

Title	解説 大阪大学大型計算機センターニュース No.2
Author(s)	
Citation	大阪大学大型計算機センターニュース. 1969, 2, p. 25-38
Version Type	VoR
URL	https://hdl.handle.net/11094/65116
rights	
Note	

Osaka University Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

Osaka University

◎ 解 説

計算機システムの稼働効率の向上について

当計算センターではその運用にあたり、ユーザーのジョブはバッチ処理方式のみの時間帯と、TSSによる端末でのサービスとバッチ処理の並行処理を行う時間帯とによって処理されている。しかしすでにご承知のように、利用者のもつ計算量は計算機システムの処理能力をはるかに上まわり、常に飽和状態にある。計算センターではシステム運用の面から少しでも稼働効率の上がるよう考慮は払っているが、まだ不満足な点も多い。昭和44年1月に131K字（16K語相当，1語48ビット）のコアの増設が実現したので、ここで従来のシステムの運用方式ならびに稼働状況を報告するとともに、問題点を明らかにし今後の改善策の要点を述べる。

1) 運用方式ならびにその問題点

従来モデル 500 には393K字のコアがあり、このうち利用者のプログラムで使用できる領域は320K字（40K語相当）であった。1日の処理時間約10時間のうちで約6時間半は TSS による処理を行わず、バッチ処理のみを行うので一つのプログラムで 320 K 字全体を使用することができた。また、残りの3時間半ではセンター内のバッチ処理と5台のデータ・ステーションからの TSS 処理を並行して行うために、この場合にはセンター内バッチ処理では131K字（16K語相当）、データ・ステーションでは65K字（8 K語相当）が使用できた。

標準的なジョブについて計算機システムでの処理の流れを追ってみると、カード・リーダーから読み込まれたプログラムは一度磁気テープにスタックされる。したがって、実際にそのプログラムがコンパイルされて実行されるときには、磁気テープから読み込まれる。また、出力データについても、プログラム実行と同時に結果は一度磁気テープ上に書き出されて、ライン・プリンターへの打ち出しは磁気テープから行なわれる。この入出力機器と磁気テープ間のデータの移送は、DT プログラム（Data Transcription Program）と呼ばれるものによって行なわれ、これはいつでも利用者ジョブの処理と並行して、別のコア領域20K字を使って働かせることができる。

次に計算機システムの稼働効率の面からみると、入出力機器の動作時間は利用者ジョブの処理時間と重複していてさしつかえないから、直接問題にはならない。しかし利用者ジョブの処理中に行なわれる主記憶と磁気テープとのデータ転送時間は、その間、そのジョブについては処理が中断され、CPU（中央処理装置）の動作が停止することになる。この遊び時間を少なくするには、主記憶と直接データのやりとりが行なわれる磁気テープ、ディスクなどの補助記憶装置を高速のものにするか、複数個のジョブを主記憶の中におき、各ジョブについて独立に動作する補助記憶装置を用意することにより、常にどちらかのプログラムを処理するようにして、互いの処理中断時間を吸収させることの2点が挙げられる。

当センターのシステムでは補助記憶として6個のファイルが必要とする。現在のオペレーティング・システムは磁気テープをベースとするもので、6個のファイルのうちシステム・オペレーティング・ファイルのみをディスク・バックに割り当て、他の5個のファイルは5台の磁気テープを使用して

いる。これらのファイルをディスク・バックに割り当てることも考えられているが、記憶容量の問題、ならびに割り当てられた各装置が独立に動作しなければ効率が悪いことなどの理由により、すべてをディスク・バックに割り当てることには問題がある。

複数個のジョブの並行処理については、オペレーティング・システムの機能としては、二つのジョブの並行処理ができるようになっている。ただし、この場合には二つのジョブが主記憶の中に同時におかれることになるため、利用者プログラムで使用できるメモリ・サイズが小さく制限される。そのため従来は、利用者のプログラムに大きなコア領域を開放して、もう一つのプログラムとしては、小さな領域で処理できる DT プログラムだけを働かせていた。

以上の点を考慮した上で、運用上ジョブを受け付けるときに、必要コア領域が 131K 字未満のものとそれ以上のものに分けて受け付け、131 K 字未満のジョブについては TSS の処理と並行して処理して来た。また、131 K 字未満のジョブでは比較的短時間のものが多く、後でも述べるように、短時間のジョブでは CPU の動作時間に比べて入出力関係の時間が大きなウェイトを占めるため、CPU の遊び時間が大きいわけであるが、この時間を TSS 処理の方で吸収させるようにはかかってきた。

2) 稼動状況の調査結果

稼動状況をみるため、昭和43年5月～7月にわたって調査した結果の表および図を後ろにまとめて挙げる。(なお、ここに集計されているデータは5月～7月のすべてのデータではない。半日処理も含めて66日分。不良データの検出に伴い、統計的性質を失なわないようその一部は除外されている。)

集計は CPU time と USE time で別に行なっているが、CPU time とはそのジョブ(または各処理段階)のために CPU が動作した時間であり、USE time はそのジョブ(または各処理段階)の処理開始時点と処理終了時点との時間差である。したがって USE time から CPU time を差し引いた残りの時間は、そのジョブの処理中に他のプログラムが CPU を使用した時間、ならびに入出力動作のための CPU の待ち時間が含まれる。したがって、利用者のジョブが一つずつ処理されている場合には、CPU time と USE time を比較することにより稼動効率の目安が得られる。

これらの表および図について、第3表以外のものはセンター内バッチ処理によるものみの集計で、データ・ステーションから使用した CPU time は含まれていない。なお、集計は MAC サービス中の時間帯と、MAC サービスなしのバッチ処理だけの時間帯に分けて行っているが、特に MAC サービス中の集計においては(たとえば第1表) USE time と CPU time との差の時間の中にデータ・ステーションかから使用した CPU time が含まれていることに注意する必要がある。

第2表はバッチ処理だけを行った時間帯の集計で、この時間帯では利用者のジョブが一つずつ処理されているのでシステムの稼動効率をみることができる。ここで USE time と CPU time の時間差は、主記憶と磁気テープとのデータ転送の間 CPU が遊んでいる時間と DT プログラムならびにモニターの CPU 使用時間の和であり、前者の遊び時間については運用方法の改善により、短縮できるはずである。一般には、CPU time の短いジョブの処理ほど効率が悪く、コンパイル時には USE time の中で CPU time の占める割合は 30～40% しかない。しかしこの時間帯では、ジョブ全体として効率のよい長時間のものが多く処理されるので、すべてのジョブの処理時間の合計では効率 (CPU

time と USE time の比) は約76%になっている。残りの24%については、モニターで使用する CPU time を考慮してもまだ改善の余地が残されていると考えられる。処理時間の長さとして述べた意味での効率の関係は、第8図の×印で結んだ折線、ならびに他のグラフの USE time と CPU time の差の部分などにみることができる。

他方、MAC サービス中の時間帯については、バッチ処理では短時間の能率の悪いジョブが処理されているわけであるが、同時にデータ・ステーションからの処理も並行して行なわれているので、データ・ステーションで使用している CPU time も考慮しなければシステム全体の効率は明らかにならない。

第3表は昭和43年7月について、データ・ステーションからの処理時間も含めて集計したものである。この表でみると MAC サービス時の効率は約54%であり能率はよくない。これには二つの原因が考えられる。その一つとしてはバッチ処理と端末からの処理の両方を並行して行っているにもかかわらず、どちらのジョブもデータ転送を行なう頻度が多く、入出力動作実行のために両方のジョブの CPU による処理がともに中断される時間帯が、かなり残っているのではないかということ。もう一つは端末からの処理については、利用者プログラムでを使用した CPU time の他に、データ・ステーションならびにファイル・システムとのデータ転送などの MAC サービス特有のモニターの動作時間が、この集計の中に現われていないことである。

なお、データ・ステーション利用の効果について、MAC サービス時の時間帯でバッチ処理だけを行ったと仮定した時の推定件数よりも23%多くの件数が処理されている。これは短時間のデバッグを目的とするプログラム、あるいはテスト・ランのプログラムの実行が、短いターン・アラウンド・タイムにより、多くのユーザーに開放されていることを意味し、この点の効果は上がっていると思われる。(注意)

後ろに添えた図において、横軸の CPU time の区分が、均一でないことに注意する必要がある。横軸が JOB CPU time, または OBJECT PROGRAM CPU time の場合、区分が20分までは1分きざみであるが、20分以上では10分きざみになっている。したがって第1図、第2図、第5図においては、グラフの形としては20分以上の区分のヒストグラムの高さは、図に表わされているものの10分の1と考えなければならない。

FÖRTRAN CPU time, LINKLOAD CPU time では2分までが10秒きざみ、2分以上は1分きざみになっているので、第3図、第4図で分布の形を見るには2分以上の区分のヒストグラムの高さは6分の1になるものとする必要がある。

3) 今後の改善策

昭和44年1月より131K字のコアの増設が実現し、また先に述べた処理に要するファイルの中で、ライブラリー・ファイルがディスク・バックに割り当てられることになった。これにより二つのジョブの並行処理が可能となり(1月16日より実施)、第1節で述べた問題点について改良が行われることになった。

しかし、中央処理装置は依然として1台であって、処理能力の向上にも限度があり、さらに 2 job

stream による運用方式が実際上どれだけ有効であるかということは、具体的なデータに基づき調査をしてみなければわからない。今後 2job streamによる稼動状況の調査を行い、運用方式の検討を行う予定である。

第1表 MACサービス時に処理されたもの（昭和43年5月～7月）

処理段階別	件数	CPU time 合計	USE time 合計	ΣCPU/ΣUSE
FÖRTRAN	2523 (100%)	10時間8分 (11.3%)	43時間13分 (20.5%)	0.234
LLNKLÖAD	2139 (84.8%)	4時間34分 (5.1%)	22時間47分 (10.8%)	0.201
ÖBJECT PRÖGRAM	1926 (76.3%)	74時間15分 (82.9%)	125時間13分 (59.5%)	0.593
JÖB	2523 (100%)	89時間30分 (100%)	210時間25分 (100%)	0.425

第2表 MACサービスなしの時間帯に処理されたもの（昭和43年5月～7月）

処理段階別	件数	CPU time 合計	USE time 合計	ΣCPU/ΣUSE
FÖRTRAN	1973 (100%)	12時間25分 (5.0%)	33時間23分 (10.3%)	0.372
LINKLÖAD	1624 (82.3%)	5時間22分 (2.2%)	17時間53分 (5.5%)	0.300
ÖBJECT PRÖGRAM	1601 (81.1%)	228時間29分 (92.6%)	260時間10分 (79.9%)	0.878
JÖB	1973 (100%)	246時間47分 (100%)	325時間36分 (100%)	0.758

第3表 昭和43年7月に於ける CPU time, USE time の集計

(端末から処理されたジョブを含む)

時間帯区分	処理状況別	件数	CPU time 合計	USE time 合計	ΣCPU/ΣUSE
MACサービス時	端末からのクローズド処理	369件 (14.7%)	6時間36分 (4.2%)	85時間41分 (36.5%)	0.542
	端末からのオープン処理	221件 (1) (8.8%)	3時間57分 (2.5%)		
	センター内処理	1067件 (42.6%)	35時間55分 (22.6%)		
MACサービスなしの時間帯	センター内処理	849件 (33.9%)	112時間23分 (70.7%)	148時間50分 (63.5%)	0.755
計		2506件 (100%)	158時間51分 (100%)	234時間31分 (100%)	0.677

注 (1) 端末オープン処理の件数は、CPU time の比率によりクローズド処理の件数から推定したものの。

注 (2) () 内の数字 (%) は処理状況別ごとの比率を示す。

第4表 MACサービス時に処理されたジョブ（第3表）を、1 job stream の
バッチ処理で処理したと仮定した時の推定 USE time

時間帯区分	処理状況別	件数	処理状況別 推定 USE time	MAC サービス時全ジョブ 推定 USE time
MAC サービス時	端末からのクロ ーズド処理	369件	15時間32分 ±49分	92時間52分 ± 2時間 3分
	端末からのオー プン処理	221件	9時間17分 ±38分	
	センター内処理	1067件	68時間 3分 ± 1時間46分	

(注) ±で示した区間は95%信頼区間である。

備考 USE time の推定方法

5月～7月に MAC サービスなしの時間帯で処理されたジョブについて、JOB CPU time区分ごとに USE time と CPU time の差の平均と分散を求める。これを JOB CPU time 区分ごとに第3表 MAC サービス時のジョブの度数分布表に適用して推定。

* 第4表の推定 USE time と第3表の実質 USE time の比

MAC サービス時相当の時間帯に於ける比率：

$$\frac{\text{推定 USE time}}{\text{実質 USE time}} = \frac{92\text{時間}52\text{分}}{85\text{時間}41\text{分}} = 1.08$$

全処理時間帯に於ける比率：

$$\frac{148\text{時間}50\text{分} + 92\text{時間}52\text{分}}{148\text{時間}50\text{分} + 85\text{時間}41\text{分}} = 1.03$$

* 端末からの並行処理による処理件数増加分の推定

第3表の MAC サービス時の全 USE time 中に、1 job stream バッチ処理だけを行ったと仮定する。この時のジョブの性質が実際にバックグラウンド処理で処理されたものと同様であってただ処理時間だけが長くなったものとする、この場合の推定処理件数は

$$1067\text{件} \times \frac{85\text{時間}41\text{分}}{68\text{時間}3\text{分}} \approx 1343\text{件}$$

実際に端末との並行処理で処理された件数は第3表から

1657件

実際に端末を用いて処理した時の件数と、バッチ処理だけを行った時の推定件数の比は

MAC サービス時相当の時間帯に於ける比率：

$$\frac{1657\text{件}}{1343\text{件}} = 1.23$$

全処理時間帯に於ける比率：

$$\frac{849\text{件} + 1657\text{件}}{849\text{件} + 1343\text{件}} = 1.14$$

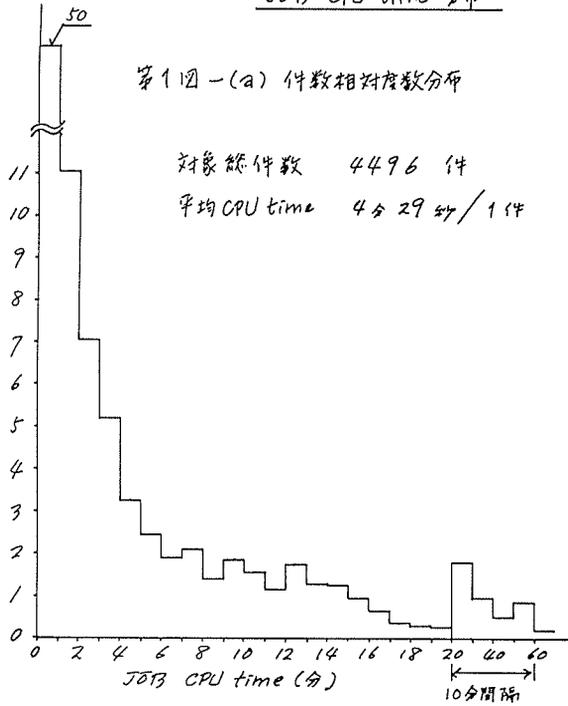
センター内バッチ処理によるジョブの

JOB CPU time 分布

件数
(相対度数
%)

第1図-(a) 件数相対度数分布

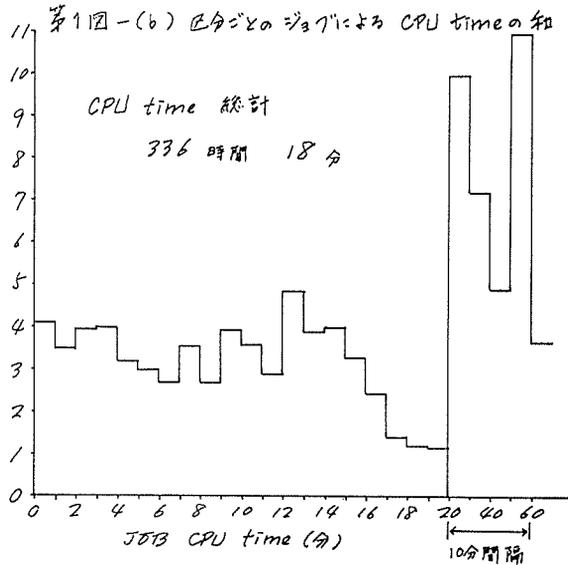
対象総件数 4496 件
平均CPU time 4分29秒/1件



処理時間
部分比率
%

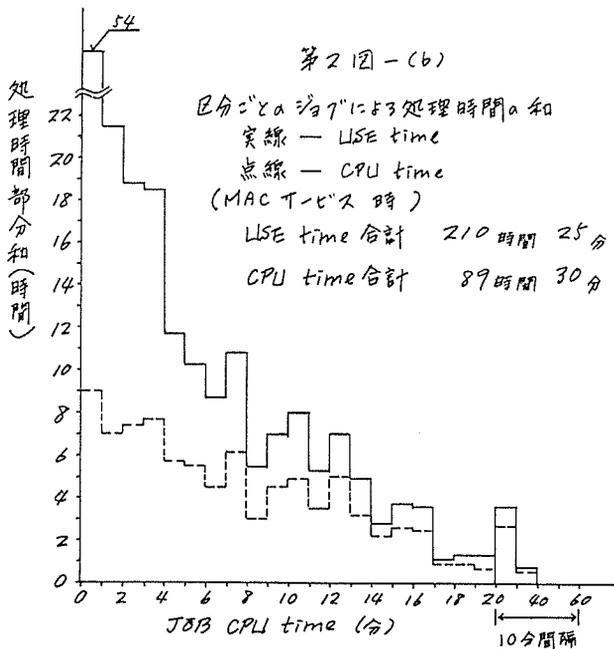
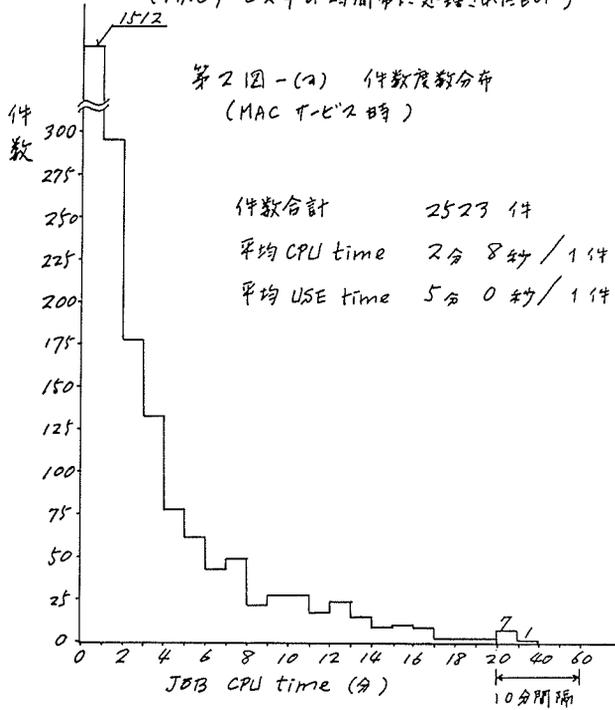
第1図-(b) 区別のジョブによる CPU time の和

CPU time 統計
336 時間 18 分

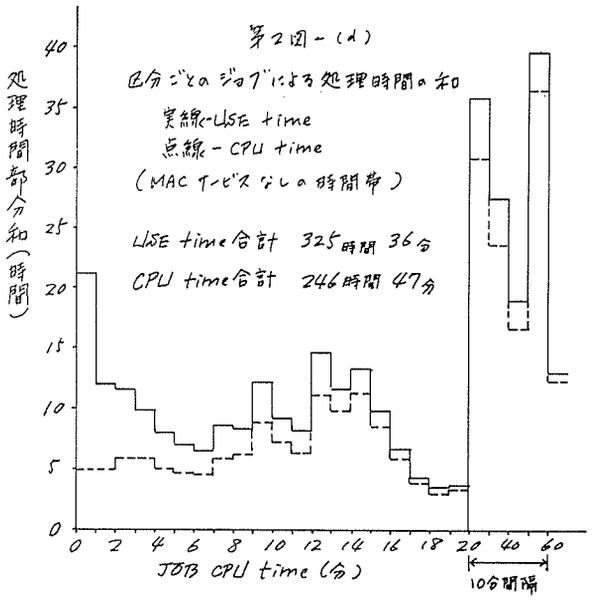
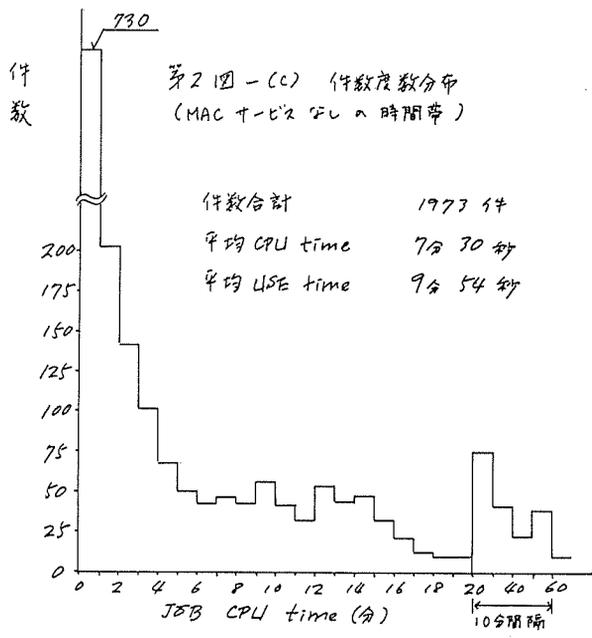


JOB time — CPU time 区分による分布

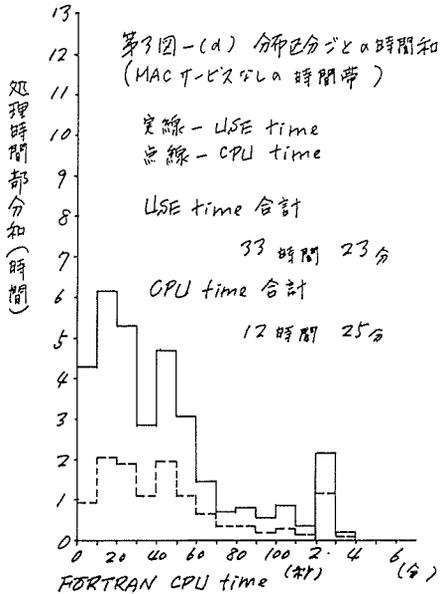
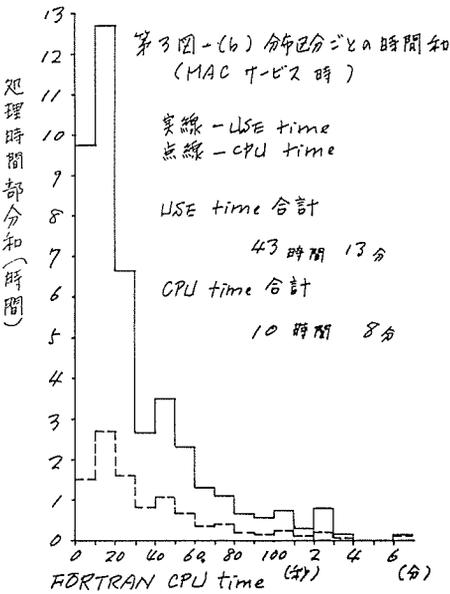
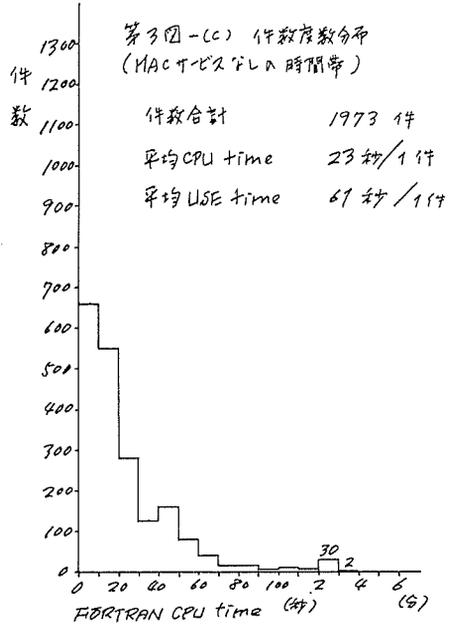
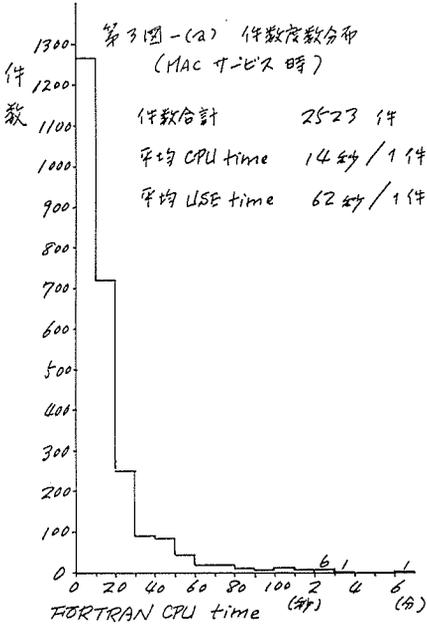
(MAC T-ビス中の時間帯に処理したものを)



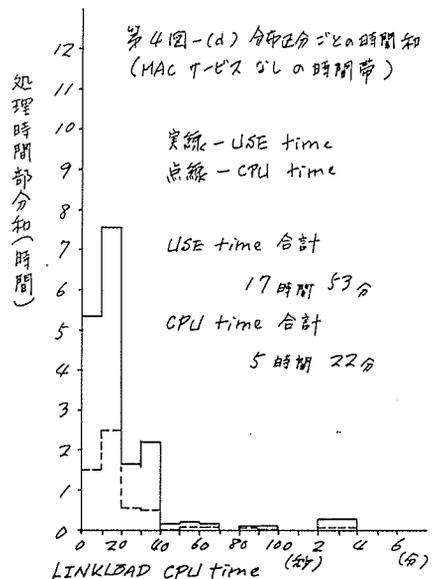
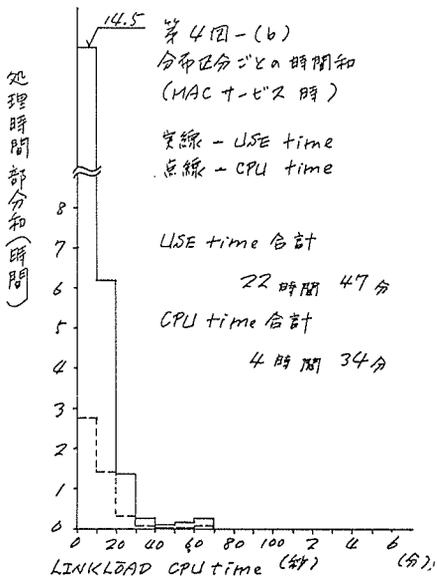
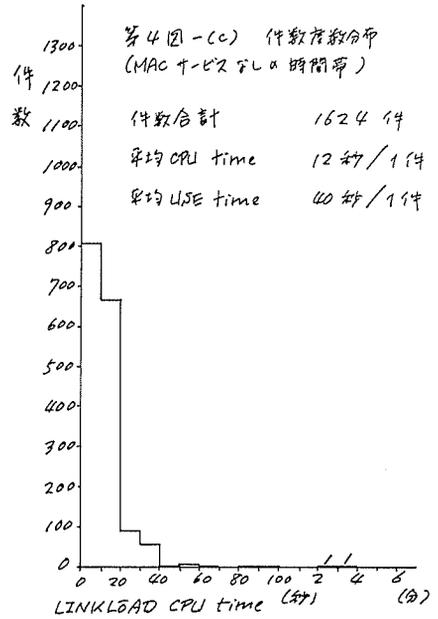
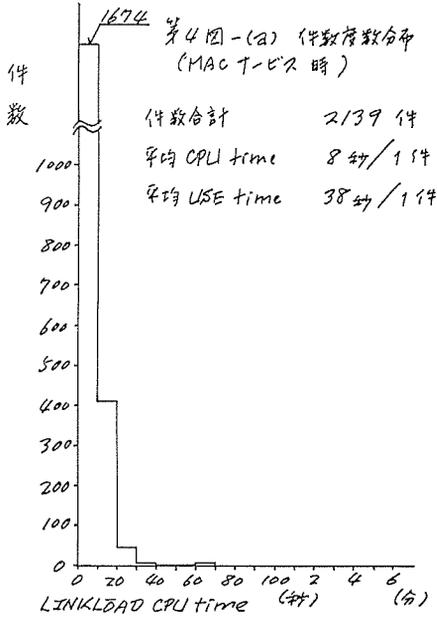
JOB time — CPU time 区分による分布
 (MAC T-ビス 5% の時間帯に処理されたもの)



FORTRAN time — CPU time 区による分布

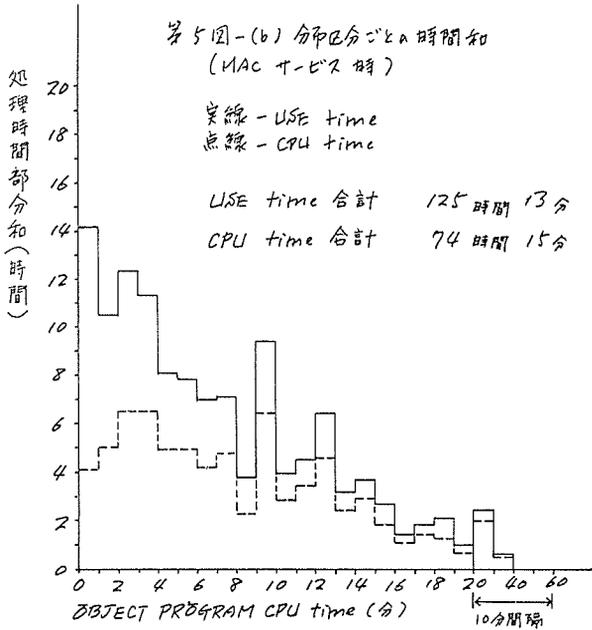
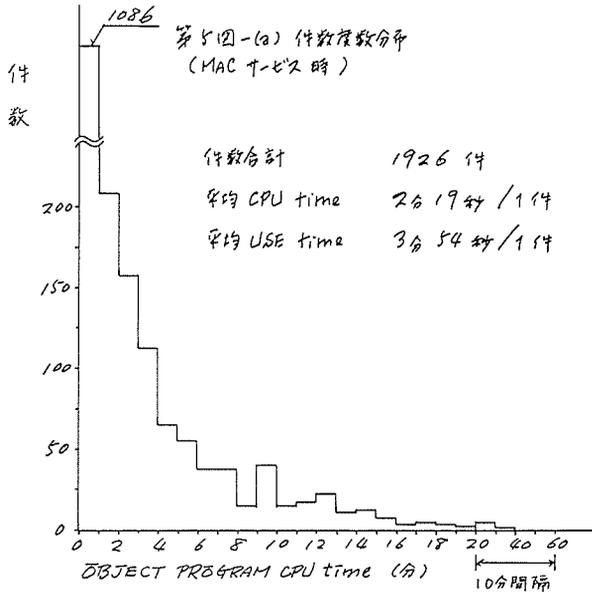


LINKLOAD time CPU time 区分け分布



OBJECT PROGRAM time 分布一

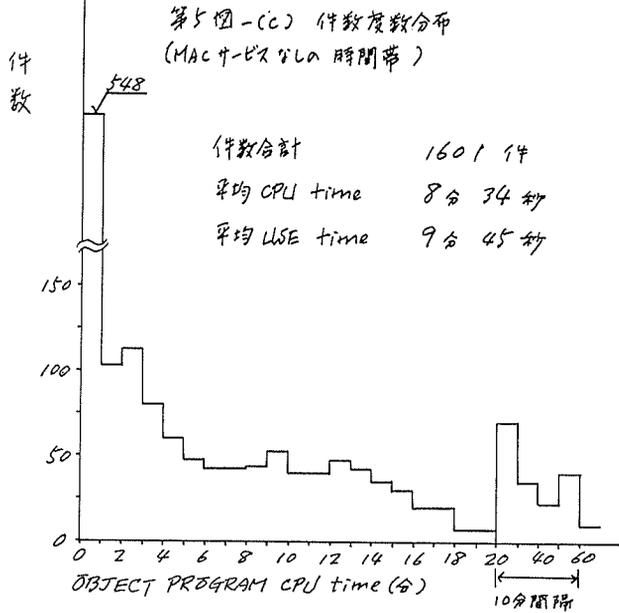
CPU time 区分による分布
(MAC サービス時)



OBJECT PROGRAM time 分布 -

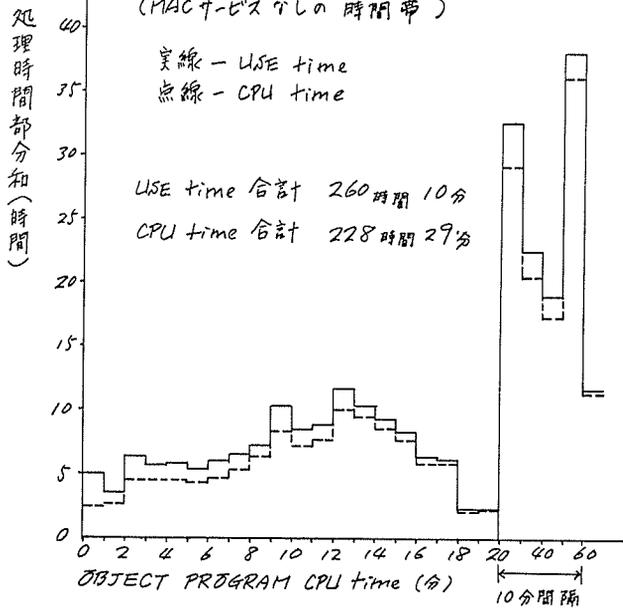
CPU time 区分による分布

(MAC サービス 終了の 時間帯)



第5図-(d) 分布区分ごとの時間総和

(MAC サービス 終了の 時間帯)



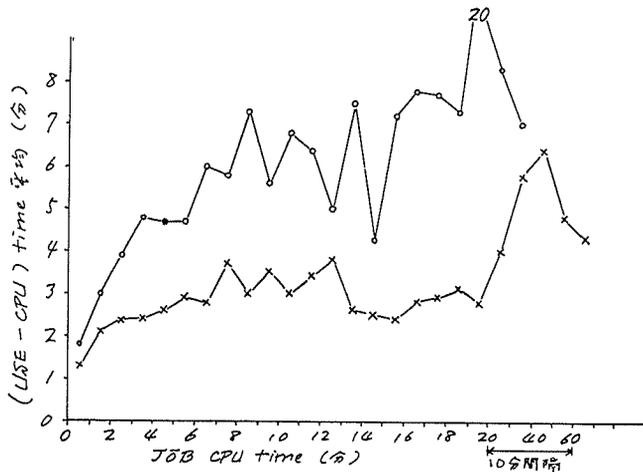
第6図

CPU time で類別されたジョブの
時間差

(USE-CPU) time

1件当りの平均

- MACサービス時
- ×— MACサービスなし

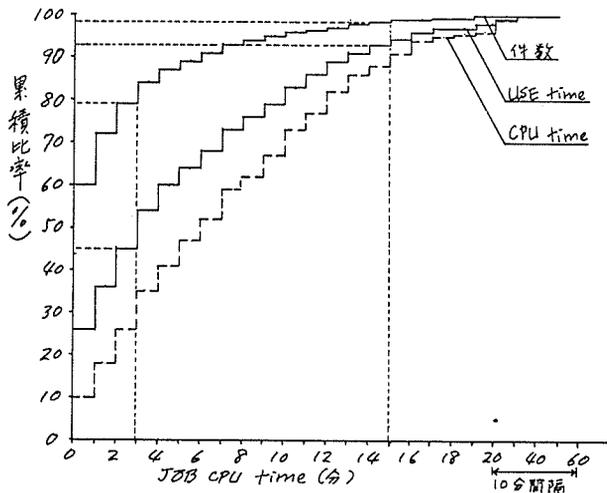


第7図—(a)

CPU time で類別されたジョブの
累積比率

(MACサービス時)

- 件数—実線
- USE time—実線
- CPU time—破線

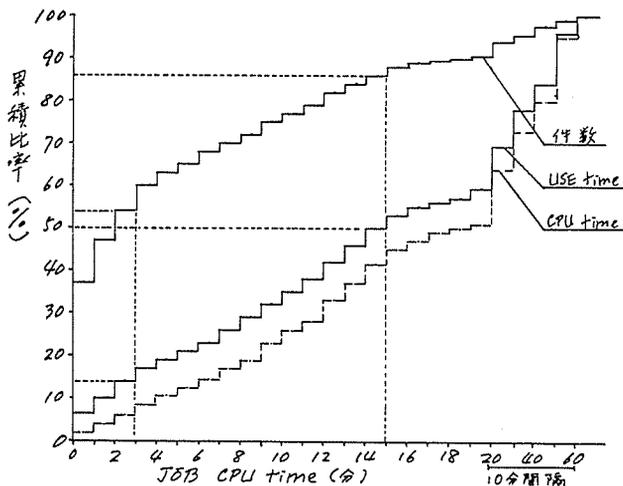


第7図—(b)

CPU time で類別されたジョブの
累積比率

(MACサービスなし)

- 件数—実線
- USE time—実線
- CPU time—破線



段階別処理時間の比率: $\frac{CPU\ time}{USE\ time}$

(CPU time により類別した処理の平均比率)

○— MAC T-ビス時 ×— MAC T-ビスの時間率

