

# シミュレーションによる 計算機システムの評価 (第2報)

菅原 英一・堅固山 幸治

Estimation of Computer System  
by Simulation (2nd Report)

Eiichi SUGAWARA, Koji KENGOYAMA

(昭和49年10月31日受理)

## 1. 緒 言

計算機システムにおいて最低限必要とされるハード・ウェアは入力装置、記憶装置、処理装置、出力装置であるが、これらの装置は非常に高価なものでそのコスト・パフォーマンスは大いに関心のもたれるものとなる。前報においてもこれらシステム資源の効率よい利用という立場から、シミュレーションによって種々のジョブ形態におけるシステムのコスト・パフォーマンスを求めたがそのシミュレーション・モデルに若干非現実的と思われる制約があったので本報告ではモデルを改め再度シミュレーションを実施した。

## 2. シミュレーション・モデル

### 2-1 ジョブの特性

ジョブ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、ジョブの到着の形態はポアソン到着である。また、ジョブの大きさ（メモリ占有容量）に関しては、このモデルのメモリの全容量（ユーザ領域）を128バンク、メモリ割当ての最小単位を1バンクとしているので、この範囲内に平均値をもつ正規乱数の発生によって定義する。なお、ジョブが入力装置や出力装置を占有する時間も確率密度関数である。

### 2-2 入出力装置

このモデルでは入力装置としてカード・リーダー、出力装置としてライン・プリンタを有しているが、その処理能力はそれぞれ750枚/分、1000行/分とする。

### 2-3 メモリ割当て方式

8バンクを1パーティションとするパーティショニング方式とする。すなわち、ジョブのメモリ占有容量は8バンク単位とする。

### 2-4 CPの処理能力

ジョブの大きさ（メモリの占有容量）に比例し、2バンク/秒とする。

### 2-5 待ち行列

カード・リーダー、メモリ、CP、ライン・プリンタにはそれぞれ待ち行列があり、ジョブは先着順にこれら待ち行列に並び、その先頭のジョブから順にサービスを受けるものとする。

以上要点を述べたモデルを〔図1〕に、そのゼネラル・フローチャートを〔図2〕に示すが、このモデルはフィード・バックのないストレート・キューである。

本モデルを記述する諸変数を以下に示す。

〔外生変数〕

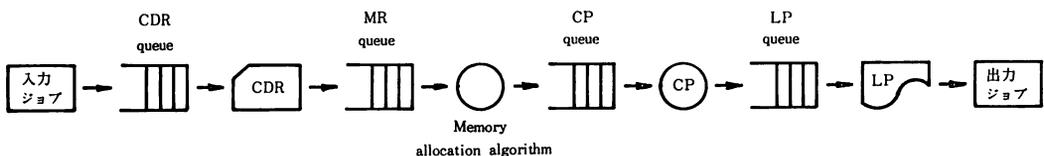


図-1 シミュレーション・モデル

$RT_i$ : 第*i*番目のジョブと第(*i*+1)番目のジョブのシステムへの到着の時間間隔

$RCD_i$ : 第*i*番目のジョブの入力カード枚数

$RCP_i$ : 第*i*番目のジョブのメモリ占有バンク数

$RLP_i$ : 第*i*番目のジョブの出力行数

[状態変数]

$TRT_i$ : 第*i*番目のジョブのシステムへの到着時刻

$CDWT_i$ : 第*i*番目のジョブがカード・リーダーのサービスを受けるまでに待った時間

$MRWT_i$ : 第*i*番目のジョブがメモリに入るまでに待った時間

$CPWT_i$ : 第*i*番目のジョブがCPのサービスを受けるまでに待った時間

$LPWT_i$ : 第*i*番目のジョブがLPのサービスを受けるまでに待った時間

$CDST_i$ : 第*i*番目のジョブに対するカード・リーダーのサービス時間

$CPST_i$ : 第*i*番目のジョブに対するCPのサービス時間

$LPST_i$ : 第*i*番目のジョブに対するライン・プリンタのサービス時間

$IDCD_i$ : カード・リーダーが第*i*番目のジョブの到着を待つ間あきになっていた時間

$IDCP_i$ : CPが第*i*番目のジョブの到着を待つ間あきになっていた時間

$IDLP_i$ : ライン・プリンタが第*i*番目のジョブの到着を待つ間あきになっていた時間

$TCDIDT_i$ : 第*i*番目のジョブがカード・リーダーのサービスを受けるまでのカード・リーダーの全あき時間

$TCPIDT_i$ : 第*i*番目のジョブがCPのサービスを受けるまでのCPの全あき時間

$TLPIDT_i$ : 第*i*番目のジョブがライン・プリンタのサービスを受けるまでのライン・プリンタの全あき時間

$MS_i$ : 第*i*番目のジョブのメモリ占有パーティション数

$CDEVNT_i$ : 第*i*番目のジョブのカード・リーダーのサービス終了時刻

$MREVNT_i$ : 第*i*番目のジョブがメモリに入った時刻

$CPEVNT_i$ : 第*i*番目のジョブのCPサービス終了時刻

$LPEVNT_i$ : 第*i*番目のジョブのライン・プリンタのサービス終了時刻

$TAT_i$ : 第*i*番目のジョブのターン・アラウンドタイム

$WCD$ : ある時点にカード・リーダーのサービスを

待っているジョブの数

$WMR$ : ある時点にメモリに入るのを待っているジョブの数

$WCP$ : ある時点にCPのサービスを待っているジョブの数

$WLP$ : ある時点にライン・プリンタのサービスを待っているジョブの数

$EVENTN$ : サービス装置の別 (1:カード・リーダー, 2:CP, 3:ライン・プリンタ)

$EVENTT$ : シミュレーションのクロック・タイム  
諸変数を上記のように定めれば, 第1番目のジョブがシステムに到着したとき, すなわち, シミュレーションの初期条件はつぎのようになる。

$RT_1=0$        $TCDIDT_1=0$

$TRT_1=0$        $TCPIDT_1=0$

$CDWT_1=0$        $TLPIDT_1=0$

$MRWT_1=0$        $EVENTN=1$

$CPWT_1=0$        $EVENTT=0$

$LPWT_1=0$

そして, その後に続くすべてのジョブについては以下に示す関係式がシステムの状態を表わす。

$TRT_i = TRT_{(i-1)} + RT_i$

$CDWT_i = CDEVNT_{(i-1)} - TRT_i + IDCD_i$

$MRWT_i = MREVNT_i - CDEVNT_i$

$CPWT_i = CPEVNT_{(i-1)} - MREVNT_i + IDCP_i$

$LPWT_i = LPEVNT_{(i-1)} - CPEVNT_i + IDLP_i$

$CDST_i = RCD_i \times 60/750$

$CPST_i = RCP \times 0.5$

$LPST_i = RLP_i \times 60/1000$

$TCDIDT_i = TCDIDT_{(i-1)} + IDCD_i$

$TCPIDT_i = TCPIDT_{(i-1)} + IDCP_i$

$TLPIDT_i = TLPIDT_{(i-1)} + IDLP_i$

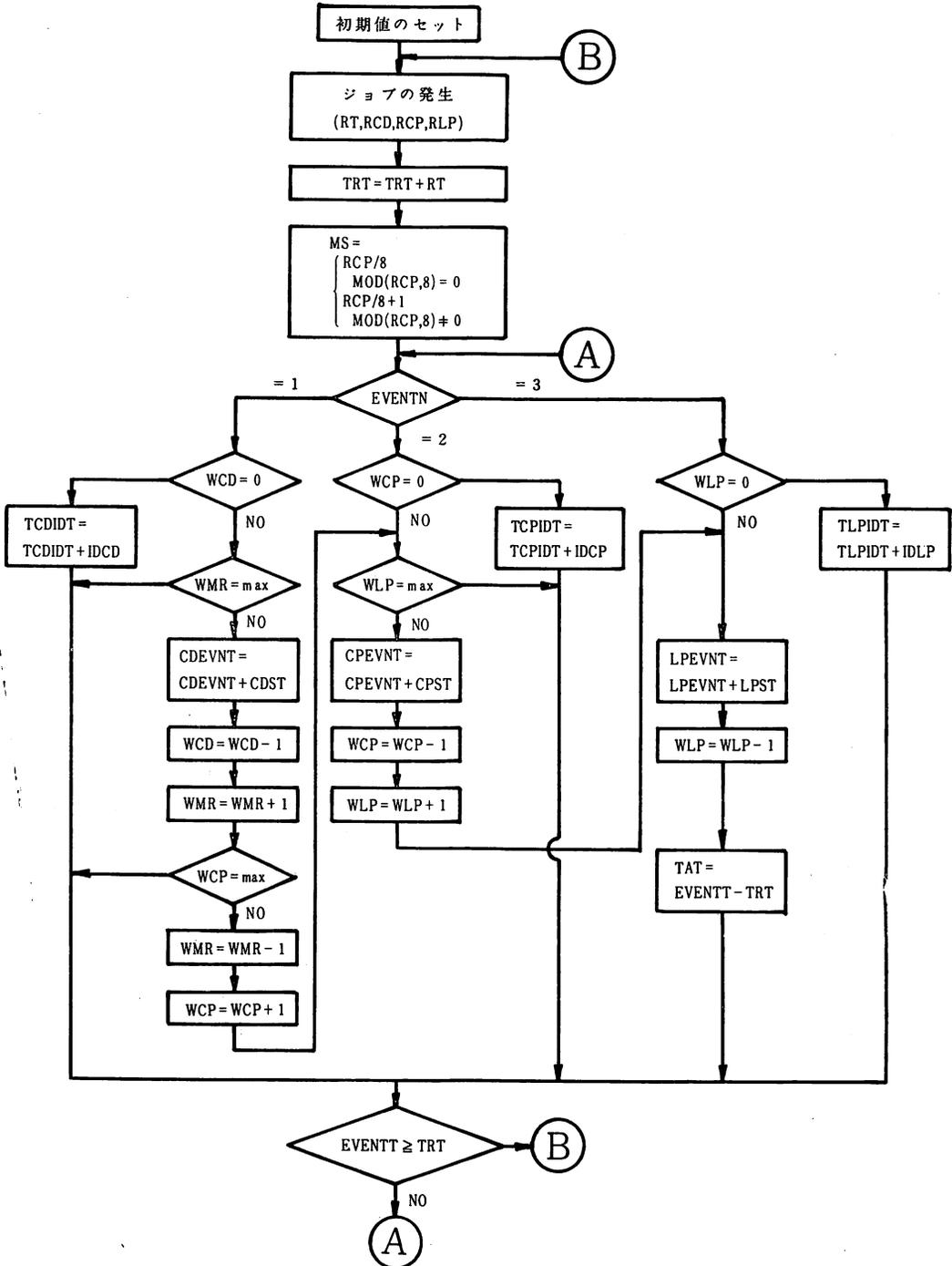
$TAT_i = TLPIDT_i - TRT_i$

$EVENTN = k(k=1 \text{ or } 2 \text{ or } 3)$

### 3. シミュレーションの実施例

上記のモデルにおいて, ジョブの到着時間間隔 ( $RT$  [秒]) の平均値:  $1/\lambda$  をパラメータとし, システムに到着するジョブのメモリ占有容量 ( $RCP$  [バンク]) に対するカード・リーダー, メモリ, CP, ライン・プリンタの各効率  $EFFCD$ ,  $EFFMR$ ,  $EFFCP$ ,  $EFFLP$  を求め, 次式によってコスト・パフォーマンス  $PERFM$  を算出した。

$$PERFM = \{(a \times EFFCD) + (b \times EFFMR) + (c \times EFFCP) + (d \times EFFLP)\} / (a + b + c + d)$$



図—2 ゼネラル・フローチャート

ただし a, b, c, d はそれぞれカード・リーダー, メモリ, CP, ライン・プリンタのコストの比率であり, 本報告の実施例では a : b : c : d = 1 : 5 : 3 : 1 と仮定した。〔図3〕にその結果を示す。

4. 検 討

一般的にジョブのシステムへの到着時間間隔  $1/\lambda$  がターン・アラウンド・タイム TAT よりも小さければ, シス

テム内に留まるジョブの数が増加し、システムの効率が上昇するかに見えるが、各装置の処理能力のバランスが悪いと、ある装置の待ち行列がどんどん長くなって、その先の装置はアイドルの状態になってしまうのでコスト・パフォーマンスが上昇するとは一概には言えない。しかし、逆に  $1/\lambda$  が TAT より大きければシステム内に留まるジョブの数は減少し、ついにはシステム内にジョブが全くなくなってすべての装置がアイドルの状態になってしまう。すなわち、 $1/\lambda$  と TAT がほぼ等しい状態がコスト・パフォーマンスを上昇させる大きな前提になることは確かである。

[図3] の例では同じ RCP (メモリ占有バンク数) では  $1/\lambda$  が小さいほど PERFM は大きくなっているが、これ

はメモリのコスト  $b$  がコスト全体の50%を占めているために  $1/\lambda$  が小さいほどメモリが満杯に近くなってメモリ効率が上昇し、したがって PERFM が上昇することを示している。

また、グラフ上の階所に極値が表われているが、これも  $b$  が全体の50%という大きな比率をもつからに他ならない。たとえば、 $RCP = 64$  で極大値をもち、その後は PERFM は低下しているが、これはメモリの全容量が128バンクであるので、 $RCP \leq 64$  ならばメモリ中に2個以上のジョブが入れるが、 $RCP > 64$  ならば1個のジョブしか入れないのでメモリ効率は下がり、したがって PERFM は大きく低下することになる。

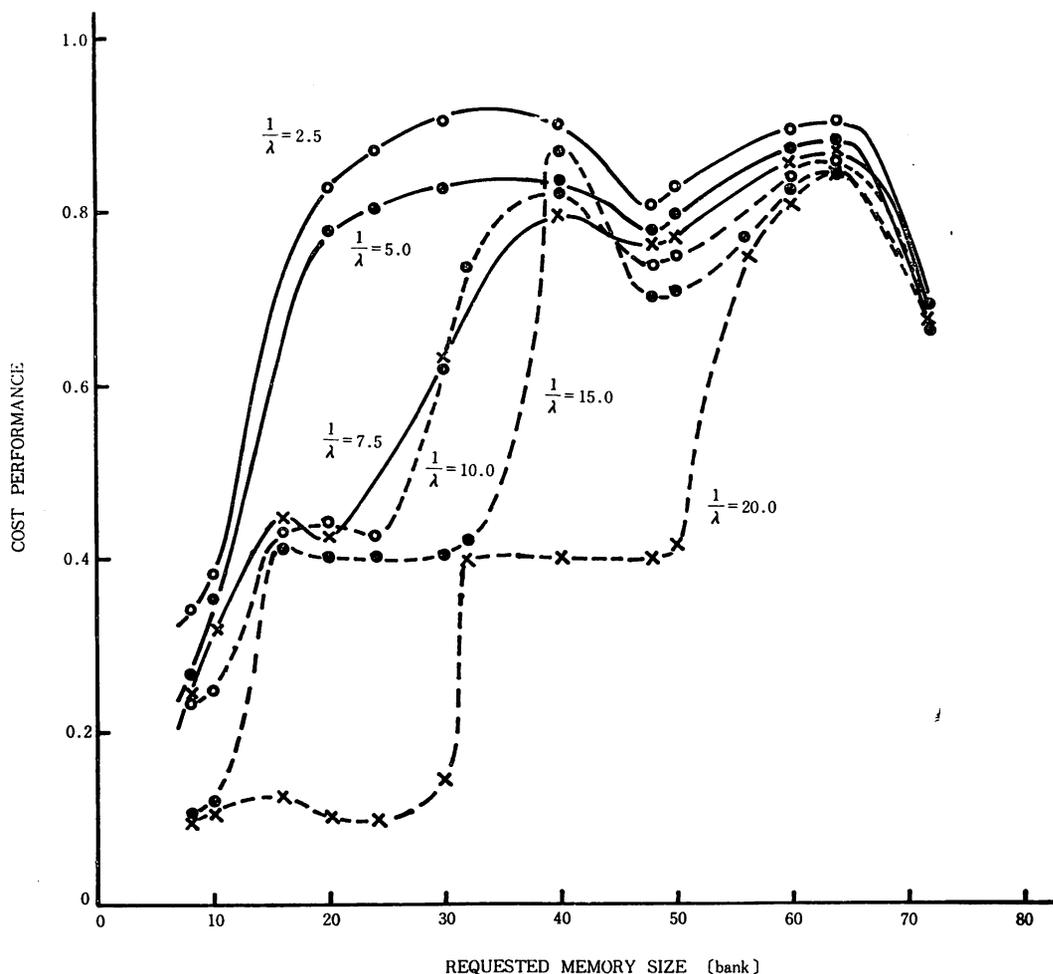


図-3 シミュレーション結果

## 5. む す び

前報のモデルの制約を大きく改良してジョブの特性がコスト・パフォーマンスに与える影響を大まかに解析できたが、このモデルにおいては補助記憶装置がないこととデータの転送時間を無視している欠点があるので、次報においてはこの点を改良してより一層現実的なモデルに接近させてシステムの効率よい利用を考察したいと思う。

## 文 献

- 1) Fundamental Algorithm, 1966
- 2) Computer Simulation Techniques, 1966
- 3) On the Simulation of Time Sharing System
- 4) 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) Queues, 1961