

Il controllo della concorrenza

di Anna Pugliese

prima puntata

Sistemi distributivi, programmazione parallela, computer multiprogrammabili e multiutenza.

Questi, in sintesi, i campi in cui il problema del controllo della concorrenza occupa una posizione di rilievo. Ma quali sono i problemi che nascono dalla concorrenza, e quali i meccanismi per controllarli?

Di che si tratta

Innanzitutto è doveroso dare una definizione di che cos'è la concorrenza sui computer; attività, lo avrete già intuito, ben diversa da quella presente nel commercio.

La cosa è fattibile purché ci mettiamo d'accordo sul fatto che, come accade sempre nel campo, assai giovane, dell'informatica, non c'è niente di più possibile del trovarsi davanti ad un libro che riporta una definizione diversa (ci auguriamo nella forma e non nel contenuto).

Una volta «messe le mani avanti», è con leggerezza d'animo che mi accingo a dire che: su un sistema di elaborazione si svolgono attività concorrenti, quando esistono più processi, eseguiti in maniera non-sequenziale, che concorrono per l'acquisizione dell'uso di un qualcosa che chiameremo RISORSA.

MF507

Giochiamo un po' con l'immaginazione...

Venerdì 1 aprile 1988. Fra due giorni è Pasqua. L'aeroporto di Fiumicino è gremito di gente e quasi tutti i voli sono al completo.

«I passeggeri in partenza con il volo Magic Fly 507 sono pregati di affrettarsi al gate 9. Ripeto...». Sono già saliti tutti a bordo, ad eccezione del solito ritarda-

tario. Eccolo lì! Dev'essere uno di quei due signori che corrono gambe all'aria verso il gate 9.

«Buonasera, sono il Sig. Rossi, ho una prenotazione su questo volo, sono ancora in tempo?»

«Certo Sig. Rossi, si accomodi. Lei?»

«Buonasera, sono il Sig. Bianchi, anch'io sono prenotato»

«Mi scusi Sig. Bianchi, le dispiace darmi la sua prenotazione? Dunque, mi faccia un po' vedere. Ma, non capisco, dev'esserci qualcosa che non va, il volo, per quanto mi risulta è già al completo».

Mi auguro che il giallo vi sia piaciuto. Comunque sia chiudiamo il sipario ed andiamo a vedere cos'è successo.

Il Sig. Rossi aveva prenotato, la sera del giovedì, dall'aeroporto di Milano, verso le 19.00, pochi istanti prima della prenotazione del Sig. Bianchi, fatta da un'agenzia di Roma. In altre parole, da due terminali erano state invocate, sul computer cui entrambi erano collegati, due attivazioni della stessa procedura, descritta in figura 1.

La procedura fa uso di una matrice bidimensionale FREE, a componenti intere, in cui FREE (X, Y) contiene il numero di posti liberi sul volo X per il giorno Y. Chiaramente FREE è una risorsa condivisa da tutti i processi che la devono utilizzare, in altre parole essa è una struttura dati comune.

A prima vista, la procedura Prenota-

```

procedure PRENOTATION ( AZ507 , "01/04/88" );
begin
N := FREE ( AZ507 , "01/04/88" );
if N > 0 then
begin
FREE ( AZ507 , "01/04/88" ) := N-1;
Result := "prenotation done"
end
else
Result := "no-prenotation"
end.

```

Figura 1

T	;	PRENOTAZIONE DA MILANO	;	PRENOTAZIONE DA ROMA
	;	begin	;	-
Ti	;	N:=FREE(AZ507,"1/4/88")	;	begin
Ti+1	;	if N>0 then begin	;	N:=FREE(AZ507,"1/4/88")
Ti+2	;	FREE(AZ507,"1/4/88"):=N-1	;	if N>0 then begin
Ti+3	;	;	FREE(AZ507,"1/4/88"):=N-1
			

Figura 2

tion sembra ben fatta. Eppure vi assicuro che è proprio questa procedura la responsabile del pasticcio capitato al Sig. Bianchi. Vediamo come, in figura 2.

Osservando la figura, si comprende che tipo di pasticcio si è verificato, se noi consideriamo il caso in cui il valore di FREE(507, «1/4/88»), che d'ora in poi indicheremo brevemente con FREE, sia stato, nell'istante T_i , pari ad 1. Vale a dire che sul volo, era rimasto un solo posto disponibile. Questa disponibilità di un posto, è stata percepita sia dall'esecuzione invocata da Milano che da quella invocata da Roma, anche se in istanti diversi, legittimando in tutte e due le procedure l'acquisizione di un unico posto sul volo.

La difficoltà della soluzione

I problemi che nascono da un uso non controllato della concorrenza, sono problemi detti di «INCONSISTENZA», nel senso di non conformità alle specifiche del sistema.

Nell'esempio delle prenotazioni, l'inconsistenza è evidente dall'esame dei risultati ottenuti: $K+1$ prenotazioni su un volo di K posti non è un risultato conforme alle specifiche del sistema. Questa stessa inconsistenza tuttavia, non è ugualmente evidente dall'interno del sistema. E mi spiego. Se FREE finisse con l'assumere, in seguito al pasticcio, il valore «-1», forse sarebbe ancora possibile salvare il salvabile, inserendo dei controlli sulla struttura con-

divisa, in modo da rilevare l'inconsistenza e segnalare all'esterno, dove ci si preoccuperà di annullare l'ultima prenotazione effettuata. Ma un attento esame della figura 2, è sufficiente per rendersi conto che il valore finale assunto da FREE sarà comunque zero, cioè un valore legittimo, essendo zero il valore dell'espressione $N-1$, assegnato a FREE nell'istante T_{i+3} . Intendiamoci: è sempre possibile utilizzare un diverso algoritmo, capace di assegnare «-1» come valore a FREE, in seguito alle due prenotazioni, ma anche se utilizzassimo questo nuovo algoritmo, credo che nessuno di noi avrebbe più il coraggio di riporre in esso un accettabile grado di fiducia, dopo aver visto la miserabile fine fatta dall'algoritmo precedente, a prima vista degno di tutto rispetto.

Le difficoltà della soluzione consiste dunque nella incapacità della nostra razionalità a sopravvivere situazioni così complesse come quelle che possono verificarsi nel caso in cui due processi, pensati in maniera indipendente, finiscono poi con l'agire in concorrenza sulle stesse strutture. In altri termini, per poterci fidare di un nuovo algoritmo, non ci basta sapere che nel caso sopra considerato, è capace di comportarsi meglio dell'algoritmo precedente, abbiamo bisogno di progettare questo nuovo algoritmo, in accordo ad una qualche metodologia che la nostra mente è capace di vedere come risolutiva del problema nel suo complesso. Tutto questo va detto senza dimenticare che il pro-

blema vero rimane: non era esattamente quello che volevamo, se siamo poi costretti a rincorrere i clienti fuori dall'agenzia, per strappargli di mano la prenotazione fittizia che pochi istanti prima gli avevamo consegnato con tanta cordialità. La soluzione dev'essere tale che queste cose non succedano.

Transizioni e risorse

Ricerare meccanismi per controllare la concorrenza, questo è il nostro problema. L'approccio migliore ad esso, consiste nell'individuare i termini del problema nel modo più generale possibile. A tal fine è opportuno individuare i soggetti e gli oggetti delle esecuzioni concorrenti. Questi sono, rispettivamente, le sequenze di operazioni appartenenti ad uniche e globali attività, e le strutture dati condivise dalle attività stesse. Chiameremo TRANSAZIONI le prime e RISORSE le seconde. In un certo istante possono essere in esecuzione sul sistema un numero imprecisato di transazioni.

Per riallacciarsi all'esempio precedente, diremo che le due attivazioni della procedura Prenotation, corrispondono a due diverse transazioni T_1 e T_2 , e le risorse coinvolte nelle transazioni sono N_1 , Result1 e FREE per quanto riguarda T_1 ; ed N_2 , Result2 e FREE per quanto riguarda T_2 . La ridenominazione delle variabili N e Result, è tesa a sottolineare la caratteristica di privatezza di tali risorse: N_1 è una variabile privata di T_1 , e

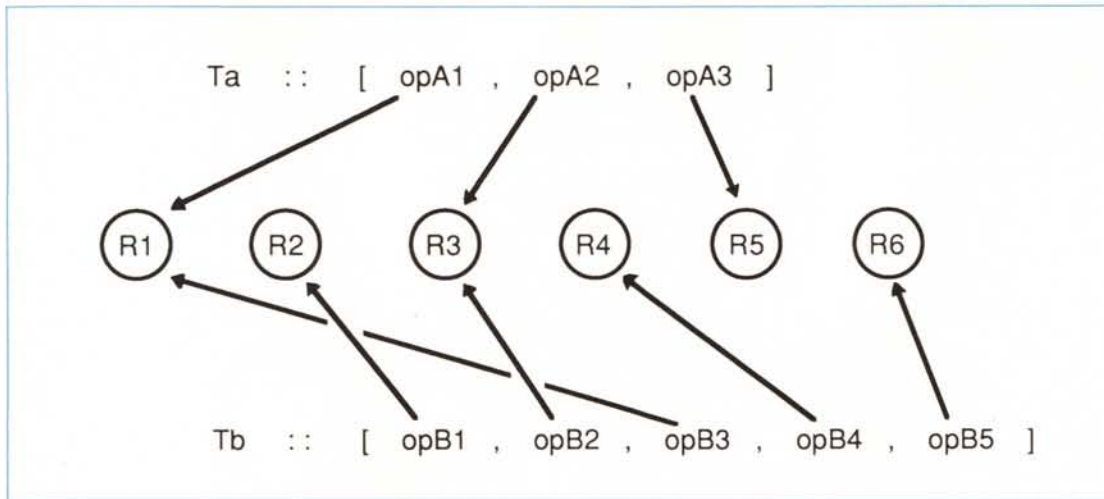


Figura 3
Due transazioni Ta e Tb agenti sulle risorse R1, ..., R6. Le frecce indicano su quale risorsa agisce ognuna delle operazioni.

perciò diversa da N2; diversamente per la risorsa FREE, che è unica e condivisa non solo da T1 e T2, ma anche da altre transazioni che potrebbero essere presenti sul sistema.

Osserviamo la figura 3.

Si comprende da tale figura come R1 ed R3 siano sicuramente condivise, essendo operanti su di esse, operazioni sia di Ta che di Tb. Dalla figura tuttavia non emerge la possibilità che anche R6 sia condivisa, non fra Ta e Tb, ma fra Tb ed una qualche altra transazione.

Per semplificare il problema, supporremo che tutte le risorse siano comunque risorse condivise, sulle quali, quindi, occorre operare in accordo alla metodologia che costituisce l'oggetto della nostra ricerca.

Questo operare in accordo ad una metodologia significa che le operazioni appartenenti ad una transazione, devono essere implementate in modo da assicurare, non solo che l'effettiva operazione sia portata a termine sulla risorsa, ma anche, e dal nostro punto di vista fondamentalmente, in modo da assicurare che eventuali interferenze sulla risorsa, da parte di altre transazioni siano portate avanti in modo controllato; al limite è possibile che la metodologia consista nell'impedire che tali interferenze possano verificarsi.

Nota

Di meccanismi di controllo della concorrenza, così come si suole dire in giro dei linguaggi di programmazione, se ne possono inventare più di uno al giorno. Il problema è poi quello di capire, da una parte fino a che punto siano affidabili, e dall'altra quanto siano efficienti.

Potrebbe sembrare che il primo pro-

blema sia più importante del secondo, ed è vero. Ma il guaio è che se un meccanismo di controllo della concorrenza mi introduce un overhead di efficienza maggiore dell'incremento di efficienza introdotto dall'uso della concorrenza, voi capite bene che il nostro lavoro è stato vano: è come se avessimo inventato i tacchi a spillo (e qui non sto a spiegarvi perché ce l'ho coi tacchi a spillo, in quanto la cosa esula sufficientemente dall'argomento trattato nell'articolo).

Orbene, è inutile nascondere, è accaduto spesso che si siano inventate proprio cose di questo genere, che magari si sono vendute lo stesso, perché portavano sopra un marchio di tutto rispetto, ma che alla fine incrementavano la performance del sistema solo nel caso in cui le transazioni venivano generate sul sistema in ordine di durata d'esecuzione, cioè se mentre era in esecuzione una transazione che dura TOT, veniva generata una transazione che dura invece TOT-1, quello che accadeva è che i meccanismi di controllo della concorrenza finivano per annullare entrambe le due transazioni e ricominciavano d'accapo l'esecuzione, stavolta nell'ordine giusto, incrociando le mani affinché nel frattempo nessuno generasse transazioni che durano TOT-2.

I meccanismi di controllo della concorrenza non basta che siano affidabili. Nondimeno l'affidabilità è, non solo necessaria ma anche difficile da ottenere, per cui alla fine è giusto dire che tutti quelli che ne hanno inventato qualcuno hanno sicuramente il loro merito. Il discorso precedente valeva solo per quelli che li hanno messi in commercio. Siccome noi non dobbiamo commercializzare niente, possiamo anche prescindere


re dall'efficienza della soluzione, e concentrarci solo sul discorso affidabilità.

La serializzabilità

Lo spazio a nostra disposizione purtroppo sta finendo, e non è possibile illustrare soluzioni concrete al problema del controllo della concorrenza, cosa che faremo sul prossimo numero della rivista. Tuttavia, il problema è stato individuato, e quello che è stato detto ci permette di gettare comunque le basi su cui poi saranno effettivamente costruite le soluzioni possibili. La considerazione da fare, e che ci permette di inquadrare definitivamente il problema, è tanto semplice quanto acuta: i problemi legati al controllo della concorrenza non si presentano nel caso in cui non esiste la concorrenza stessa. Lo so, lo so. È una considerazione banale, ma ve lo avevo detto. In assenza di concorrenza, le attività presenti sul sistema sono eseguite una dopo l'altra, cioè in maniera seriale.

Supponiamo allora di avere N transazioni. L'esecuzione concorrente di queste N transazioni, è sicuramente corretta se produce gli stessi risultati della loro esecuzione seriale. Per esprimere questa coincidenza nei risultati è stato coniato appositamente un termine: la serializzabilità.

Un modo per eseguire concorrentemente N transazioni in maniera corretta, è quello di garantirne la serializzabilità, cioè la coincidenza dei risultati ottenuti, con quelli di una qualunque fra le possibili esecuzioni seriali.

Abbiamo detto che questo è un modo, non è l'unico, ma è il più diffuso fra i meccanismi esistenti; gli altri sono solo oggetto di ricerca. A risentirci. 

GUERRA COMPUTER

Via Bissuola, 20/A - Mestre (VE) - Tel. 041/974944 - Fax 041/975735

Via Cairolì, 95 - Treviso

Via Vizzotto, 29 - S. Donà di Piave (VE)

OFFERTE DEL MESE:

ADATTATORE TELAMATICO	L. 39.000
MOUSE 1351 - GEOS C 64	L. 45.000
DIGITALIZZATORE VIDEO AMIGA	L. 135.000
DIGITALIZZATORE SUONO AMIGA	L. 135.000
NIKI CARTRIDGE II	L. 65.000
TAPPETINO ISOLANTE PER MOUSE	L. 15.000
DRIVE ESTERNO PER AMIGA	L. 270.000
EMULATORE C 64 PER AMIGA	L. 49.000
INTERFACCIA MIDI PER AMIGA	L. 55.000

DISCHETTI

DISCHETTI BULK 3 1/2 2S2D	L. 1.950
DISCHETTI BULK 5 1/4 2S2D	L. 800
DISCHETTI NASHUA 3 1/2 2S2D	L. 2.500
DISCHETTI NASHUA 5 1/4 2S2D	L. 1.600
DISCHETTI SCOTCH 5 1/4 2S2D	L. 1.550
DISCHETTI DIASPRON 3 1/4 2S2D	L. 3.200
DISCHETTI DIASPRON 5 1/4 2S2D	L. 2.800
DISCHETTI SELECT 5 1/4 1S2D	L. 1.350

STAMPANTI

STAR LC 10	L. 530.000
STAR ND 15	L. 900.000
STAR ND 10	L. 970.000
STAR NB 24+10	L. 1.370.000
STAR NB 24-15	L. 1.800.000
CITIZEN 120 D	L. 390.000
PANASONIC KX-P 1081	L. 480.000
NEC P 2200	L. 920.000
NEC CP 6	L. 1.350.000

HARDWARE

SCHEDA GRAFICA EGA	L. 420.000
SCHEDA MULTI I/O	L. 180.000
SCHEDA SERIALE	L. 61.000
SCHEDA PARALLELA	L. 44.000
SCHEDA PORTA JOYSTICK	L. 50.000
SCHEDA GRAFICA VGA	L. 580.000
SCHEDA HERCULES	L. 150.000
SCHEDA CGA	L. 150.000
SCHEDA FAX	L. 1.100.000

ACCESSORI

HANDY SCANNER	L. 550.000
GENIUS MOUSE PER PC	L. 160.000
MOUSE PHILIPS	L. 120.000
MIKI MOUSE PER PC	L. 135.000
MONITOR NEC MULTISYNC II	L. 1.400.000
COPRI PC AT XT	L. 25.000
FAX MURATA M-1	L. 1.950.000
JOYSTICK PER PC-IBM ANALOGICO	L. 40.000
STREAMER TAPE	L. 900.000
DATA SWITCH	L. 150.000

COMPUTERS

PC XT COMPATIBILE TURBO	
512K RAM 2 DRIVE 5 1/4	
MONITOR F.V. - TASTIERA	L. 1.250.000
PC AT COMPATIBILE	
512K RAM - 1 DRIVE 5 1/4	
H.D. 20 MEGA-MONITOR TASTIERA	L. 3.150.000

INOLTRE ABBIAMO DISPONIBILI LE SEGUENTI MARCHE: AMSTRAD, PHILIPS, COMMODORE, BONDWELL, IBICOMP.

SOFTWARE

ABBIAMO DISPONIBILE PRESSO LA NOSTRA SEDE UNA VASTA GAMMA DI PROGRAMMI PER I SEGUENTI COMPUTERS: COMMODORE AMIGA, COMMODORE C 64/C 128. ATARI ST, MSX 1 E 2. SCRIVETEICI O TELEFONATECI NELLE VARIE SEDI COSÌ POTRETE ESSERE SODDISFATTI SUL PROGRAMMA CHE PIÙ VI INTERESSA PER IL VOSTRO COMPUTER.

TUTTI I PREZZI SOPRA ELENCATI SONO COMPRESIVI DI IVA E PER LE SPESE DI SPEDIZIONE C'È UN ADDEBITO FORFETTARIO DI L. 8000. LA MERCE È GARANTITA UN ANNO DALLA DATA DI VENDITA. PREZZI SPECIALI PER QUANTITÀ E PER I RIVENDITORI. RICHIEDETE IL NOSTRO CATALOGO IN QUANTO NON POSSIAMO ELENCARE TUTTO CIÒ CHE ABBIAMO DISPONIBILE.

La tecnica dell'informazione e della comunicazione dell'AEG e la tecnica per ufficio della Olympia sono state conglobate nell'AEG Olympia per dar vita ad un nuovo settore d'attività, presente a livello internazionale, del gruppo tecnologico AEG.

Quante cose faremo insieme (con i nuovi computers Olystar).

Nel 1930 Olympia creava le sue macchine per scrivere: robuste, affidabili e con una mania, tutta tedesca, della qualità. Oggi le macchine per scrivere Olympia, a forza di lavorare sodo, hanno aguzzato talmente il cervello che... sono diventate dei computers, gli Olystar Olympia. L'evoluzione del prodotto ha camminato di pari passo con l'evoluzione della marca: Olympia è diventata AEG Olympia, per operare in modo ancora più competitivo a livello mondiale nel campo dell'alta tecnologia. Per questo gli Olystar offrono, su tutta la linea, tecnologia avanzata e qualità garantita.

Olystar 80. È un Personal di dimensioni molto compatte ma di grandi prestazioni, tali da rispondere anche alle esigenze aziendali più complesse, per le quali la velocità e i futuri sistemi operativi rappresentano criteri di valutazione fondamentali. L'alta frequenza di clock del processore 80386 (20Mhz), la capacità fino a 16 MB della memoria centrale, il disco rigido da 3 1/2", con capacità fino a 120 MB e tempi di accesso rapidi, l'ampia gamma di stampanti, di video e di elementi di comunicazione ampliano notevolmente le possibilità del sistema.

Olystar 80 è inoltre totalmente compatibile con programmi in ambiente MS-DOS, Xenix e OS2.

Olystar 60. Con due drive per floppy da 5,25" ed il disco rigido, Olystar 60 rappresenta, nella famiglia dei computer Olympia, il classico Personal AT. Grazie agli ampliamenti Hard e Soft, il sistema può essere integrato in tutti i sistemi attuali e futuri di elaborazione dati e di comunicazione. Il sistema operativo MS-DOS è indicato in modo particolare per l'utilizzo di programmi standard internazionali. Il sistema operativo Prologue multiuser e multitask è indicato soprattutto in ambienti di multiutenza in collegamento con terminali.

Olystar 50. Olystar 50 è particolarmente adatto per sistemi di Desktop Publishing e come Master-PC (Server) nella rete 3 COM proposta da Olympia. La multifunzionalità di Olystar 50 consente il collegamento ai servizi postali di Teletext, Videotext, Telefax, Datex P, chiamata diretta, nonché la comunicazione con grandi calcolatori e l'integrazione in reti locali.

Olystar 40. Olystar 40 risponde in modo ottimale alle esigenze

dei sistemi operativi MS-DOS e OS2, nonché ai campi di utilizzo classici dei Personal Computers, compresa la grafica. Olystar 40 può essere utilizzato sia come monoposto che come posto di lavoro in rete, nell'ambito della rete 3 COM proposta da Olympia per consentire l'utilizzazione in comune di programmi, dati e periferiche.

Olympia Italiana spa - viale Brianza 20 - 20092 Cinisello Balsamo

Se volete fare qualcosa con noi, cercateci sulle Pagine Gialle, alla voce "Vendita macchine per ufficio".

MS-DOS e OS2 sono marchi registrati Microsoft Co. - Xenix è marchio registrato Microsoft Co. e The Santa Cruz Operation Inc.

AEG
OLYMPIA