

# メモリー・アロケーションの シミュレーションについて (第1報)

菅原 英 一

On the Simulation of Memory Allocation  
(1st Report)

Eiichi Sugawara

(昭和46年10月30日受理)

## 1. 緒 言

計算機システムにおける Memory 管理は

(1) Memory の効率よい使用

(2) Response Time の短縮

を達成する上で極めて重要であるが、両者は互いに相反する一面を有する。すなわち、効率よい Memory の使用のみに注目すれば Memory Allocation の機能が複雑になり、それに伴って Over Head が大きくなるので Response Time の短縮は望めないことになる。

また、Memory Allocation の方法は計算機が処理すべき仕事の種類によって種々考えられているが、汎用機の場合には考慮すべき要素が余りに多種多様で、System Performance を最適ならしめるような Memory Allocation の Algorithm の開発はなかなかできないのが現状のようである。

ところが、最近の Simulation 技術の急速な発達に伴って、これらの問題を計算機を用いた Simulation によって解析することが試みられている。

本論文においても上記の問題を Simulation によって解析しようとするものである。その第1段階としてRO-LL I/O を考えない単純なシステムにおいて、

(1) 各 JOB の Memory 使用時間

(2) Memory を Reserve するまでに待つ時間

の2点に注目して、要求される Program size に対する待ちの状態を Simulation によって求めたので、それに検討を加えて報告する。

なお、本論文で用いたモデルは、全 Memory を等しい大きさの10個の Partition に分割して、要求される Program sizeは Partition 単位で与えられるものとする。

## 2. シミュレーション・モデル

図1に本研究で用いたモデルを示す。

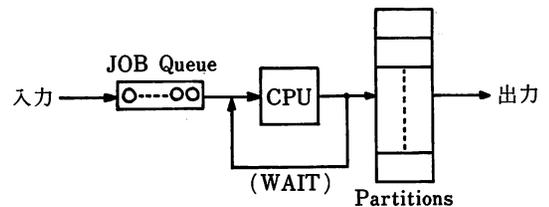


図1 シミュレーション・モデル

ある確率分布でJOBがシステムに到着したとき、ひとまずJOB Queueの中に入り、Queueの先頭にいるJOBからサービスを受ける。したがって、JOBが到着した時点でQueueがあいていなければそのJOBは待ちに入ることになり、さらにQueueの先頭にきても要求するだけのMemory sizeがあいていなければ、サービス中のJOBの終了によって要求したMemory sizeがあくまで待つ。すなわち、First-Come-First-Serviceの思想を基にしている。

このモデルでは、(1)JOBの到着、(2)JOBの要求するPartition数、(3)そのJOBに対するCPUのサービス時間が確率変量として発生するが、それぞれについてもう少し詳しく述べる。

(1) JOBの到着

JOBがシステムに到着する形態は完全ランダム到着と考えるのが妥当であるから、ある到着とそれに続くつぎの到着との間の時間間隔の確率密度関数は平均値 $1/\lambda$ の指数分布 $\lambda e^{-\lambda t}$ となる。したがって、JOBの到着は指数分布の乱数発生によって定義できる。

(2) 要求 Partition 数

1個のJOBは最大全 Partition を要求することが

できるが、要求した Partition 数があいていても、それらが連続していなければ Reserve できないものとする。また、JOBの要求を満たす Location が複数個あるときには First-fit method によるものとする。

JOBが要求する Memory の大きさについては東北大学大型計算機センターにおける昭和46年3月中のTSSのOBJECT PROGRAMのサイズを基にして分布を決定した。表1にその分布を示すが、この分布では最大コア容量の20 bank を要求するプログラムの割合が11.7%を占めている。これは実際には20 bank を使用しなくても、COMMON 文を宣言すると自動的に20bank とられてしまうためである。

BANK数	PROGRAM数	確率密度 (%)
5	1	0.04
6	107	4.45
7	286	11.91
8	460	19.15
9	354	14.74
10	400	16.65
11	131	5.45
12	133	5.54
13	18	0.75
14	21	0.87
15	58	2.42
16	45	1.87
17	48	2.00
18	43	1.79
19	16	0.67
20	281	11.70
TOTAL	2402	100.00

表1 東北大 TSS における OBJECT PROGRAM SIZE の分布

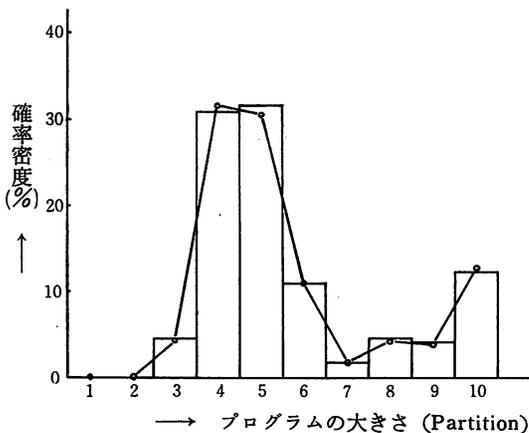


図2 要求 Partition 数の分布

さらに、このモデルでは Memory を10個の Partition に分割しているので、1 Partition を2 bank と仮定すれば、各JOBの要求する Partition 数の分布は図2のようになる。

そこで、このモデルにおいては、システムに到着する各JOBに対応して図2の折線で示すような分布を有する乱数を発生させ、この乱数をもってJOBの要求する Partition 数とした。図2は2000個の乱数を発生させた場合の分布であるが、発生させた乱数は実際のプログラム・サイズにかなりよく合致する。

### (3) サービス時間

ROLL I/Oを考えないモデルではCPUのサービス時間はプログラム・サイズにはほぼ比例するものと仮定している。ただし、JOBの要求する1 Partition 当たりのCPUのサービス時間は明らかでないので、このモデルでは1 Partition 当たりのCPUのサービス時間は6秒とした。

図3に本モデルのゼネラル・フローチャートを示す。このモデルを記述する諸変数を示せば、次のようである。

$RT_i$  = 第  $i$  番目の JOB の到着と第  $(i-1)$  番目の JOB の到着との時間間隔;  $i = 1, 2, \dots, 2000$

$TRT_i$  = 第  $i$  番目の JOB がシステムに到着したときの全到着時間;  $i = 1, 2, \dots, 2000$

$PN$  = 第  $i$  番目の JOB の要求する Partition 数;  $i = 1, 2, \dots, 2000$

$ST_i$  = 第  $i$  番目の JOB のサービス時間 (Memory の使用時間);  $i = 1, 2, \dots, 2000$

$MIN_i$  = 第  $i$  番目の JOB が到着したとき、システム中に存在する JOB のすべてにわたる TT の最小値;  $i = 1, 2, \dots, 2000$

$WT_{i,j}$  = 第  $i$  番目の JOB が到着したとき、要求 Partition 数がないためにシステム中の JOB のうち  $j$  個の JOB のサービス終了まで待つ時間;  $i = 1, 2, \dots, 2000$ ;  $j = 1, 2, \dots, \max 10$

$TWT_i$  = 第  $i$  番目の JOB が到着してからサービスを受けるまでの待ち時間;  $i = 1, 2, \dots, 2000$

$TT_i$  = 第  $i$  番目の JOB のサービス終了までの全時間;  $i = 1, 2, \dots, 2000$

諸変数を上述のように定めれば、第1番目のJOBがシステムに到着するとき、次の関係式が成立するものと

仮定される。

- $RT_1 = 0$
- $TRT_1 = 0$
- $WT_{1j} = 0$
- $TWT_1 = 0$
- $TT_1 = ST_1$

その後続くすべてのJOBに対しては次の関係式がシステムの状態を表わす。

- $TRT_i = TRT_{i-1} + RT_i$
- $WT_{ij} = \text{MIN} - TRT_i$

$$TWT_i = \sum WT_{ij}$$

$$TT_i = TRT_i + TWT_i + ST_i$$

このモデルにおいて、最も大きな制約は、Partitionは連続して Reserve しなければならないということであった。この制約を満たす Algorithm を次に示す。図4はそのフローチャートである。

[Partition を Reserve する Algorithm]

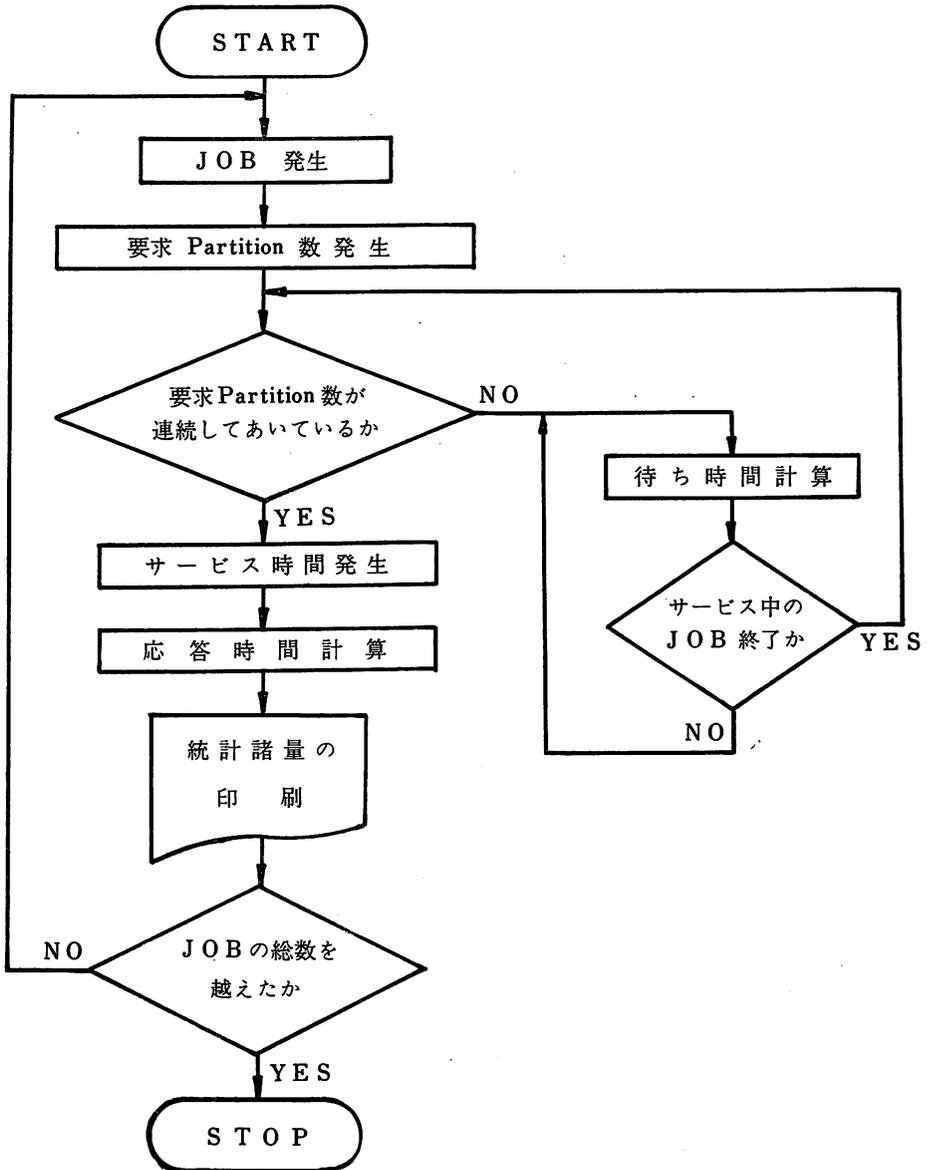


図3 モデルのゼネラル・フローチャート

各JOBのサービス時間STは要求 Partition 数PNの発生で一義的に定まるものと仮定する。

各 Partition に STATUS WORD を設け、JOBがシステムに到着したとき Available Partition には0を既に Reserve されている Partition にはサービス終了の時刻TTをセットしておく。また、Partition のサーチは Memory の上位から順次行なうものとする。

- (1) Partition Status が0ならば、Available Partition 数APを1だけ増す。
- (2) もし、その Partition が既に Reserve されているならば、 $AP=0$ とにおいて次の Partition の Status を調べる。
- (3)  $AP=PN$ になったならば連続したPN個の要求 Partition が Reserve される。

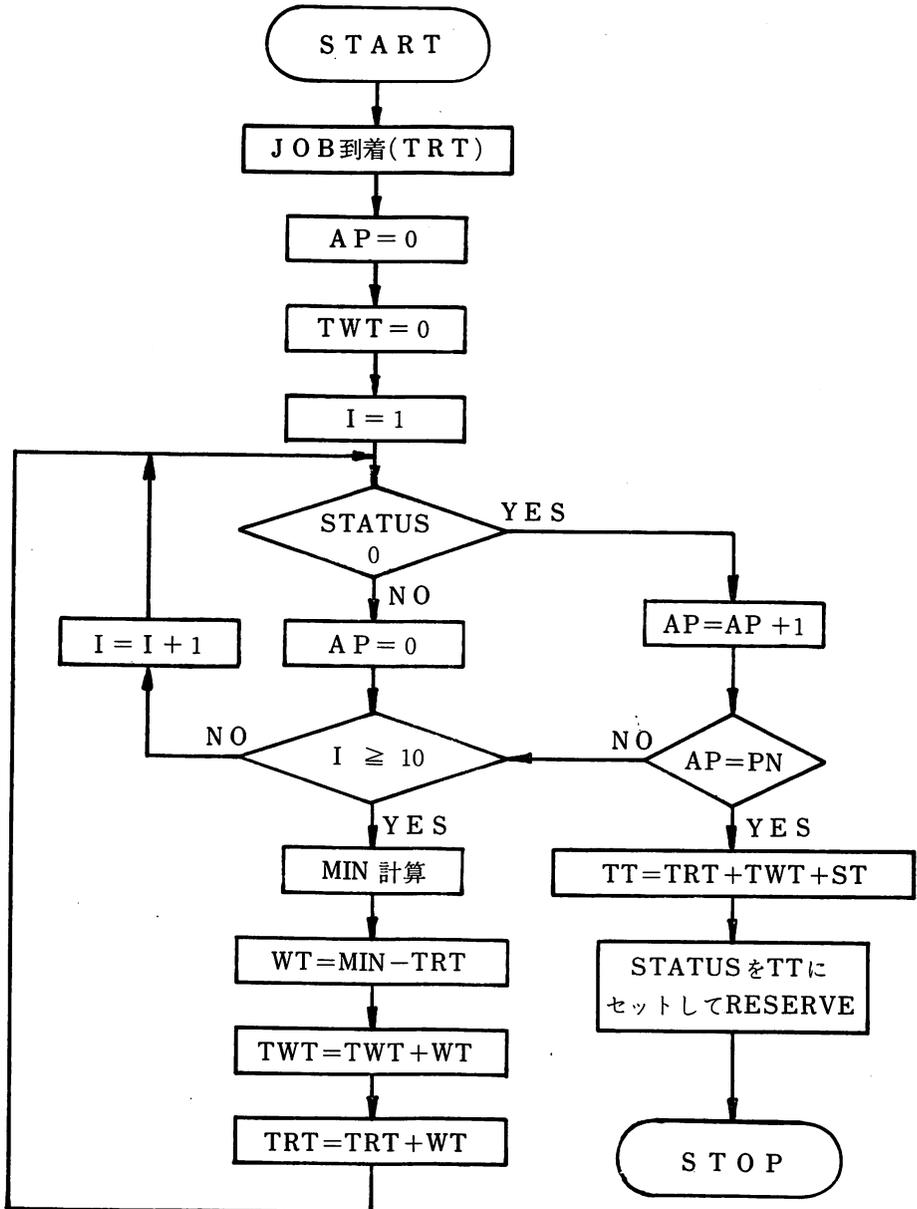


図4 Reserve Algorithm のフローチャート

(4) もし、すべての Partition を調べても AP が PN に等しくならなければ待ち時間を生ずる。

このとき、JOB が到着した時点 TRT でシステム中に存在するすべての JOB のサービス終了時刻 TT の最小値 MIN を計算して

$$WT = MIN - TT$$

とおく。

(5) JOB の到着時刻 TRT に上記待ち時間 WT を加えたものを新たに TRT とおいて、ステップ(1)にもどり、以上のステップを繰返す。

### 3. シミュレーションの実施例

上述のモデルにおいて、JOB の到着時間間隔 RT の平均値  $1/\lambda$  をパラメータとし、システムに到着する JOB について、その要求する Partition 数に対する待ち時間が得られたので図5に示す。

Response Time はこの待ち時間にサービス時間を加えることによって求められる。

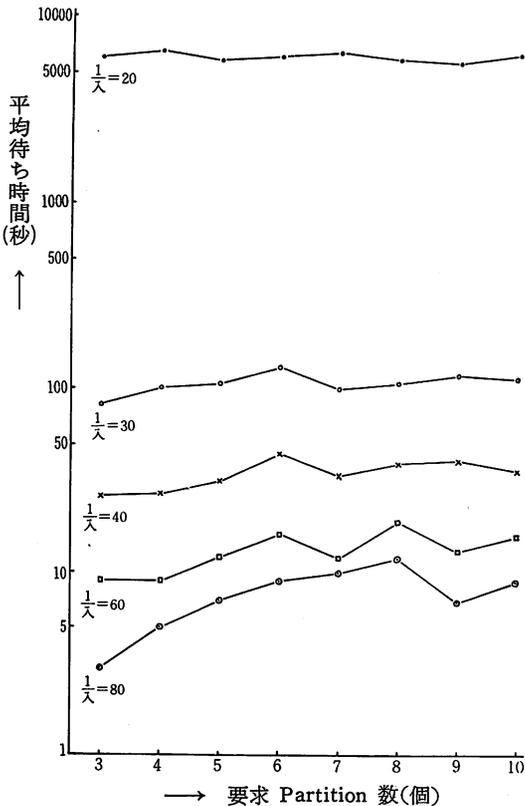


図5 各 JOB の待ち時間

### 4. 検 討

パラメータ  $1/\lambda$  は Traffic Density に対応するものであるが、これが一定ならば要求 Partition 数が大きくなっても待ち時間はそれほど変わらない。これは当初の予想に反した結果ではあるが、いわゆる First-Come-First-Service の思想から言えば納得できる。すなわち、ある JOB が到着したとき、その JOB の要求する連続した Partition があいていても、その前に大きな Partition を要求する JOB がサービスを受けられずに待っていれば、その JOB は待たされるということである。

また、当然のことながら、平均到着時間間隔  $1/\lambda$  が小さくなれば待ち時間は指数関数的に増加する。

### 5. む す び

本研究は極めて単純なモデルのシミュレーションに終わったが、連続した Partition を Reserve する Algorithm をつくる目的は一応達成された。

また、今回は Program サイズの分布は実際のデータを基準にしたが、今後は種々の分布について、さらには ROLL I/O を考慮に入れたモデルについても同様のシミュレーションを実施したいと考えている。

最後に、本研究を行なうに際し、終始熱心に御検討下さった東北大学電気通信研究所 野口正一教授、並びに大学院学生 相沢正俊、海老原義彦の両君に心から感謝いたします。また、この研究を種々お手伝いいただいた本校電気工学科 堅固山幸治技官、並びに第5学年学生 近藤淳一君に感謝いたします。

### 文 献

- 1) D.E.Knuth : Fundamental Algorithms, 1966
- 2) T.H.Naylor, J.L.Balintfy, D.S.Burdick: Computer Simulation Techniques, 1966
- 3) Hiroshi Hagiwara, Hajime Kitagawa, Tetsuzo Uehara : On the Simulation of Time Sharing System
- 4) 安井裕, 北浦隆, 福本真憲: 阪大MACシステムの利用者習性とその解析, 日本電気技報No.95, 1969
- 5) D.R.Cox, W.L.Smith : Oueues, 1961
- 6) 三浦大亮: シミュレーション入門, 1970, オーム社