予測手法

本章では，比較に使用した予測手法の本実験における適用方法を説明する．

3.1 予測手法の適用方法

SMP は現在の状態とその経過時間を参照する状態過程である．今回の実験では現在の状態を直前 N 個の状態として定義している．

遊休時間の遷移確率を，遷移の出現頻度から計算する．PC ごとに計算資源の使用状況が異なることを反映するために，遷移確率はすべての PC で統一するのではなく PC ごとに独立した遷移確率を与える．得られた遷移確率のうち最も遷移確率の高いグループ（後述）を，遷移先として選択する．最も遷移確率の高い遷移先が複数ある場合，遊休時間の分布を反映し，それらの遷移先のうち，最も遊休時間が短い状態を遷移先とする．

自己回帰モデルは以下の式に従う．

$$Y_t = c + \sum_{i=1}^{p} \varphi_i X_{t-i} + \varepsilon_t \quad (1)$$

X_tは実際の遊休時間，Y_tは予測結果，c は定数項，φ_iはモデルのパラメータ，ε_tは時系列データの分散に従うホワイトノイズを表している．φ_iは Yule-Walker 法により求める．t の値は t 番目の遊休時間であることを表す．

定数項は単純化のために省略できる．ただし，$c = 0$ とする場合，時系列データの平均も 0 となる．遊休時間の平均は 0 ではない．しかし，各遊休時間の値から遊休時間の平均値を引くことにより，平均 0 の時系列データを得ることができる．したがって，式(1)を変形すると以下のようになる．

$$Y_t = \sum_{i=1}^{p} \varphi_i (X_{t-i} - \bar{X}) + \bar{X} + \varepsilon_t \quad (2)$$

$\bar{X}$はX_tの平均値である．

遊休時間は 1 秒から 1 万秒を超えるものまであるため，分散が大きくなる．時系列データの分散が大きい場合，ε_tが極端に大きな値をとるため，予測が上手く行かない．そこで，時系列データを底が 10 である対数に変換する．このことにより，分散を小さくすることができる．そのため，ε_tが極端な値をとることはない．この時の式(2)は以下のように変形する．

$$Y'_t = \sum_{i=1}^{p} \varphi_i (X'_{t-i} - \bar{X'}) + \bar{X'} + \varepsilon_t \quad (3)$$

$$X'_t = \log_{10} X_t \quad (4)$$

$$Y'_t = \log_{10} Y_t \quad (5)$$

$\bar{X'}$は遊休時間の対数X'_tの平均値である．式(3)よりY_tの対数Y'_tを得られる．したがって，Y'_tの逆変換を予測の値とする．

直前 p 個の平均を使用した予測は，以下の式で表せる．

$$Y_t = \sum_{i=1}^{p} X_{t-i} \ / \ p \quad (6)$$

3.2 遊休時間のグループ分け

SMP では，実行ログを入力として遷移確率を得る．ここで，遊休時間の長さ 1 秒ごとに遷移確率を得る場合，1 秒あたりの標本数が減少するため，遷移確率の信頼性が低下する．そこで，遷移確率を得るために十分な標本数を確保するため，遊休時間をいくつかのグループに分類する．

まず，遊休時間の分布が有する特徴について説明する．遊休時間の分布は冪分布にしたがう（図 1）．ただし，他の遊休時間と比較して突出して出現回数が多い特異点が存在する．この特異点は予測精度を低下させる可能性がある．したがって，特異点の周辺を独立したグループとして分割することにより，予測精度の向上を図る．

図2に，グループ作成のアルゴリズムを示す．アルゴリズムの入力は，各グループの要素数における閾値α_s(1≦ s ≦ n)，特異点の閾値β，遊休時間の長さe_t(1≦t ≦Z)，e_tの数p_tである．出力はグループの数m，およびグループ内における最短の遊休時間g_mである．ただし，α_s < 1(1≦ s ≦ n)，合計値$\sum_{k=1}^{n} \alpha_k = 1$とする．グループの数は少なくとも$n$個以上になる．

出力されるグループ数mは入力時のグループ数nを上回ることがある．これは，特異点が検出されたことにより，指定したグループ数nより多くのグループが作成されるためである．グループxに含まれる遊休時間の範囲はグループにおいて最短の遊休時間g_xにより表現可能である．したがって，各グループxに含まれる最短の遊休時間のみを出力する．

最初に，どの遊休時間も含まないグループを 1 つ作成する．グループには閾値が設定されており，遊休時間の数が閾値を超えるまで作成したグループに遊休時間を追加していく．閾値を超えるとグループの作成を終了し，すべての遊休時間がグループ分けされているか確認する．すべてグループ分けされていなければ新たなグループを作成し，すべての遊休時間がグループ分けされるまで繰り返す．

グループに遊休時間を追加する際に，特異点となる遊休時間を検出する．特異点を検出する場合，遊休時間の数が閾値に満たない場合においてもグループの作成を終了し，新たなグループを作成する．特異点となる遊休時間が続く限り，グループに遊休時間を加えていく．特異点となる遊休時間が途切れた場合，そのグループの作成を終了する．

ある長さの遊休時間より短い遊休時間 5 つの出現回数を平均したものに，閾値βをかけた数を特異点の閾値 w とする．w を超える遊休時間より長い遊休時間のうち，初めて w を下回る遊休時間よりひとつ短い遊休時間までを特異点とする． 21 行目以降は，2 行目の for ループでグループ分けできなかった遊休時間のグループ分けをする．

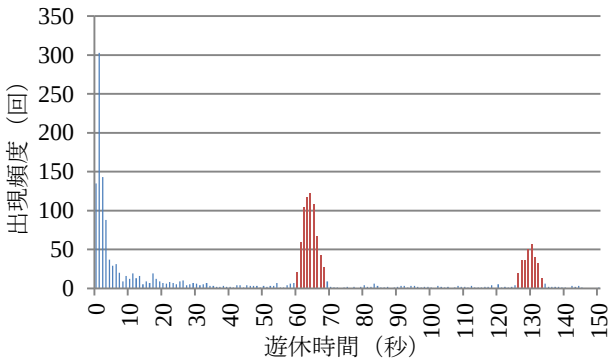

図 1: 遊休時間の分布

入力：グループの閾値α_s(1≦ s ≦n)，特異点の閾値β，遊休時間の長さe_t(1≦t ≦Z)， $p_t = e_t \ / \ \sum_{k=1}^{Z} e_k$ (1≦t ≦Z)

出力：グループの数 m(m≧n)，グループ内で最短の遊休時間$g_1, g_2, \ldots, g_m$

```
1:  i = 1;j = 1;m = 1;
2:  for (i = 1; i <= n; i++){
3:      P = 0;                          //グループに含まれるe_tの合計
4:      g_m = e_j;
5:      while ((P < α_i) && (j < Z)){
6:          w = Σ_{k=j-5}^{j-1} p_k/5 * β;      //特異点の閾値を計算
7:          if (p_j > w){
8:              m++; i++;
9:              g_m = e_j;
10:             while (p_j > w){
11:                 j++;
12:             }
13:             P = 1;
14:         }else{
15:             P += p_j;
16:             j++;
17:         }
18:     }
19:     m++;
20: }
21: while (グループ分けされていないe_tがある) {
22:     i = n に固定し，5〜18 行を実行
23: }
```

図 2: グループ分けのアルゴリズム

4. 評価実験

実験では，研究室の学生が日常的に使用している6台のPCから遊休時間の実行ログを取得した．表1に実験環境を示す．実行ログは，計算資源の状態（遊休または繁忙），状態の遷移時刻，およびその持続時間を含んでいる．グループ分けのパラメータはn = 13に固定し，閾値もすべて固定した．表2に用いた閾値の一覧を与える．特異点の閾値はβ = 4とする．表3に，実行ログの取得期間をPCごとに示す．取得したログのうち前半をSMPモデルの学習，および自己回帰モデルにおけるパラメータφ_iの計算に用い，後半

を予測対象に使用した．なお，SMPでは1≦N≦5とし，直前p個の平均では1≦p≦4とした．

図 3 に SMP を用いたときの予測精度を示す．SMP の予測精度は 20%～45%であり，N=2 のときに最も高い．この理由は，3 個以上前の遊休時間は予測する遊休時間との関係性がないためと考えられる．以降では，最も精度の高い N=2 のときにおける予測結果を比較対象とする．

図 4 に SMP および直前 p 個の平均を用いたときの予測精度を示す．直前 p 個の平均を用いた場合，予測精度は 20%～40%であり，p=1 のときに最も高い．しかし，これらの予測精度は SMP のものよりも低い．実際に，SMP および直前 1 個の平均による予測精度の優劣を有意水準 5%で t 検定により検定したところ（表 4），半数の PC において SMP の予測精度が統計的に優れていた．ただし，残りの PC においては予測精度に関して統計的な差異を確認できなかった．この理由は，直前の遊休時間と同じような長さの遊休時間が続くような PC に対しては，直前 p 個の平均を使う場合と似た予測をするためと考えられる．また，p の値が増えるにつれ，SMP が優位な傾向があった．この理由は，2 個以上前の遊休時間は予測する遊休時間と異なる値の可能性が大きいためと考えられる．

図 5 に SMP および自己回帰モデルの予測精度を示す．自己回帰モデルの予測精度は 14%～32%に留まっており，すべての PC において SMP の予測精度が高い．表 4 は SMP および自己回帰モデルによる予測結果の優劣を示している．t 検定の結果， SMP は自己回帰モデルよりも優れている．この理由は，自己回帰モデルによる予測は冪分布よりも正規分布に従う傾向にあるためと考えられる．

5. おわりに

本研究では，GPU を活用するサイクル共有システムにおいて，遊休時間の長さを予測することを目的として，SMP，自己回帰モデルおよび直前 p 個の平均を用いた予測手法の予測精度を比較した．

実験では，SMP による予測が自己回帰モデルよりも優れており，直前 p 個の平均による予測と比較しても半数のユーザにおいて優位となることがわかった．しかし，SMP の予測精度は 20%～45%に留まっており，予測精度のばらつきも大きい．今後の課題としては，さらに精度を高める予測手法の考案が挙げられる．

表 1: 実験環境

OS	Windows 7 Professional 64bit
CPU	Intel Core i7-3770K 3.50 GHz
主記憶容量	16 GB
GPU	NVIDIA GeForce GTX 680
ビデオメモリ容量	2048 MB
ドライバ	310.90
CUDA	5.0

表 2: グループの閾値一覧

i	1	2	3	4~5	6~8	9	10~11	12~13
α_i	0.25	0.2	0.15	0.1	0.05	0.02	0.01	0.005

表 3: ログの取得期間

	実験期間	状態数	起動時間（h）	遊休時間（h）
PC1	17 日	1,320	40	39
PC2	41 日	12.537	235	221
PC3	70 日	3,288	535	533
PC4	43 日	968	269	259
PC5	17 日	7,707	104	99
PC6	41 日	3,233	389	379

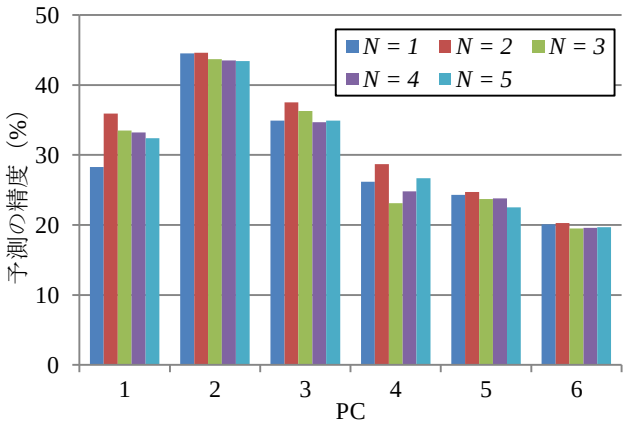

図 3:SMP（1≦N≦5）の予測精度

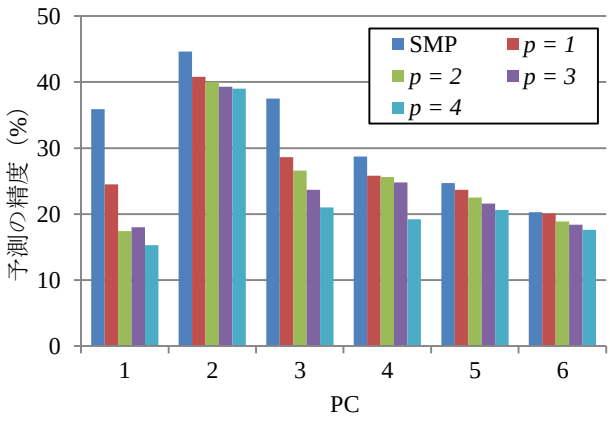

図 4: SMP と直前 p 個の平均を用いた予測精度

表 4: SMP および直前 p 個の平均の比較結果（有意水準 5%）

	p= 1	p = 2	p = 3	p = 4
PC1	SMP	SMP	SMP	SMP
PC2	SMP	SMP	SMP	SMP
PC3	SMP	SMP	SMP	SMP
PC4	N/A	N/A	N/A	SMP
PC5	N/A	SMP	SMP	SMP
PC6	N/A	N/A	N/A	SMP

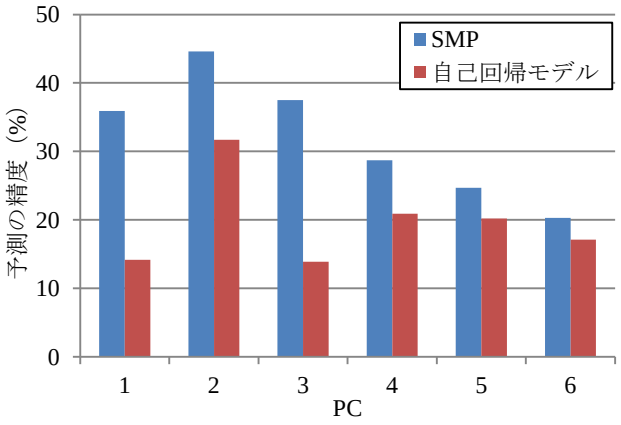

図 5: SMP と自己回帰モデルの予測精度

表 5: SMP および自己回帰モデルの比較結果

	有意水準 5%	有意水準 1%
PC1	SMP	SMP
PC2	SMP	SMP
PC3	SMP	SMP
PC4	SMP	N/A
PC5	SMP	N/A
PC6	SMP	N/A

謝辞

本研究の一部は，JST CREST「進化的アプローチによる超並列複合システム向け開発環境の創出」，科研費 23300007，および 23700057 の補助による．

参考文献

[1] M.J. Litzkow, M. Livny, and M.W. Mutka, Condor - A Hunter of Idle Workstations, In *Proc. 8th Int'l Conf. Distributed Computing Systems (ICDCS'88)*, pp.104–111, June 1988.
[2] J.D. Owens, M. Houston, D. Luebke, S. Green, J.E. Stone, and J.C. Phillips, GPU Computing, In *Proc. IEEE*, Vol. 96, No. 5, pp.879–899, May 2008.
[3] F. Ino, Y. Munekawa, and K. Hagihara, Sequence Homology Search Using Fine Grained Cycle Sharing of Idle GPUs, *IEEE Trans. Parallel and Distributed Systems*, 23(4):751–759, April 2012.
[4] X. Ren, S. Lee, R. Eigenmann, and S. Bagchi, Prediction of Resource Availability in Fine-Grained Cycle Sharing Systems Empirical Evaluation, *J. Grid Computing*, 5(2):173–195, March 2007.
[5] Peter A. Dinda, RPS: An Extensible Toolkit for Resource Prediction in Distributed System, 2011. http://www.cs.northwestern.edu/_RPS/.
[6] Douglas Thain, Todd Tannenbaum, and Miron Livny, Distributed computing in practice: The condor experience, Concurrency and Computation: Practice and Experience, 17(2-4):323–356, Feb. 2005.
[7] Brent Rood and Michael J Lewis, Resource Availability Prediction for Improved Grid Scheduling, In IEEE 4th Int'l Conf. eScience (eScience'08), pp.711–718, Dec. 2008.