

大規模オンラインシステムの メモリ使用状況とページング予測

海老根 秀明 摩須 稔 高橋 雅宏 村松 洋
(富士通株式会社)

1. 概要

仮想記憶システムのメモリの振る舞いについては、ワーキングセットモデルが提案され、これに基いた多数の解析例が報告されている。しかしワーキングセットモデルは比較的小規模なバッチやTSSのシステムの理解には有効であるが、大規模なオンラインシステム等については、システムの振る舞いを理解する手段とはなりえていない。

大規模システムでのメモリの使用状況の分析例としては、Petrella等⁽⁴⁾によるMVSのオンラインシステムの報告があるが、経験的な理解に留まっている。

本稿では、大規模なオンラインシステムのメモリ使用状況を表現するための概念を述べ、この概念に基いた実測例を示す。この概念の量はいづれも、実システム上で測定可能な量である。

またこの概念の適用としてメモリ増設後のページフォルト回数の予測を行う。オンラインシステムの実際の運用においては、新規業務の追加や接続端末の増設により、ページフォルトが多発することがしばしば生じる。しかし、どの程度実記憶を増設すればよいかを判断するのは必ずしも容易ではない⁽³⁾。本文では上記概念をこの問題に適用すると、実用上十分な精度で実記憶の増設後のページフォルト回数の予測ができ、上記概念が有効であることを述べる。

2. メモリ使用状況把握のための概念

メモリ使用状況を把握するための概念は、さまざまな提案がある。本文では、各論理ページの実メモリ使用状況を把握する立場から、以下で述べる概念を用いる。図1に、ある論理ページの動き（ページフォルトを発生しメモリ上にページインされる。その後参照を繰り返す。最後に参照をしてから暫くメモリ上に存在する。しかしその後実メモリを奪われページアウトする）を示し、以降で使用される概念の定義を述べる。

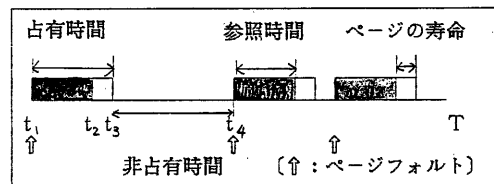


図1 論理ページの動き

<定義>

- ・メモリ占有時間：実メモリを占有していた時間
(占有時間) $(t_3 - t_1)$
- ・メモリ非占有時間：実メモリを奪われてから次に
(非占有時間) ページインするまでの時間 $(t_4 - t_3)$
- ・参照時間：ページインしてから最後に参照
するまでの時間 $(t_2 - t_1)$
- ・ページの寿命：最後に参照してから実メモリを
奪われるまでの時間 $(t_5 - t_4)$
- ・常駐率：ある論理ページが実メモリ上に
存在した確率

3. 大規模オンラインシステムのメモリ使用状況

3.1 オンラインシステムの特長

オンラインシステムではTSSやバッチなどと違い、メッセージはランダムに到着する。つまりメッセージ（電文）は前のメッセージには影響されず独立と考えてよい。

また一つのメッセージに対する仕事は固定的であり、その種類も少ない場合が多い。つまり同一の動作を数多く繰り返すのが特長である。

またオンラインシステムでは、大量のメッセージ（電文）を処理する必要があるなどの理由により、性能を非常に重視する。このためモジュールのローディングなどのオーバヘッドを少なくするため、予め必要なプログラムをメモリ上に展開する場合が多い。つまりメモリの内容のダイナミックな変更は少ない。

3.2 オンラインシステムのメモリ使用状況

(a)測定方法

テストは別のシステムからメッセージ（電文）を発生させる方式を用いた。測定時にはメモリ使用状況の変化を調べるためにオンラインアプリケーションの一部を止めている。〔以下停止前/停止後と表す〕

測定及び解析ツールとして以下を用いた。

①SEAP/DR（メモリ系情報収集サンプラー）

主に実メモリを管理するテーブルの情報を収集する。

②GTF（イベントトレーサ）

主にページフォルト情報を収集する。

③TRAP/MEMORY

①、②データの解析ツール

(b)全体状況

システムは約 9万件/H（25件/S）の処理を行っており、停止したアプリケーションの処理件数は、1800件/H（0.5件/S）である。

また実メモリ量は10MBであるが、停止前のシステムのページフォルト率は26回/秒であり、すでにメモリ不足になっている。

(c)メモリ非占有時間分布

メモリ非占有時間分布を示す、アプリケーションの停止前のグラフは急な（超）指数関数型をしている。このようにページフォルトが多発している環境ではメモリ非占有時間の短いものが多い。これはメモリが足りない状況では、メモリの占有時間が短く、しばらく参照しないとすぐに実メモリを奪われていることを示している。当例では非占有時間の7～8割は30秒以内である。つまり、実メモリを奪われてから30秒もたないうちに再度メモリのアクセスがあったことを示している。

逆に、メモリが充分なときは、最後に参照されたからも長い時間メモリ上に残っている。つまりページの寿命が長い場合次にアクセスがあった場合ページフォルトを起こさずに済む。

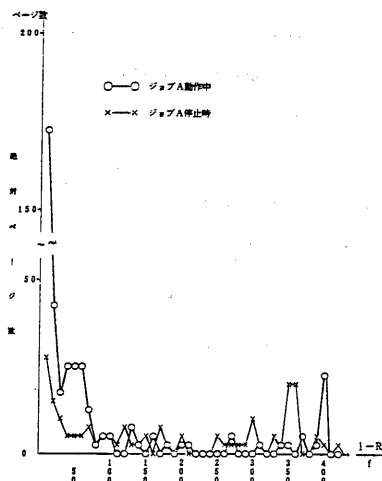


図2 非占有時間分布

(d)常駐率の分布

常駐率は論理ページが実メモリを占めていた確率である。つまり良く参照されるページは常駐率は高く、逆に参照されないページの常駐率は低い。

図3に示すように、メモリが不足している状態(停止前)でも、常駐率の100%のものは全体の約6割を占める。つまり、参照頻度の高いページはその殆どが常にメモリ上に乗り、残り4割の実メモリを多数の仮想ページが奪いあう。そこでページフォルトが発生することになる。

停止中の場合は実に8割ものページの常駐率が100%である。

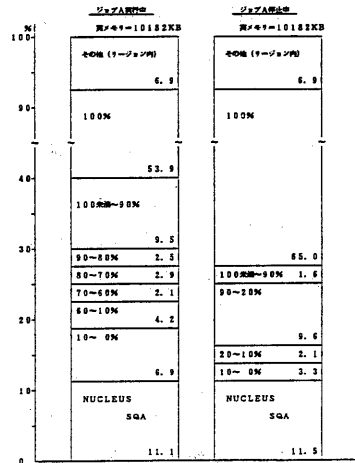


図3 常駐率分布

(e)メモリ占有時間の分布

メモリ占有時間は論理ページが実メモリを占有していた時間である。停止前ではユーザのアプリケーションは約30秒しか占有していない。しかし停止後では約70秒も占有している。

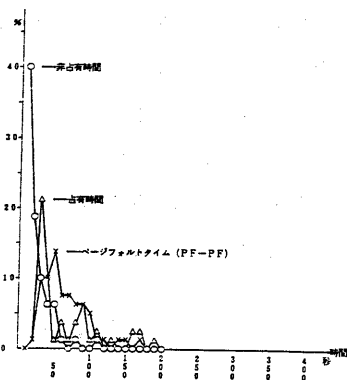


図4 占有時間分布 (ジョブ停止前)

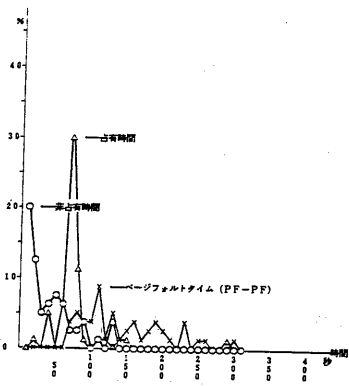


図5 占有時間分布 (ジョブ停止後)

4. メモリ増設後のページング予測方法

メモリ増設後のページフォルト回数の予測方法を示す。但し、メモリ増設前後での負荷の変化（処理件数の増加等）はないものとする。

4.1 仮定

(a)：メモリ増設前後でのメモリ参照間隔に変化はない。

(メモリ増設前のページフォルト発生間隔でのメモリ参照は増設後も変化しない。)

(b)：論理ページの割当は変化しない。

(論理ページに割り与えられたロードモジュールは変更されない。(LOAD/DELETEされない。)
またデータ領域は解放されない。(GETMAIN/FREEMAINされない。)

4.2 基本的な考え

メモリを増設すれば、ページの寿命が延び、その結果ページフォルトの発生を解消する。

ページの寿命の延びの影響は以下の2種類に分けられる。

(a)：ページの寿命が延び、ページフォルトを解消する。

(b)：ページの寿命が延びたが、ページフォルトを解消しない。

図6にその様子を図示する。

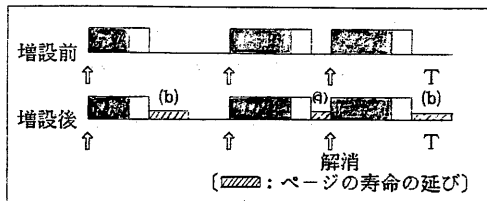


図6 メモリ増設後の変化

4.3 定式化

以下各種記号を定義する。

R_{ai} ：常駐率

$P_{ai}(t)$ ：メモリ非占有時間確率分布

F_{ai} ：ページフォルト率 (回/秒)

M ：実メモリ量 (ページ数)

L_{ai} ：ページの寿命の延び (秒)

(添字は以下を示す)

i ：ページ番号

a ：メモリ優先順位グループ番号

(メモリ増設後の変数 X の値、変化量は以下と表す)

X' ：メモリ増設後の変数 X の値

ΔX ：メモリ増設後の変化量

論理ページ i に着目する。メモリ増設後の常駐率 (R) を式(1)で表す。

$$R'_i = R_i + \Delta R_{ai} \quad (1)$$

常駐率の延び (ΔR_{ai}) はページの寿命の延びである。

ページの寿命の延びよりも、メモリ非占有時間が短い場合は非占有時間だけページの寿命が延びる。これが式(2)の第一項である。また非占有時間がページの寿命の延びよりも長い場合は、ページの寿命だけ延びる。これが式(2)の第二項である。

(右辺の第一項は図6の(a)に対応し、第二項は(b)に対応する)

$$\Delta R_{ai} = F_{ai} \left\{ \int_0^{L_{ai}} t P_{ai}(t) dt + \Delta L_{ai} \int_{L_{ai}}^{\infty} P_{ai}(t) dt \right\} \quad (2)$$

実メモリ量 (M) は、常駐率の和に等しいから

$$M = \sum_a \sum_i R_{ai} \quad (3)$$

従って、メモリ増設後の実メモリ量と常駐率の関係は、式(4)で表せる。

$$M + \Delta M = \sum_a \sum_i (R_{ai} + \Delta R_{ai})$$

$$= M + \sum_a \sum_i \Delta R_{ai} \quad (4)$$

メモリ増設量 (ΔM) に対応するページの寿命の延び (ΔL_i) は式(5)で求められる。

$$\Delta M = \sum_a \sum_i F_{ai} \left\{ \int_0^{\Delta L_i} t P_{ai}(t) dt + \Delta L_i \int_{\Delta L_i}^{\infty} P_{ai}(t) dt \right\} \quad (5)$$

またメモリ優先順位に比例してページの寿命が延びると考えて良いから

$$\Delta L_i = n_a \Delta L \quad (6)$$

(n_a = グループ a の優先度 / 基本優先度)

よって式(5)は

$$\Delta M = \sum_a \sum_i F_{ai} \left\{ \int_0^{n_a \Delta L} t P_{ai}(t) dt + n_a \Delta L \int_{n_a \Delta L}^{\infty} P_{ai}(t) dt \right\} \quad (7)$$

よってメモリの増設後のページフォルト率 (F^*)

はページの寿命が非占有時間より長い場合であるから以下の式で求められる。

$$F^* = \sum_a \sum_i F_{ai} \int_{n_a \Delta L}^{\infty} P_{ai}(t) dt \quad (8)$$

また、メモリ増設により削減されるページフォルト率は、以下の値である。

$$\sum_a \sum_i F_{ai} \int_0^{n_a \Delta L} P_{ai}(t) dt \quad (9)$$

4.4 応用

(a) 応用その 1

全ての論理ページのメモリ優先度が同一で且つ、非占有時間分布が同一と仮定すると以下の式でメモリ増設後のページフォルト率が求められる。

$$\Delta M = F \left\{ \int_0^{\Delta L} t P(t) dt + \Delta L \int_{\Delta L}^{\infty} P(t) dt \right\} \quad (10)$$

$$F^* = F \int_{\Delta L}^{\infty} P(t) dt \quad (11)$$

(b) 応用その 2

各論理ページ毎に非占有時間確率分布を求め代りに、非占有時間の平均値 (N_i) を用いる。各ページ毎の非占有時間の平均値は、常駐率とページフォルト率より求まる。

$$N_i = (1 - R_i) / F_i \quad (12)$$

この時メモリ増設後のページフォルト率は

$$\Delta M = \sum_i F_i N_i + \Delta L \sum_i F_i \quad (13)$$

$$F^* = \sum_i F_i \quad (14)$$

$$\{l\} = \{i : N_i \leq \Delta L\} \quad \{m\} = \{i : N_i > \Delta L\}$$

4.5 算出アルゴリズム

応用その 2 の方法 (平均非占有時間を用いる方法) の算出アルゴリズムを示す。

(ステップ 1)

各論理ページ毎に以下の値を計算する。

- ・常駐率 (R)
- ・非常駐率 ($1 - R$)
- ・ページフォルト率 (F)
- ・非占有時間 (N)

また論理ページ数の合計値を (K) とする。

(ステップ 2)

非占有時間の昇順に論理ページ (J) を並べる。

(ステップ 3)

i) 論理ページ (J) に対し以下の値を計算する。

$$\sum_{j=1}^J (1 - R_j) + \sum_{j=J+1}^K F_j N_j$$

ii) i) で求めた値が増設メモリ量 (ΔM) に等しくなければ (または越えなければ)

$$J = J + 1$$

とし i) にもどる。

もし等しければ以下の値が増設後のページフォルト率である。

$$\sum_{j=J+1}^K F_j$$

4.6 検証

検証は、メモリ増設の方法はとらずに、ジョブの一部を止める方法を用いた。即ち、止めたジョブが使用していた実メモリ量だけ、残りのジョブが使用できる実メモリ量が増加する。この増加量を増設メモリ量と

考えて、4.5で述べた方法を適用した。

表1 適用例

	実測値	予測値
増設前	12.6回/秒	—
増設後	0.6回/秒	0.9回/秒

本方法は、実測値をかなりの精度で予測出来る。また本方法は、UICによるメモリ管理方式を意識したChow等の方法(1)にくらべ、大規模なシステムでも適用が可能であり、又UICを用いないシステムでも適用が可能であり適用範囲が広い。

5. まとめ

メモリ使用状況を分析するための概念を述べ、その適用として、大規模オンラインシステムにおけるメモリ使用状況を報告した。またこの概念を応用し、メモリ増設後のページフォルト回数の予測方法について述べた。今後、当概念(モデル)の応用や各種システムでの適用を計るつもりである。

(参考文献)

- 1.W.M.Chow and W.V.Chiu
 "Program Behavior Model for Paging System"
 Computer Performance Modeling, North Holland
- 2.W.M.Chow and W.V.Chiu
 "A Performance Model of MVS"
 IBM System Journal No 4, 1978
- 3.M.G.Kienzle and K.C.Sevcik
 "Survey of Analytic Queueing Network Models
 of Computer Systems"
 Conference on Simulation, Measurement and
 Modeling of Computer Systems, 1979
- 4.A.Petrella and H.Farry
 "Simulating Working Sets Under MVS"
 CMG IX Conference Proceedings, 1978
- 5.海老根, 摩須, 高橋
 "大規模オンラインシステムのメモリ使用状況と
 ページング予測"
 情報処理学会26回全国大会