

Consiglio Nazionale delle Ricerche

Logica di funzionamento

del BATCH - CMS

del CNUCE

A. Pagni

126

CNUCE

Divisione Servizio Elaborazione Dati

A cura di : Andrea Pagni

Copyright - Maggio 1977

by - CNUCE - Pisa

Istituto del Consiglio Nazionale delle Ricerche

INDICE

Introduzione	1
Job validation ed accounting del Batch-CMS standard.	3
Limiti del Batch-CMS standard.	6
Il Batch-CMS del CNUCE	7
Logica del Batch-CMS del CNUCE	8

INTRODUZIONE

Il BATCH-CMS e' la versione batch del Conversational Monitor System. Si tratta di un batch sequenziale, senza possibilita' di montaggio di nastri o dischi, ma che ha il vantaggio di una buona velocita' e di una notevole semplicita' di uso.

Il BATCH-CMS gira su una o piu' macchine virtuali dedicate a questo servizio. Una macchina virtuale dedicata al BATCH-CMS e' una normale macchina CMS, con il vincolo di dover avere un minidisco di indirizzo virtuale '95, che viene utilizzato come minidisco di lavoro ed il cui contenuto viene cancellato alla fine di ogni job, e di dover avere l'opzione ACCOUN™ definita a livello di directory.

Tale macchina viene fatta partire dall'operatore di sistema facendone LOGON su di un terminale e dando i comandi IPL CMS e CMSBATCH, oppure il solo comando IPL CMS PARM BATCH, dopo di che, in generale, l'operatore disconnette la macchina virtuale dal terminale.

La macchina puo' anche essere fatta partire direttamente disconnessa mediante il comando di operatore AUTOLOG, sempre che abbia l'IPL CMS definito a livello di directory ed un comando di IPL CMS PARM BATCH all'interno del PROFILE EXFC.

Qualunque sia il modo scelto per farla partire, in una macchina virtuale dedicata al BATCH-CMS viene dapprima chiamata una routine di bootstrap, DMSBTB, che ha il compito di caricare in memoria il Batch Processor, DMSBTP e due eventuali exit-routines, BATEXIT1 e BATEXIT2, che non vengono fornite con il sistema ma sono da scriversi a cura dell'installazione. Al termine del caricamento DMSBTB cede il controllo a DMSBTP che inizia l'elaborazione del job. Alla fine del job, DMSBTP lancia il comando IPL CMS PARM BATCH che ha l'effetto di re-iplare il CMS e richiamare di nuovo DMSBTB.

Il Batch Processor, DMSBTP, quando riceve il controllo da DMSBTB tenta dapprima una lettura dal card-reader. Se questo e' vuoto si mette in attesa di un interrupt che gli segnali l'arrivo di un file. Se il card-reader non e' vuoto, DMSBTP legge una scheda alla volta fino a che non trova una scheda JOB. Una volta trovata una scheda JOB, e se questa e' valida, DMSBTP legge una alla volta le successive schede e le passa al CMS simulando che siano righe immesse da terminale. E' chiaro quindi che il BATCH-CMS si comporta come un CMS conversazionale a cui i comandi siano passati via card-reader invece che via terminale. Quando DMSBTP

incontra una scheda /*, la interpreta come end of-job e lancia il comando IPL CMS PARM BATCH.

Da notare che una macchina virtuale dedicata al BATCH-CMS produce sempre, oltre alle uscite su stampante o su perforatore eventualmente richieste dal job, una ulteriore uscita che riporta fedelmente i comandi contenuti nel job e le risposte del CMS ai comandi. Questa uscita su stampante non e' altro che il foglio di console che si avrebbe battendo direttamente a terminale i comandi contenuti nel job, ed e' indispensabile per determinare eventuali cause di errori.

JOB VALIDATION ED ACCOUNTING DEL BATCH-CMS STANDARD

Per 'il BATCH-CMS standard la scheda job ha il seguente formato:

/JOB nome-mv codice [nome-job]

- nome-mv: e' il nome di una macchina virtuale che deve essere definita nel directory;
- codice: e' il codice di addebito con cui si vuole passare il job;
- nome-job: e' il nome del job (opzionale, se omesso viene assunto "JOB").

Quando DMSBTP trova la scheda job, la passa immediatamente alla exit routine BATEXIT2, se questa esiste ed e' stata caricata. Se BATEXIT2 esiste, al ritorno da questa DMSBTP esamina il registro 15. Se questo non e' zero la scheda job e' ritenuta invalida e DMSBTP si mette alla ricerca di una nuova scheda job leggendo dal card-reader. Se al ritorno da BATEXIT2 il registro 15 e' zero, oppure se BATEXIT2 non esiste, DMSBTP lancia una diagnosi di codice 4C e sottocodice 4, passando come parametri al CP "nome-mv" e "codice".

L'effetto di tale diagnosi e' il seguente:

- 1) Il CP chiude ogni eventuale record di addebito che aveva precedentemente aperto per la macchina virtuale che ha lanciato la diagnosi;
- 2) Il CP cerca nel directory una macchina virtuale che abbia per nome "nome-mv". Se la trova apre un record di addebito per la macchina virtuale che ha lanciato la diagnosi, addebitando pero' le risorse che saranno consumate nel seguito e fino alla prossima diagnosi al nome "nome-mv" ed al codice "codice". Ritorna quindi il controllo alla macchina virtuale che ha lanciato la diagnosi con condition code 0.
- 3) Se il CP non trova nel directory una macchina virtuale che abbia per nome "nome-mv", ritorna il controllo alla macchina virtuale che ha lanciato la diagnosi con c.c.?

Al ritorno dalla diagnosi DMSBTP testa il c.c. e se e' diverso da zero la scheda job viene ritenuta invalida e viene iniziata la ricerca di una nuova scheda job come descritto sopra. Se il c.c. e' zero la scheda job e' ritenuta valida e DMSBTP lancia un messaggio alla macchina

virtuale di nome "nome-mv". Tale messaggio e':

```
JOB "nome-job" STARTED
```

contemporaneamente vengono aperte in modo continuo le uscite di console, stampante e perforatore.

DMSBTP controlla poi se la scheda successiva ha il formato:

```
/SET |TIME secondi| |PRINT linee| |PUNCH schede|
```

dove:

"secondi", "linee", "schede" possono assumere valori da 0 a 32767. Se incontra questa scheda, DMSBTP confronta i valori specificati per tempo di CPU, linee e schede con quelli contenuti in una tabella di label BATLIMIT. Se i valori scritti sulla scheda SET sono tutti non maggiori di quelli specificati in BATLIMIT, i valori della scheda vengono sostituiti ai valori di BATLIMIT. Se almeno uno e' maggiore, il job viene terminato e la macchina virtuale "nome-mv" riceve il messaggio:

```
JOB "nome-job" ABEND
```

Sull'uscita di console, che viene iniziata non appena e' riconosciuta valida la scheda job, appare il messaggio:

```
/SET CARD FORMAT INVALID
```

e DMSBTP re-ipla il CMS.

Se la scheda SET non c'e' o c'e' ed e' valida, DMSBTP legge la scheda seguente dal card-reader e la passa alla exit-routine BATEXIT1, se questa esiste ed e' stata caricata. Se al ritorno di BATEXIT1 il registro 15 e' zero, oppure se BATEXIT1 non esiste, la scheda viene trascritta sull'uscita di console ed e' poi passata al CMS, che la elabora come se fosse un comando dato da terminale, ponendo a console, e quindi sull'uscita di console, le risposte e gli eventuali codici di errore. Alla fine dell'esecuzione del comando DMSBTP legge la prossima scheda e cosi' via. Questo processo termina quando:

- 1) DMSBTP legge una scheda /*, a questo punto viene mandato alla macchina virtuale "nome-mv" il messaggio:

```
JOB "nome-job" ENDED
```

vengono chiuse tutte le uscite di console, stampa e

perforazione e viene lanciato il comando:

```
IPL CMS PARM BATCH
```

- 2) il tempo di CPU consumato o il numero di linee stampate o il numero di schede perforate ha superato il proprio limite specificato in BATLIMIT. A questo punto viene mandato alla macchina virtuale "nome-mv" il messaggio:

```
JOB "nome-job" ABEND
```

sull'uscita di console viene scritto il messaggio:

```
|CPU |  
|PRINT| LIMIT EXCEEDED  
|PUNCH|
```

vengono chiuse tutte le uscite di console, stampa e perforazione e viene lanciato il comando:

```
IPL CMS PARM BATCH
```

- 3) uno dei comandi contenuti nel job provoca un ABEND di CMS, in questo caso viene mandato alla macchina virtuale "nome-mv" il messaggio:

```
JOB "nome-job" ABEND
```

vengono chiuse tutte le uscite di console, stampa e perforazione e viene lanciato il comando:

```
IPL CMS PARM BATCH
```

In ambedue i casi il processo inizia di nuovo con la ricerca di una scheda job. Il record di addebito aperto all'atto della convalida della scheda job viene chiuso o quando viene incontrata una nuova scheda job o, se il card-reader e' vuoto, subito prima che DMSBTP si metta in attesa dell'interrupt dal card-reader.

LIMITI DEL BATCH-CMS STANDARD

Senza dubbio i piu' grossi limiti del BATCH-CMS sono la sequenzialita' e l'impossibilita' di avere il montaggio di nastri o dischi, tuttavia, anche accettandoli, restano alcune grosse lacune. Un loro approssimativo elenco puo' essere il seguente:

- 1) jobs che abbiano la scheda JOB riconoscibile come tale ma con parametri errati (per esempio con un nome di macchina virtuale non esistente), spariscono senza lasciare traccia; cioe' il job viene soppresso ma non viene prodotto alcun output, ed il System Operator non viene informato.
- 2) i limiti di tempo consumato, righe stampate, schede perforate sono uguali per tutti i jobs, poiche' la tabella BATLIMIT e' interna a DMSBTP e l'unico modo per cambiarla e' ricompilare DMSBTP.
- 3) i codici di addebito sono lasciati alla fantasia dell'utente; infatti il CP, quando riceve la diagnosi di addebito, controlla solo che esista nel directory la macchina virtuale specificata nella scheda job e costruisce il record di addebito con il codice specificato nella scheda stessa.
- 4) per consumare risorse di sistema a spese altrui basta conoscere il nome (e non la password!) di una macchina virtuale nel directory.
- 5) non risulta su alcun output quanto tempo di CPU ha consumato il job.
- 6) l'unica macchina virtuale che puo' ricevere (se e' collegata) i messaggi di inizio e fine job e' quella specificata sulla scheda job; non esistono altri metodi per sapere cosa stia facendo la macchina virtuale che gira il BATCH-CMS; neanche il System Operator ha possibilita' di avere informazioni.
- 7) non vi e' modo di stabilire, per un particolare codice o gruppo di codici, un limite massimo di tempo di CPU "a consumo", cioe' che venga decrementato alla fine di ogni job.
- 8) sebbene ci sia la possibilita' di mandare gli output sui lettori di macchina virtuale, a stazioni remote o su devices di output particolari, occorre codificare per esteso tutti i comandi relativi.
- 9) non comparando in alcun luogo sulla mascherina degli output il nome del job, gli output di jobs inviati da uno stesso utente sono indistinguibili fra loro.

THE UNIVERSITY OF CHICAGO

THE UNIVERSITY OF CHICAGO
DEPARTMENT OF CHEMISTRY
5800 S. UNIVERSITY AVENUE
CHICAGO, ILLINOIS 60637
TEL: (773) 835-3100
FAX: (773) 835-3101
WWW: WWW.CHEM.UCHICAGO.EDU

THE UNIVERSITY OF CHICAGO
DEPARTMENT OF CHEMISTRY
5800 S. UNIVERSITY AVENUE
CHICAGO, ILLINOIS 60637
TEL: (773) 835-3100
FAX: (773) 835-3101
WWW: WWW.CHEM.UCHICAGO.EDU

THE UNIVERSITY OF CHICAGO
DEPARTMENT OF CHEMISTRY
5800 S. UNIVERSITY AVENUE
CHICAGO, ILLINOIS 60637
TEL: (773) 835-3100
FAX: (773) 835-3101
WWW: WWW.CHEM.UCHICAGO.EDU

THE UNIVERSITY OF CHICAGO
DEPARTMENT OF CHEMISTRY
5800 S. UNIVERSITY AVENUE
CHICAGO, ILLINOIS 60637
TEL: (773) 835-3100
FAX: (773) 835-3101
WWW: WWW.CHEM.UCHICAGO.EDU

LOGICA DEL BATCH-CMS DEL CNUCE

Le modalita' di caricamento, inizializzazione e ricerca della scheda job sono identiche a quelle standard.

La scheda job ha il seguente formato:

```
/JOB nome-mv codice nome-job |dest stampa |dest perforazione||
```

I primi tre parametri hanno il significato standard, ma "nome-job" e' obbligatorio, "dest-stampa" e' un parametro formato da due campi separati tra di loro da almeno un blank, che serve a specificare la destinazione degli output di stampa e di console, con il seguente significato:

\$ xxxxxxxx: gli output devono andare sul lettore della macchina virtuale di nome xxxxxxxx

= xxxxxxxx: gli output sono accodati sul remoto di nome xxxxxxxx

& c : gli output sono accodati sulla stampante locale di classe c

% % : nessun effetto (stampa locale normale)

"dest-perforazione" ha lo stesso significato per la destinazione dell'output di perforazione.

Qualora si debba specificare "dest perforazione" e non "dest stampa", quest'ultimo deve essere specificato con "% %".

Quando DMSBTP trova una scheda che contiene /JOB ed almeno tre ulteriori campi, la interpreta come scheda job e la passa immediatamente a BATEXIT2.

BATEXIT2 ha al suo interno una tabella con la seguente struttura:

JOBINFO	DS	0D
JOBMSGID	DC	CL8' '
JOBACCT	DC	CL8' '
JOBNAME	DC	CL8' '
JOBCTLCO	DC	CL8' '
JOBDESCO	DC	CL8' '
JOBCTLPR	DC	CL8' '
JOBDESPR	DC	CL8' '
JOBCTLPU	DC	CL8' '
JOBDESPU	DC	CL8' '
*		
SCHEDA	DS	0C133
CODE	DC	CL4' '
LIM	DC	CL1' '
TOTCPU	DC	CL5' '
LINES	DC	CL5' '
CARDS	DC	CL5' '
JOBCPU	DC	CL5' '
CODEPASS	DC	CL8' '

ed appena viene chiamata trasferisce il contenuto della scheda job a partire dal campo JOBMSGID. La coppia di parametri "dest stampa" viene riportata sia a partire da JOBCTLCO che a partire da JOBCTLPR mentre la coppia "dest perforazione" viene riportata a partire da JOBCTLPU; l'indirizzo di tale tabella viene quindi salvato in una locazione di bassa memoria.

Viene quindi agganciato in lettura/scrittura un minidisco su cui si trova il file che contiene l'elenco dei codici autorizzati, ciascuno con le sue risorse spendibili; BATEXIT2 legge sequenzialmente i records del file nel buffer SCHEDA all'interno della tabella JOBINFO. Ogni record del file e' lungo 33 bytes ed i records sono organizzati in ordine alfabetico rispetto al campo CODE. I significati dei vari campi sono i seguenti:

CODE	:	e' il codice di addebito
LIM	:	puo' essere L o blank
TOTCPU	:	se LIM=L e' il tempo di CPU spendibile globalmente da tutti i jobs con codice CODE se LIM=blank e' il massimo tempo di CPU spendibile da ogni job
LINES	:	e' il massimo numero di linee che il job puo' stampare
CARDS	:	e' il massimo numero di schede che il job puo' perforare
JOBCPU	:	ha senso solo se LIM=L e' il massimo tempo di CPU spendibile da ogni job
CODEPASS:		e' la password associata al codice.

BATEXIT2 legge sequenzialmente i records ed ad ogni lettura confronta i primi quattro caratteri di JOBACCT con CODE. Se leggendo si arriva alla fine del file od al punto in cui CODE segue JOBACCT in ordine alfabetico, il codice viene considerato invalido. In questo caso viene prodotto il messaggio:

```
JOB "jobmsgid jobacct jobname" INVALID ACCT CODE
```

tale messaggio:

- 1) viene inviato a "jobmsgid"
- 2) viene inviato al System Operator se questo non e' "jobmsgid"
- 3) viene stampato in un output di console che viene inviato alla destinazione specificata in "dest stampa".

Inoltre BATEXIT2 ritorna il controllo a DMSBTP con il registro 15 diverso da zero, il che provoca la ricerca di una nuova scheda job.

Nel caso invece che si trovi un record per cui CODE e' uguale ai primi quattro caratteri di JOBACCT, viene testato il campo TOTCPU. Nel caso che questo sia zero, viene prodotto il messaggio:

```
JOB "jobmsgid jobacct jobname" NO MORE CPU TIME
```

Tale messaggio viene comunicato come detto sopra ed anche in questo caso BATEXIT2 ritorna il controllo a DMSBTP con il registro 15 diverso da zero.

Nel caso infine che TOTCPU non sia zero, vengono forzati nella tabella BATLIMIT, all'interno di DMSBTP, i limiti specificati nel record relativo al codice, e precisamente:

- 1) come tempo di CPU viene preso il minimo fra TOTCPU e JOBCPU (se quest'ultimo non e' blank, altrimenti viene preso TOTCPU);
- 2) come numero di linee di stampa viene preso il campo LINES;
- 3) come numero di schede perforate viene preso il campo CARDS.

BATEXIT2 mette poi ON un flag in bassa memoria per indicare che, avendo trovata una scheda JOB valida, e' attesa una scheda SET. Viene inoltre testato il campo CODEPASS e, se questo non e' blank, viene messo ON un ulteriore flag per

indicare che e' attesa anche la password associata al codice.

BATEXIT2 lancia poi una diagnosi di addebito 4C con sottocodice 20 (nuovo sottocodice introdotto nel CP), passando come parametri il nome del job, il codice di addebito ed ancora il codice di addebito rispettivamente come userid, codice di addebito e distribution code. L'effetto di tale diagnosi e', per quanto riguarda l'apertura e la chiusura dei records di addebito, uguale allo standard, pero' non viene fatto alcun controllo sul directory, quindi l'addebito viene fatto al nome del job invece che allo userid, ed in particolare la stampa esce con sulla mascherina il nome del job come userid ed il codice di addebito come distribution code.

Il passo successivo e' la chiamata, da parte di BATEXIT2, del comando PERMLoad (comando aggiunto al CMS), che provoca il caricamento in un'area con chiave NUCLFUS acquisita tramite DMSFFFE della routine BATCHEXT.

Viene quindi chiamata la routine TRAP fornendogli l'indirizzo di caricamento di BATCHEXT, per dichiarare al CMS che deve passare il controllo a quella locazione di memoria ogni volta che arriva una interruzione di external che non sia di timer. Nella lista di chiamata c'e' un flag che indica a DMSINT di non annullare questa richiesta di gestione alla fine del comando (modifica introdotta nel CMS).

A questo punto BATEXIT2 emette una diagnosi di codice 68, sottocodice 0, cioe' la diagnosi di AUTHORIZE della Virtual Machine Communication Facility, per dichiarare che da quel momento la macchina BATCH-CMS accetta messaggi di VMCF.

Lo scopo di tutto questo e' di permettere agli utenti, tramite la VMCF, di conoscere lo stato della macchina BATCH-CMS e di permettere al System Operator, sempre tramite la VMCF, di terminare l'esecuzione di un job senza bisogno di riconnettere la macchina BATCH-CMS.

Viene successivamente chiamata la routine CMSTIME per avere l'ora di inizio job e virtual e total CPU time al momento della partenza del job. Viene quindi rilasciato il disco che contiene il file dei codici autorizzati ed il controllo viene passato a DMSBTP con il registro 15 a zero.

DMSBTP esegue le normali operazioni di inizio job, esclusa la diagnosi di addebito che e' gia' stata emessa da BATEXIT2, legge quindi una scheda dal lettore e la passa a BATEXIT1.

BATEXIT1 per prima cosa controlla un flag in bassa memoria. Se questo e' OFF lo mette ON ed emette tutti i comandi di SPOOL e TAG necessari per soddisfare le richieste

descritte dai campi "dest stampa" e "dest perforazione" riportate nella tabella JOBINFO.

Fatto questo, o se il flag era già ON, controlla se è ON il flag che indica che è richiesta una scheda SET. Se questo è ON verifica se la scheda passatagli da DMSBTP è una scheda SET. Se non lo è viene scritto nell'output di console il messaggio:

MISSING SET CARD

quindi BATEXIT1 salta direttamente alla routine diabend di DMSBTP. Se la scheda è una scheda SET si procede come segue. La scheda SET per il BATCH-CMS del CNUCE ha il seguente formato:

```
/SET |TIME xxxxx| |PRINT xxxxx| |PUNCH xxxxx| |COPY xx| |PASS xxxxxxxx|
```

e per i parametri TIME, PRINT, PUNCH il trattamento è identico a quello fatto da DMSBTP nel BATCH-CMS standard, compresa la chiamata della routine diabend nel caso che uno dei limiti superi il corrispondente valore trovato in BATLIMIT.

Il parametro COPY provoca l'emissione del comando CP SPOOL E COPY xx, il parametro PASS non ha alcun effetto se il flag che indica che è richiesta una password è OFF. Se invece è ON si fa il confronto fra il campo CODEPASS della tabella JOBINFO ed il parametro successivo a PASS nella scheda SET. Se il confronto è negativo viene fatta una chiamata della routine diabend dopo aver prodotto il messaggio:

PASSWORD MISSING OR INVALID

se il confronto è positivo viene messo OFF il flag relativo alla password. Alla fine del trattamento della scheda SET, BATEXIT1 controlla se il flag della password è ON. Se si fa una chiamata diabend dopo aver prodotto il messaggio precedente, altrimenti mette OFF il flag relativo alla scheda SET e ritorna il controllo a DMSBTP con il registro 15 diverso da zero per segnalargli di non passare la scheda al CMS.

Da qui in poi l'esecuzione del job continua nel modo standard, eccetto che ogni scheda letta viene passata a BATEXIT1, che torna sempre il controllo con il registro 15 a zero, escluso che sulla scheda si trovi un comando SPOOL. Il controllo della destinazione degli output deve infatti essere affidato alle specifiche sulla scheda job.

Come nel BATCH-CMS standard l'esecuzione del job termina o perché viene letta una scheda /*, o perché vengono superati i limiti riportati in BATLIMIT, o perché c'è un

abend di qualche natura. In piu, nel BATCH-CMS del CNUCE, il job puo' terminare perche' il System Operator, tramite la VMCF, manda un messaggio che provoca la terminazione del job. In tutti i casi DMSBTP richiama di nuovo BATEXTT2 all'entry BATEENTR1.

Questa fa una nuova chiamata di CMSTIME per avere l'ora di fine job nonche' la differenza fra virtual e total CPU time attuali ed i relativi valori al momento della prima chiamata, e scrive queste informazioni nell'output di console.

Poi se il campo LIM della tabella JOBINFO e' blank il controllo ritorna a DMSBTP, altrimenti, se LIM contiene una L, viene di nuovo acceduto il disco che contiene il file dei codici autorizzati, viene sottratto il total CPU time consumato dal job dal campo TOTCPU di JOBINFO, l'intero record contenuto nel buffer SCHEDE viene sostituito al corrispondente record del file, viene rilasciato il disco, viene stampato il messaggio:

TEMPO RESIDUO xxxxx SECONDI

nell'output di console ed il controllo ritorna a DMSBTP che fa le normali operazioni di fine job che si concludono con un nuovo IPL.

