

**La risorsa memoria**

**in ambiente batch**

**con carico variabile**

R. Bartoli - O. Signore - E. Stefanelli

58

**GNUCE**

A cura di: R. Bartoli  
O. Signore  
E. Stefanelli

Copyright 1 Giugno 1974  
by CNUCE - Pisa  
Istituto del Consiglio Nazionale delle Ricerche

CONSIGLIO NAZIONALE DELLE RICERCHE  
CNUCE

La risorsa memoria  
in ambiente batch  
con carico variabile.

R. Bartoli

O. Signore

E. Stefanelli

## INTRODUZIONE

Quando un sistema di calcolo deve effettuare un servizio per utenti molto diversi e .... "imprevedibili", nel senso che la loro produzione ricorrente e' molto meno pronunciata del lavoro .... "straordinario", molto difficili diventano la definizione delle risorse necessarie ad adempiere questo compito e le scelte di gestione per incontrare le esigenze dell'utente, anche quando il carico non e' mediamente pesante.

In particolare in ambiente "multiprogramming batch", per la necessita' di seguire il piu' possibile l'andamento fluttuante del carico, riveste speciale importanza la risorsa memoria.

Lo Hardware e il Software standard concedono al proposito due possibili scelte: o utilizzare un sistema a memoria reale di caratteristiche sufficienti, o usufruire delle tecniche di paginazione per la memoria virtuale. Il primo sistema, a causa dell'impossibilita' di definire tali caratteristiche per i massimi di carico, impone al gestore delle precise scelte operative, fortemente dipendenti dall'ambiente sollecitante, per cui e' difficile prevedere buoni risultati. Il secondo, anche se e' ottenuto sottraendo parte delle risorse, elimina quasi totalmente l'incidenza sul sistema sia dell'utente sia di quella parte di gestione necessaria per la definizione dei parametri di carico.

Premesso che un'analisi comparativa teorica tra queste scelte difficilmente potrebbe tradursi in considerazioni efficaci, un'indagine pratica risulta tanto piu' significativa quanto piu' sono simili le condizioni di carico. L'avvenuto scambio al CNUCE tra un sistema IBM 370/155 con OS-MVT/HASP e un sistema IBM 370/158 con OS-VS2(Rel. 1.6)/HASP, quindi nelle stesse condizioni di sollecitazione esterna, ha suggerito di effettuare misure per valutare le due situazioni.

In tale ambiente il problema di misurare l'utilizzo della memoria in relazione alla sollecitazione del carico, sarebbe stato, anche se parzialmente, di complicata ma di possibile soluzione. Infatti, tramite i dati ricavati dalle SMF e i valori conosciuti da HASP, sarebbe stato possibile effettuare ad ogni unita' di tempo una "fotografia" delle code e della memoria. Cio' pero' sarebbe stato "biased" da importanti fattori non quantizzabili, quali l'intenzione dell'utente, che conosce le scelte di gestione ed in generale il sistema, le decisioni dell'operatore troppo coinvolto nel metodo, gli spazi inutilizzabili a causa dell'allocazione tipo OS... etc. Il risultato di una indagine cosi' concepita, gia' in partenza difficile poi da inquadrare nel giusto contesto, non avrebbe portato a conclusioni tanto

valide quanto sarebbe stato giustificato dall'alto numero di procedure necessarie per definire, il plu' a fondo possibile, l'ambiente.

Per questo si e' pensato di utilizzare un numero abbastanza limitato di valori sperimentali, che, con una opportuna elaborazione e con considerazioni non sempre quantizzabili, ma senza dubbio efficaci, fossero in grado di giungere a risultati significativi, in assoluto e in confronto.

La nostra supposizione si basa sul fatto che non e' indicativo, in ambiente reale, pensare che il carico sollecitante sia indipendente dal sistema di utilizzo della memoria e dalla memoria stessa, ma che questa funzionalita' sia sempre a due vie. Non e' corretto, quindi, misurare il carico come forza sollecitante, e la memoria come risorsa sollecitata, e pensare che questa funzione possa rappresentare il livello di bonta' del campo d'indagine; mentre, conoscendo, anche in modo non numerico, il tipo di ambiente potenziale sollecitante, e rilevando effetti di produzione, si possono, anche se non sempre in assoluto sicuramente in paragone, ottenere risultati efficaci.

Mentre si rimanda all'appendice per una breve illustrazione delle scelte di gestione influenzanti la memoria, si anticipa che questo paragone e' risultato nettamente favorevole al /158. Cio' non solo per le sue maggiori "performances" hardware, ma anche per la diversa filosofia dei sistemi software, che riducono, in tal caso, il numero di interazioni tra i sistemi e l'uomo, esplicito richiedente della memoria quando utente, e forzato punto di legame tra l'utente e il software quando gestore.

Misure, definizione dei parametri significativi e risultati.

I valori di produzione utilizzati per una indagine di questo tipo sono stati rilevati dai records di tipo 4 (Step Termination) e di tipo 5 (Job Termination) delle SMF.

In particolare sono stati esaminati questi records relativi a circa 3 mesi di normale produzione del 370/155 e a circa 3 mesi del 370/158 che hanno dato rispettivamente 73 e 62 giorni lavorativi normali. Elaborando opportunamente questi records si sono individuate e definite per lo scopo prefisso le seguenti grandezze:

- numero di jobs e di steps prodotti
- numero di jobs versus tempo di CPU
- numero di steps versus memoria allocata
- numero di steps versus memoria utilizzata
- memoria allocata media AT/Ut=A1
- memoria allocata media per step (job) AT/St=A2
- memoria usata media UT/Ut=U1
- memoria usata media per step (job) UT/St=U2
- massima memoria usata MT/Ut=M1
- massima memoria usata per step (job) MT/St=M2
- fattore di multiprogrammazione  
 $A1/A2 = U1/U2 = M1/M2 = F1$
- fattore "ideale" di multiprogrammazione  
 $U1/A2 = F2$
- tabella oraria di allocazione della memoria nel giorno  
(A1 relativo all'intervallo di un'ora)
- tabella della frequenza di occupazione della memoria, nei giorni misurati, da parte di almeno un job
- tabella dell'andamento della frequenza di cui sopra nelle ore del giorno, ottenuta come media pesata
- fattore di utilizzo relativo della CPU  
 $C1 = T/Ut$

- fattore di utilizzo relativo della CPU per step (job)  
 $C2 = T/St$

dove, se S = somma su tutti gli steps prodotti,:

AT = S ((memoria allocata allo step) x (tempo di permanenza in memoria dello step))

UT = S ((memoria usata dallo step) x (tempo di permanenza in memoria dello step))

MT = S ((memoria massima usata dagli steps di uno stesso job) x (tempo di permanenza in memoria dello step))

Ut = Unione dei tempi di permanenza in memoria degli steps (jobs); in tale periodo esiste in memoria centrale almeno un job

St = S (tempo di permanenza in memoria dello step)

T = S (tempo di CPU utilizzato dallo step)

(Quando si parla di memoria, si deve intendere memoria reale sul 370/155 e memoria virtuale sul 370/158)

Si sono ottenuti per il /155 13075 jobs (28002 steps) e per il /158 13958 jobs (32009 steps) che corrispondono rispettivamente a circa 380 steps/giorno e 510 steps/giorno. Le grandezze T, Ut, St sono state rispettivamente

|      |               |               |               |
|------|---------------|---------------|---------------|
| /155 | 1.188.170 sec | 2.336.167 sec | 3.081.355 sec |
| /158 | 848.514 sec   | 1.823.000 sec | 4.431.259 sec |

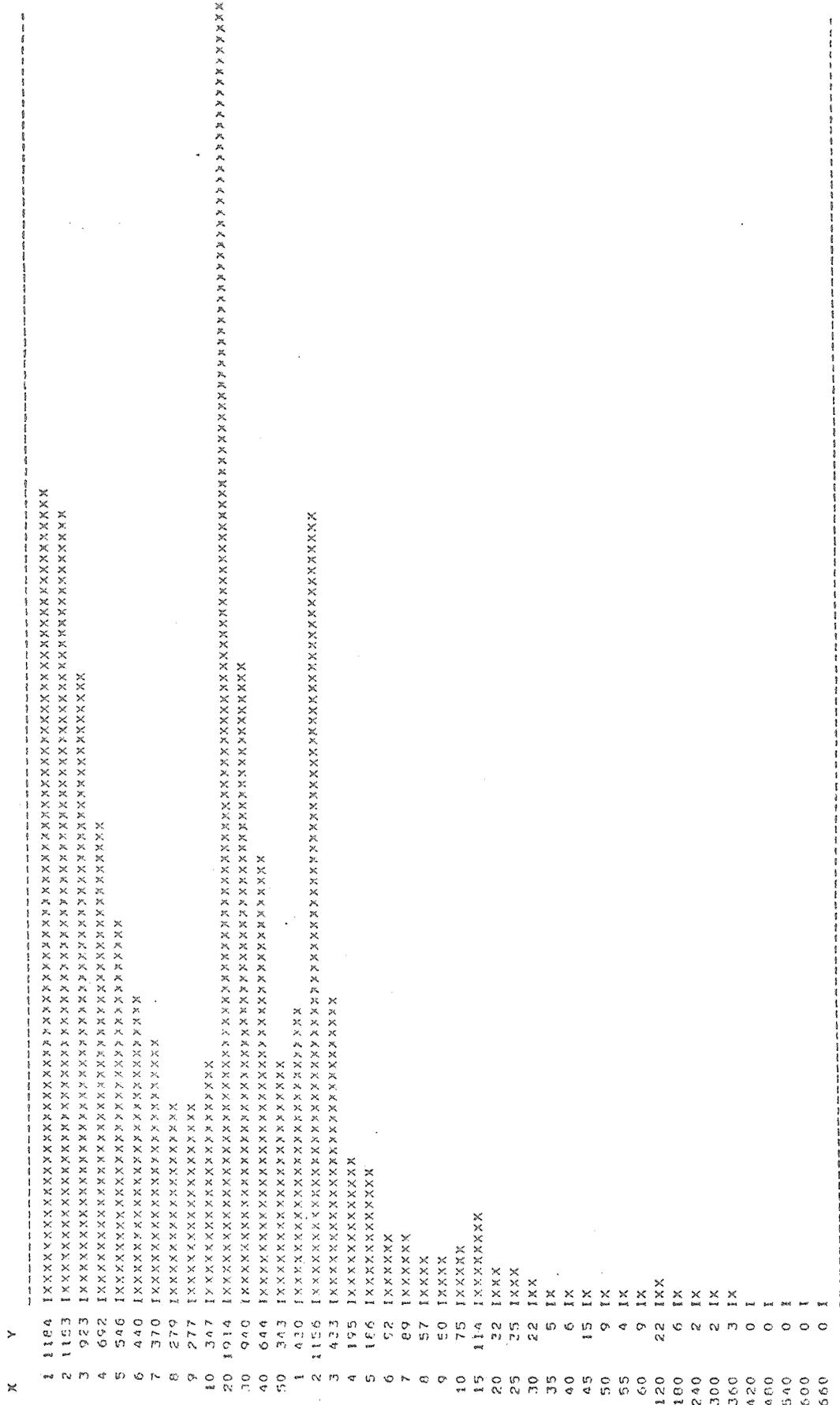
e cio' significa che rispettivamente lo step medio utilizza la CPU 50 e 26 secondi, i valori C2 sono 0.38 e 0.19, i valori C1 sono 0.51 e 0.45, il rapporto di attivita' medio tra le due macchine, (tempo in cui esiste in memoria almeno un job) e' circa 1.08 a favore del /155.

Gli istogrammi che mostrano il numero di jobs versus il tempo di CPU sono illustrati nelle figure 1,2.

Le grandezze AT, MT, UT sono state rispettivamente

|      |                   |                  |                  |
|------|-------------------|------------------|------------------|
| /155 | 619x10**6 Kb sec  | 545x10**6 Kb sec | 417x10**6 Kb sec |
| /158 | 1140x10**6 Kb sec | -----            | -----            |

ISTIGRAMMA DI N. JCBS VS. CRUTIME



NUMERO CAST IN ISTIGRAMMA = 13101

ALL THE VALUES PLOTTED IN HISTOGRAM

FIG. 1

ISTOGRAMA DI R. JOBS VS. CRUTTIE

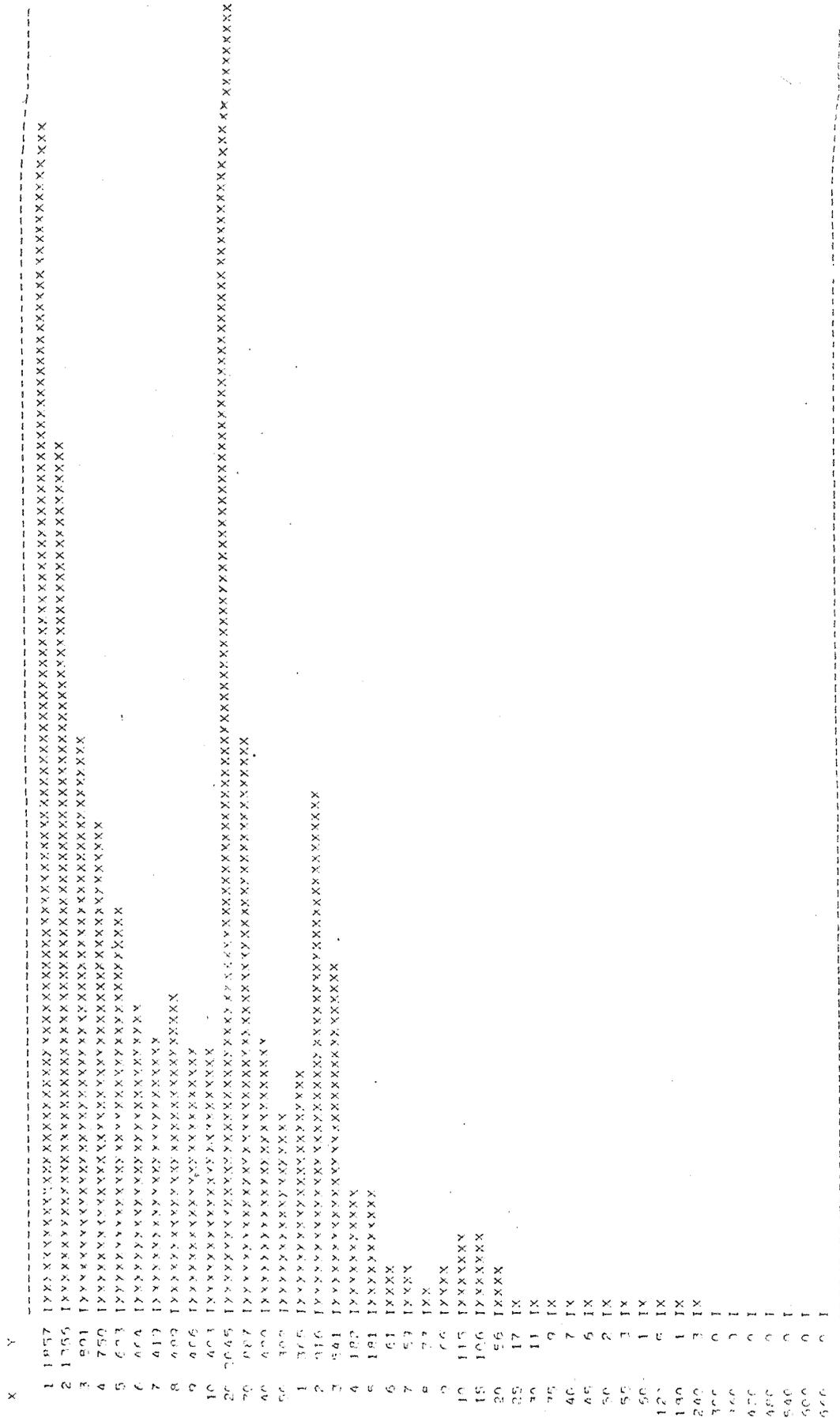


FIG. 2

ed hanno prodotto per

|    |        |       |
|----|--------|-------|
| A1 | 256 kb | 620kb |
| A2 | 201 kb | 257kb |
| M1 | 233    | --    |
| M2 | 176    | --    |
| U1 | 201    | --    |
| U2 | 152    | --    |
| F1 | 1.32   | 2.4   |
| F2 | 1      | --    |

Cio' significa che, in media, sul /155 e' occupato il 50% della memoria disponibile e solo il 75% di tale percentuale e' veramente utilizzata, essendovi un 10% di spreco dovuto all'utente e un 15% al metodo di allocazione per job; sul /158 si ha una occupazione media di 620 kb.

Il fattore di multiprogrammazione vale 1.32 per il /155, e 2.4 per il /158, con un rapporto di circa 1.77.

Gli istogrammi che mostrano il numero di steps versus la REGION allocata e quello relativo solo al /155 versus la REGION usata sono rispettivamente nelle figure 3,4,5.

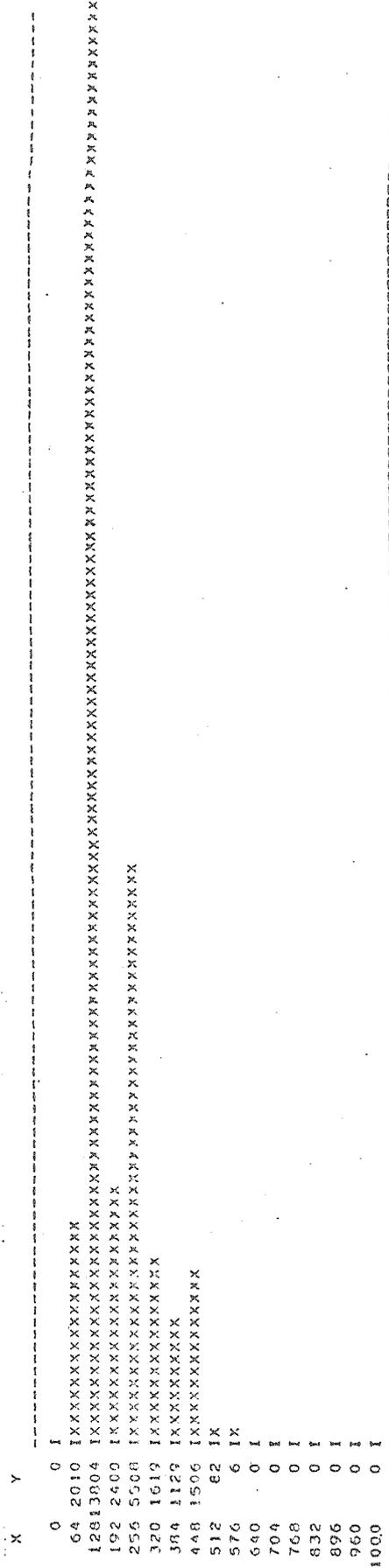
Per concludere, si vuole esaminare l'andamento dei valori trovati precedentemente nell'arco della giornata.

In fig. 6 e' espressa, rispettivamente per il /155 e il /158, nelle 24 ore, la media della memoria allocata, calcolata col criterio sopra esposto: la zona A corrisponde ad una zona di multiprogrammazione, che sul /158 e' completa, mentre sul /155 e' riservata solo a piccoli jobs; la zona B, invece, rappresenta il periodo in cui l'utenza deve individuarsi, sul /155, nel rilascio dei jobs piu' grandi e nei jobs "notturni" (vedi appendice), mentre, sul /158, in programmi particolarmente CPU-bounded, utenza particolare e jobs "notturni". Si notino, per il /158, i picchi che rappresentano, dopo le 18, l'inizio del rilascio, e, sulle 4, l'utenza speciale.

In figg. 7,8 sono invece evidenziati, in secondi, i tempi in cui esiste in memoria centrale almeno un job; la fig. 9 mostra l'andamento di tale frequenza ottenuto come media in un'ora delle misure. Si noti come l'andamento relativo al /158 sia molto simile a quello relativo al /155, anche se sensibilmente ridotto, specie nelle ore notturne.

Per quanto concerne i tempi di CPU, il rapporto di /notte, sul /155 e' circa 1.5 (la zona B tra le 13 e le 15 rappresenta circa il 15% del totale giornaliero), mentre sul /158 e' circa 3. Questo rapporto, mentre sul /155 non potrebbe aumentare in ogni caso con tali scelte di sistema e di gestione, sul /158 potrebbe essere molto piu' grande se i jobs che l'utenza, per ragioni di costo, vuole di notte, fossero "liberi".

ISTOGRAMMA DI N-STEPS VS. ALL-REG.

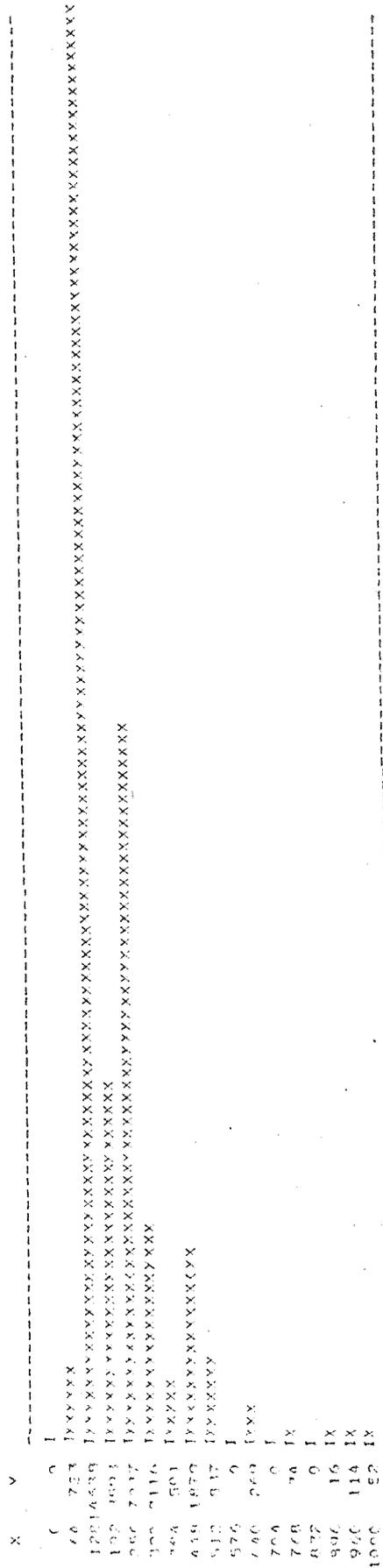


NUMERO CASI IN ISTOGRAMMA = 26064

ALL THE VALUES PLOTTED IN HISTOGRAM

FIG. 3

ISTOGRAMMA DI 16 STEPS VS. ALL. OFG.



NUMERO CASI IN ISTOGRAMMA = 32009

ALL THE VALUES PLOTTED IN HISTOGRAM

FIG. 4



MEMORIA ALLOCATA

XXX Kb MEDIA

/155

/158

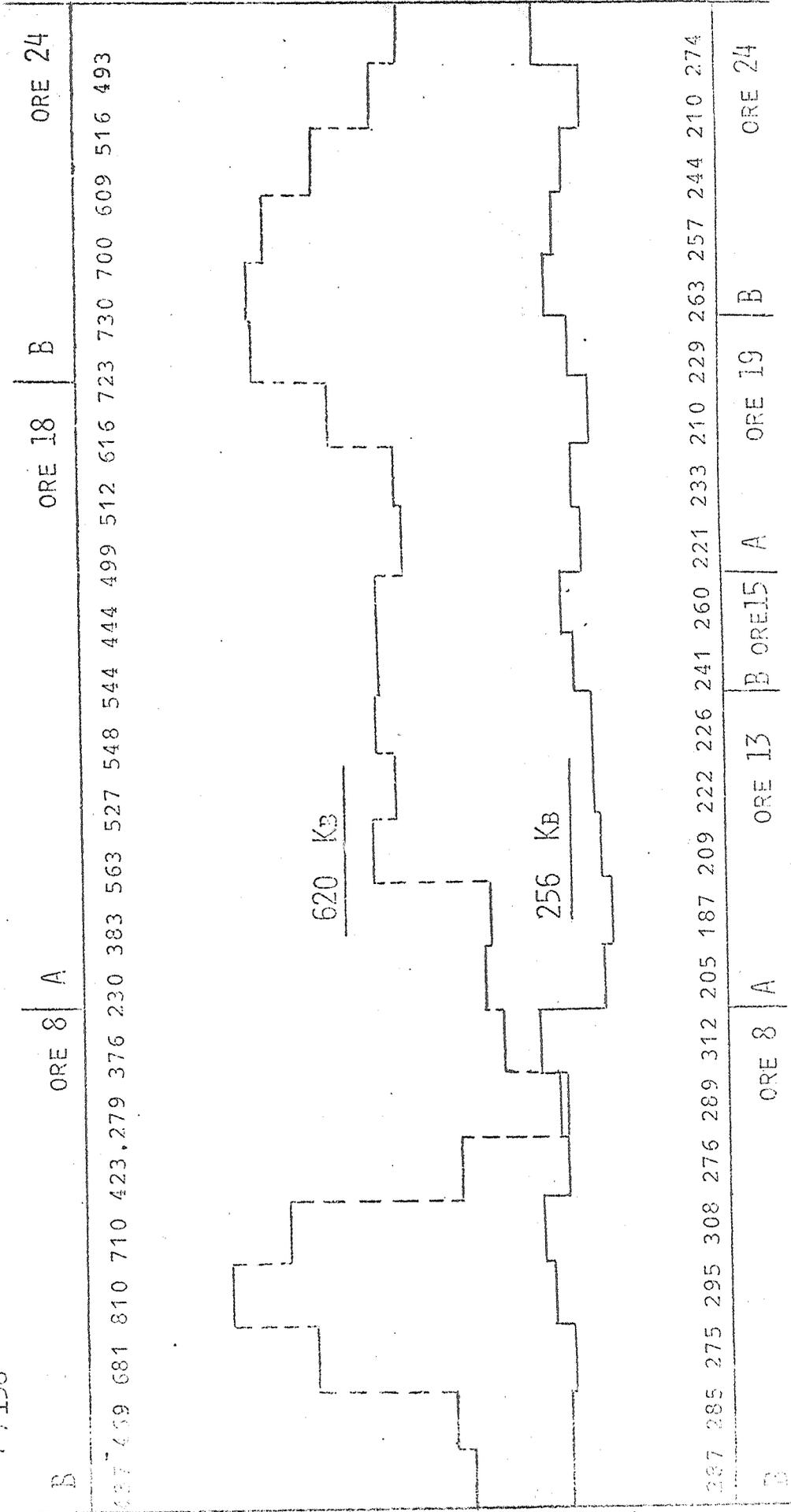


FIG. 6





FREQUENZA DI OCCUPAZIONE DELLA MEMORIA DA PARTE DI ALMENO UN JOB

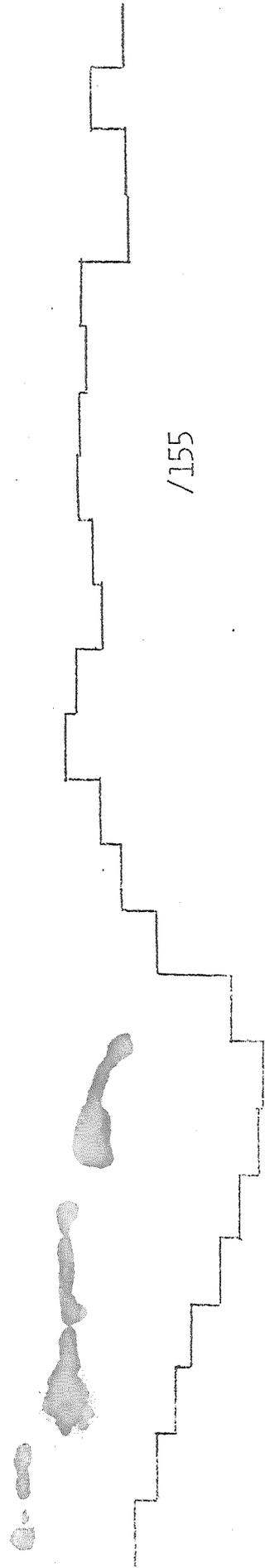
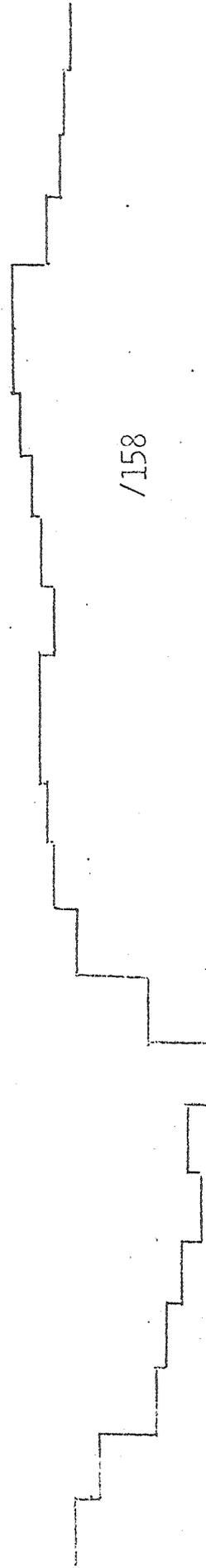


Fig. 9

## Conclusioni

Nonostante i valori precedentemente mostrati rappresentino di per se' un indice sufficientemente espressivo, si vuole concludere cercando di tradurre questi valori in risposte ad alcune domande idonee ad indagare l'incidenza della risorsa "memoria" nell'economia generale di questi sistemi.

Sono la memoria e i processi che ad essa sono legati un "collo di bottiglia" per un ambiente "multiprogramming batch" con carico non tempificabile?

Il sistema /155 con OS-MVT/HASP, sia per la filosofia generale di gestione, sia per le scelte che tale filosofia impone di effettuare, ha dimostrato una spiccata criticita' della risorsa "memoria"; l'occupazione media molto ridotta della memoria centrale, la distribuzione della stessa, maggiore di notte che nel di', la multiprogrammazione effettuata solo su piccoli jobs e mediamente inesistente, hanno prodotto frequenti residui per difetto di carico, tempi di risposta dei jobs di sensibile carico enormi, soffocamento delle richieste potenziali e in conclusione un livello di efficienza generale sostanzialmente limitato. Si potrebbe obiettare che cio' sia accaduto per mancanza di sollecitazione ma questa affermazione e' smentita dalle misure sul /158 che hanno mostrato una produzione superiore di circa il 30% rispetto alla precedente, dovuta in massima parte a jobs di sensibile grandezza.

E' cio' dipeso dalle differenti velocita' delle due CPU o dalla effettiva multiprogrammazione rilevata sul /158? I risultati ottenuti, per esempio 0.19 per la grandezza C2, dimostrano che la velocita' della CPU non costituisce in questo ambiente di carico una risorsa critica anche se, come sempre, qualche "risonanza" puo' essere attenuata con velocita' piu' alte.

E' la memoria virtuale o l'aumento della memoria reale la risoluzione al problema?

Considerando l'impossibilita' di tempificazione del lavoro, l'aumento della memoria reale avrebbe senz'altro rappresentato un consistente miglioramento all'efficienza generale ma non avrebbe potuto assorbire i picchi di carico che impongono l'elasticita' della memoria in ambiente multiprogrammato. Inoltre cosa sarebbe successo con carichi maggiori?

Si puo' dire che con carichi molto pronunciati, la memoria virtuale puo' cominciare a produrre anche effetti di trashing ma sono questi effetti effettivamente antieconomici o invece solo percentuali irrelevanti confrontati alle possibilita' di avere sempre un livello di efficienza elevato?

Mentre si vuole volutamente non indagare su questo ulteriore

problema, si conclude osservando che, come accennato nella introduzione, un sistema tipo /158 rispetto ad un sistema tipo /155 non solo si presenta piu' efficiente in assoluto, ma soprattutto non coinvolge cosi' direttamente nelle performance del sistema ne' l'utente, che rappresenta il richiedente della risorsa, ne' il gestore, che si trova costretto ad effettuare altrimenti scelte sempre difficili per parametrizzare il legame sistema-ambiente.

## Appendice.

### 370/155.

A causa dell' occupazione della memoria centrale da parte dell' OS/HASP, la memoria disponibile e' di poco superiore ai 500K.

L' utente deve specificare sulla scheda //JOB la memoria necessaria allo step piu' "grande" del job. Il sistema alloca per ogni step di questo job tale memoria (cio' e' stato realizzato per evitare che nell' allocazione di due step dello stesso job l' initiator possa rimanere "bloccato").

L' associazione job-classe viene fatta automaticamente da Hasp, in modo che la classe risulti dal prodotto della memoria centrale richiesta in kbytes e il tempo di CPU previsto in minuti, secondo questa tabella:

|               |     |     |     |     |      |       |
|---------------|-----|-----|-----|-----|------|-------|
| region x time | <64 | 224 | 416 | 672 | 1568 | >1568 |
| classe        | B   | H   | C   | D   | E    | G     |

Inoltre vi sono altre tre classi:

F per jobs >320 kbytes

T per jobs da elaborare di notte

W per i sistemisti

attribuibili indicando esplicitamente il parametro CLASS.

Le priorit  all' interno della classe vengono calcolate in base alle linee di stampa e al tempo di CPU richiesti, e incrementate nel tempo.

Allo scopo di favorire l' elaborazione dei jobs con richieste globali meno pesanti, e' stata adottata questa associazione classi-initiators:

|     |      |
|-----|------|
| IN1 | BH   |
| IN2 | BHC  |
| IN3 | HBCD |
| IN4 | B    |
| IN5 | W    |

Le classi non presenti in tale tabella vengono associate agli initiators dagli operatori, con il criterio di non influire sulla multiprogrammazione delle classi specificate sopra. In pratica cio' accade saltuariamente dalle 13 alle 15 e dopo le 19, momenti in cui l'utenza relativa ai piccoli jobs risulta assente.

Inoltre, i jobs che richiedono l' utilizzo di dispositivi on-line (in particolare nastri), vengono automaticamente posti in stato di HOLD da HASP, e rilasciati

a discrezione dell' operatore (cio' per evitare di bloccare initiators per mancanza di supporto o di unita').

370/158.

Tutti i jobs vengono considerati di classe A. Uniche eccezioni sono le classi W e T. Gli initiators sono 7 di classe A e uno di classe W.

L' operatore trattiene i jobs con richiesta di CPU-TIME molto alta, e associa le classi T agli initiators nel periodo notturno.