

Parte C – Allegato “Schema della simulazione”

Premessa

Coerentemente con l'approccio adottato nel testo “*Impianti, Reti e Sistemi Informatici*”, che privilegia la produzione dei modelli di impianto, rispetto all'applicazione di modelli già dati e quindi solo da risolvere, si è privilegiata la produzione indipendente di uno schema di simulazione, piuttosto che l'applicazione meccanica di quanto proposto sul testo “*Simulazione*”.

Si è cercato pertanto di modellare la simulazione, coerentemente con i requisiti di calcolo richiesti e con il tipo di impianto in esame (impianto di tipo chiuso); il modello creato non è generale come quello proposto nel testo “*Simulazione*”, ma con gli opportuni accorgimenti può essere esteso anche al caso di impianti di tipo aperto e/o a differenti requisiti di calcolo.

Per descrivere il modello di simulazione prodotto, si procede come segue:

- si elencano le ipotesi e le assunzioni preliminari;
- si mostrano l'inizializzazione e una sequenza significativa di 11 passi della simulazione;
- si descrive il meccanismo di calcolo delle varie misure richieste.

Infine si allega un esempio di esecuzione: l'output di stampa del programma di simulazione viene commentato e confrontato con quanto descritto precedentemente.

Ipotesi e assunzioni preliminari

Si fanno le seguenti ipotesi:

- si lavora su un impianto di tipo chiuso, per cui esiste un numero finito N di utenti (nel seguito si assumerà $N=5$ per semplicità);
 - un evento è modellato come un oggetto dotato di:
 - un identificativo ID univoco;
 - una sorgente SRC, da cui l'evento è originato;
 - una destinazione DST, su cui l'evento termina;
 - un tempo simulato CLK in cui l'evento avviene;
 - un riferimento FK all'eventuale evento “genitore”.
 - l'ID è un numero progressivo, rappresentativo dell'ordine temporale della creazione dell'evento; questa quale può avvenire in fase di inizializzazione o successivamente, come risultato del processamento di un altro evento; il progressivo non coincide quindi con la posizione nel calendario degli eventi, che riflette il tempo di occorrenza degli stessi;
 - gli utenti non possano inviare comandi quando sono in attesa di risposte (coerentemente con l'ipotesi di impianto chiuso);
 - per quanto riguarda le code dei centri di servizio, essi sono modellate in due modi, a seconda che si tratti di Terminali o Cpu/Disk:
 - per i Terminali T consideriamo gli “utenti in attesa”:
 - quando T è DST di un evento, l'evento viene rimosso (l'utente ha ricevuto la risposta, non è più in attesa e può generare un nuovo comando);
 - quando T è SRC di un evento, l'evento viene aggiunto (l'utente ha generato un comando di cui attende una risposta);
 - per CPU e DiskX consideriamo i “job in coda”:
 - quando CPU/DiskX è SRC di un evento, l'evento viene rimosso (il CdS ha terminato il job e ne può accettare un altro);
 - quando CPU/DiskX è DST di un evento, l'evento viene aggiunto (il CdS riceve un nuovo job da processare);
 - con le assunzioni precedenti, si ha che la somma dei “job nelle code” di CPU e DiskX è sempre pari al numero di “utenti in attesa” su T ;
-

- le code di CPU e DiskX sono gestite con disciplina di servizio FIFO; per i Terminali ogni utente attende la risposta al proprio comando;
- l'unita' di misura del tempo simulato e' il clock di processore ed in questa unita' viene espresso anche il tempo di generazione dei comandi da parte degli utenti;
- la lista degli eventi (il "Calendario") e' implementato in modo da mantenersi ordinato, quando viene inserito un nuovo evento, rispetto ai tempi simulati in cui tali eventi avvengono;
- al fine di eliminare la polarizzazione iniziale ("*bias*") si sceglie di scartare tutte le misure eventualmente effettuate prima che tutti gli utenti abbiano inoltrato almeno un comando ("*primo arrivo*" al sistema);
- il clock dell'ultimo evento inserito in fase di inizializzazione e' quello al quale si ha il "*primo arrivo*" dell'ultimo utente; tale clock viene pertanto detto "*Edge*" ed e' usato come soglia oltre la quale considerare valide le misure effettuate.

Inizializzazione

- 1) imposto la proprietà "clock" dell'oggetto "Simulazione" a 0;
- 2) inizializzo le code dei centri di servizio (tutte vuote – non si è ancora verificato alcun evento):

<i>UTENTI in ATTESA / JOB nelle CODE</i>			
<i>T</i>	<i>CPU</i>	<i>Disk1</i>	<i>Disk2</i>
0	0	0	0

- 3) popolo l'oggetto Calendario (lista degli eventi) coi primi 5 comandi che saranno generati dagli utenti:

<i>LISTA degli EVENTI</i>					
<i>ID</i>	<i>FK</i>	<i>SRC</i>	<i>DST</i>	<i>CLK</i>	<i>stato</i>
1	0	T	CPU	210	non processato
2	0	T	CPU	431	non processato
3	0	T	CPU	682	non processato
4	0	T	CPU	884	non processato
5	0	T	CPU	932	non processato

Esecuzione - 1

inizio processando il primo evento, quello di ID=1:

- 1) imposto la proprieta' "clock" a 210;
- 2) processo l'evento di ID=1:
 1. sulla SRC (T) un utente ha inviato il comando: aggiungo un job nella coda T;
 2. la DST (CPU) e' libera (non ha nulla in coda) la occupo (aggiungo un job nella coda):

<i>UTENTI in ATTESA / JOB nelle CODE</i>			
<i>T</i>	<i>CPU</i>	<i>Disk1</i>	<i>Disk2</i>
1	1	0	0

- 3) chiedo alla DST (CPU) quanto impieghera' ad eseguire il job (secondo la sua distribuzione di probabilita' – ipotizziamo 120) e quale sara' il DST (in base al suo routing – ipotizziamo Disk1);
- 4) creo l'evento di ID=6 corrispondente a quanto al punto 3), dove 330 = 210+120;
- 5) aggiorno lo stato dell'evento di ID=1 a "processato":

<i>LISTA degli EVENTI</i>					
<i>ID</i>	<i>FK</i>	<i>SRC</i>	<i>DST</i>	<i>CLK</i>	<i>stato</i>
1	0	T	CPU	210	processato
6	1	CPU	Disk1	330	non processato
2	0	T	CPU	431	non processato
3	0	T	CPU	682	non processato
4	0	T	CPU	884	non processato
5	0	T	CPU	932	non processato

Esecuzione - 2

proseguo processando il prossimo evento, quello di ID=6:

- 1) imposto la proprieta' "clock" a 330;
- 2) processo l'evento di ID=6:
 1. la SRC (CPU) ha terminato l'esecuzione, pertanto rimuovo un job dalla coda;
 2. la DST (Disk1) e' libera (non ha nulla in coda) la occupo (aggiungo un job nella coda):

<i>UTENTI in ATTESA / JOB nelle CODE</i>			
<i>T</i>	<i>CPU</i>	<i>Disk1</i>	<i>Disk2</i>
1	0	1	0

- 3) chiedo alla DST (Disk1) quanto impieghera' ad eseguire il job (secondo la sua distribuzione di probabilita' – ipotizziamo 320) e quale sara' il DST (in base al suo routing – ipotizziamo CPU);
- 4) creo l'evento di ID=7 corrispondente a quanto al punto 3), dove 650 = 330+320;
- 5) aggiorno lo stato dell'evento di ID=6 a "processato":

<i>LISTA degli EVENTI</i>					
<i>ID</i>	<i>FK</i>	<i>SRC</i>	<i>DST</i>	<i>CLK</i>	<i>stato</i>
1	0	T	CPU	210	processato
6	1	CPU	Disk1	330	processato
2	0	T	CPU	431	non processato
7	6	Disk1	CPU	650	non processato
3	0	T	CPU	682	non processato
4	0	T	CPU	884	non processato
5	0	T	CPU	932	non processato

Esecuzione - 3

proseguo processando il prossimo evento, quello di ID=2:

- 1) imposto la proprieta' "clock" a 431;
- 2) processo l'evento di ID=2:
 1. sulla SRC (T) un altro utente ha inviato il comando: aggiungo un job nella coda T;
 2. la DST (CPU) e' libera (non ha nulla in coda) la occupo (aggiungo un job nella coda):

<i>UTENTI in ATTESA / JOB nelle CODE</i>			
<i>T</i>	<i>CPU</i>	<i>Disk1</i>	<i>Disk2</i>
2	1	1	0

- 3) chiedo alla DST (CPU) quanto impieghera' ad eseguire il job (secondo la sua distribuzione di probabilita' – ipotizziamo 102) e quale sara' il DST (in base al suo routing – ipotizziamo T);
- 4) creo l'evento di ID=8 corrispondente a quanto al punto 3), dove 533 = 431+102;
- 5) aggiorno lo stato dell'evento di ID=2 a "processato":

<i>LISTA degli EVENTI</i>					
<i>ID</i>	<i>FK</i>	<i>SRC</i>	<i>DST</i>	<i>CLK</i>	<i>stato</i>
1	0	T	CPU	210	processato
6	1	CPU	Disk1	330	processato
2	0	T	CPU	431	processato
8	2	CPU	T	533	non processato
7	6	Disk1	CPU	650	non processato
3	0	T	CPU	682	non processato
4	0	T	CPU	884	non processato
5	0	T	CPU	932	non processato

Esecuzione - 4

proseguo processando il prossimo evento, quello di ID=8:

- 1) imposto la proprieta' "clock" a 533;
- 2) processo l'evento di ID=8:
 1. la SRC (CPU) ha terminato l'esecuzione, pertanto rimuovo un job dalla coda;
 2. sulla DST (T) un utente ha ricevuto la risposta: rimuovo un job dalla coda T;
 3. calcolo il tempo di risposta ripercorrendo 8->2: $533-431=102$ per il job 2;

<i>UTENTI in ATTESA / JOB nelle CODE</i>			
<i>T</i>	<i>CPU</i>	<i>Disk1</i>	<i>Disk2</i>
1	0	1	0

- 3) chiedo alla DST (T) quanto impieghera' ad inviare un altro comando (secondo la sua distribuzione di probabilita' – ipotizziamo 502) e quale sara' il DST (in base al suo routing e' sempre CPU);
- 4) creo l'evento di ID=9 corrispondente a quanto al punto 3), dove $1035 = 533+502$
- 5) aggiorno lo stato dell'evento di ID=8 a "processato":

<i>LISTA degli EVENTI</i>					
<i>ID</i>	<i>FK</i>	<i>SRC</i>	<i>DST</i>	<i>CLK</i>	<i>stato</i>
1	0	T	CPU	210	processato
6	1	CPU	Disk1	330	processato
2	0	T	CPU	431	processato
8	2	CPU	T	533	processato
7	6	Disk1	CPU	650	non processato
3	0	T	CPU	682	non processato
4	0	T	CPU	884	non processato
5	0	T	CPU	932	non processato
9	0	T	CPU	1035	non processato

NB: al clock 533, uno (il secondo, ma non e' rilevante) dei due utenti che avevano gia' inviato il comando, ha ricevuto la risposta e puo' inviare un nuovo comando. L'altro utente e' ancora in attesa. I restanti 3 utenti devono ancora generare il comando: lo faranno, indipendentemente dai primi due, ai tempi simulati 682, 884, 932.

Esecuzione - 5

proseguo processando il prossimo evento, quello di ID=7:

- 1) imposto la proprieta' "clock" a 650;
- 2) processo l'evento di ID=7:
 1. la SRC (Disk) ha terminato l'esecuzione, pertanto rimuovo un job dalla coda;
 2. la DST (CPU) e' libera (non ha nulla in coda) la occupo (aggiungo un job nella coda):

<i>UTENTI in ATTESA / JOB nelle CODE</i>			
<i>T</i>	<i>CPU</i>	<i>Disk1</i>	<i>Disk2</i>
1	1	0	0

- 3) chiedo alla DST (CPU) quanto impieghera' ad eseguire il job (secondo la sua distribuzione di probabilita' – ipotizziamo 80) e quale sara' il DST (in base al suo routing – ipotizziamo CPU);
- 4) creo l'evento di ID=10 corrispondente a quanto al punto 3), dove $730 = 650+80$;
- 5) aggiorno lo stato dell'evento di ID=7 a "processato":

<i>LISTA degli EVENTI</i>					
<i>ID</i>	<i>FK</i>	<i>SRC</i>	<i>DST</i>	<i>CLK</i>	<i>stato</i>
1	0	T	CPU	210	processato
6	1	CPU	Disk1	330	processato
2	0	T	CPU	431	processato
8	2	CPU	T	533	processato
7	6	Disk1	CPU	650	processato
3	0	T	CPU	682	non processato
10	7	CPU	CPU	730	non processato
4	0	T	CPU	884	non processato
5	0	T	CPU	932	non processato
9	0	T	CPU	1135	non processato

Esecuzione - 6

proseguo processando il prossimo evento, quello di ID=3:

- 1) imposto la proprieta' "clock" a 682;
- 2) processo l'evento di ID=3:
 1. sulla SRC (T) un altro utente ha inviato il comando: aggiungo un job nella coda T;
 2. la DST (CPU) e' gia' occupata (ha 1 job in coda): non la posso occupare, ma posso aggiungere un job in coda;
 3. essendoci solo 1 job in coda, quello che vado ad aggiungere verra' eseguito al termine di quello gia' in coda, ossia ad un CLK pari a quello del prossimo evento con SRC CPU (quello di ID=10), ossia al tempo 730 (disciplina di servizio FIFO);
 4. la differenza $730-682=48$ e' il tempo di attesa nella coda CPU;
 5. tenendo conto di quanto sopra, aggiungo un job nella coda di CPU;

<i>UTENTI in ATTESA / JOB nelle CODE</i>			
<i>T</i>	<i>CPU</i>	<i>Disk1</i>	<i>Disk2</i>
2	2	0	0

- 3) chiedo alla DST (CPU) quanto impieghera' ad eseguire il job (secondo la sua distribuzione di probabilita' – ipotizziamo 90) e quale sara' il DST (in base al suo routing – ipotizziamo CPU);
- 4) creo l'evento di ID=11 corrispondente a quanto ai punti 2) e 3), dove $820 = 730+90$;
- 5) aggiorno lo stato dell'evento di ID=3 a "processato":

<i>LISTA degli EVENTI</i>					
<i>ID</i>	<i>FK</i>	<i>SRC</i>	<i>DST</i>	<i>CLK</i>	<i>stato</i>
1	0	T	CPU	210	processato
6	1	CPU	Disk1	330	processato
2	0	T	CPU	431	processato
8	2	CPU	T	533	processato
7	6	Disk1	CPU	650	processato
3	0	T	CPU	682	processato
10	7	CPU	CPU	730	non processato
11	3	CPU	CPU	820	non processato
4	0	T	CPU	884	non processato
5	0	T	CPU	932	non processato
9	0	T	CPU	1135	non processato

NB: in questo "schema di simulazione" ho considerato la coda CPU ed il relativo tempo di attesa, ma un ragionamento analogo si puo' fare per la coda di Disk1 e il relativo tempo di attesa, di cui si richiede la stima.

Esecuzione - 7

proseguo processando il prossimo evento, quello di ID=10:

- 1) imposto la proprieta' "clock" a 730;
- 2) processo l'evento di ID=10:
 1. la SRC (CPU) ha terminato l'esecuzione, pertanto rimuovo un job dalla coda;
 2. la DST (CPU) e' gia' occupata (ha 2-1=1 job in coda): non la posso occupare, ma posso aggiungere un job in coda;
 3. essendoci solo 1 job in coda, quello che vado ad aggiungere verra' eseguito al termine di quello gia' in coda, ossia ad un CLK pari a quello del prossimo evento con SRC CPU (quello di ID=11), ossia al tempo 820 (disciplina di servizio FIFO);
 4. la differenza 820-730=90 e' il tempo di attesa nella coda CPU;
 5. tenendo conto di quanto sopra, aggiungo un job nella coda di CPU:

<i>UTENTI in ATTESA / JOB nelle CODE</i>			
<i>T</i>	<i>CPU</i>	<i>Disk1</i>	<i>Disk2</i>
2	2	0	0

- 3) chiedo alla DST (CPU) quanto impieghera' ad eseguire il job (secondo la sua distribuzione di probabilita' – ipotizziamo 130) e quale sara' il DST (in base al suo routing – ipotizziamo T);
- 4) creo l'evento di ID=12 corrispondente a quanto ai punti 2) e 3), dove 940 = 820+130;
- 5) aggiorno lo stato dell'evento di ID=11 a "processato":

<i>LISTA degli EVENTI</i>					
<i>ID</i>	<i>FK</i>	<i>SRC</i>	<i>DST</i>	<i>CLK</i>	<i>stato</i>
1	0	T	CPU	210	processato
6	1	CPU	Disk1	330	processato
2	0	T	CPU	431	processato
8	2	CPU	T	533	processato
7	6	Disk1	CPU	650	processato
3	0	T	CPU	682	processato
10	7	CPU	CPU	730	processato
11	3	CPU	CPU	820	non processato
4	0	T	CPU	884	non processato
5	0	T	CPU	932	non processato
12	10	CPU	T	940	non processato
9	0	T	CPU	1135	non processato

NB: vedi sopra per il tempo di attesa nella coda di Disk1

Esecuzione - 8

proseguo processando il prossimo evento, quello di ID=11:

- 1) imposto la proprieta' "clock" a 820;
- 2) processo l'evento di ID=11:
 1. la SRC (CPU) ha terminato l'esecuzione, pertanto rimuovo un job dalla coda;
 2. la DST (CPU) e' gia' occupata (ha 2-1=1 job in coda): non la posso occupare, ma posso aggiungere un job in coda;
 3. essendoci solo 1 job in coda, quello che vado ad aggiungere verra' eseguito al termine di quello gia' in coda, ossia ad un CLK pari a quello del prossimo evento con SRC CPU (il 12), ossia al tempo 940 (disciplina di servizio FIFO);
 4. la differenza $940-820=120$ e' il tempo di attesa nella coda CPU;
 5. tenendo conto di quanto sopra, aggiungo un job nella coda di CPU:

<i>UTENTI in ATTESA / JOB nelle CODE</i>			
<i>T</i>	<i>CPU</i>	<i>Disk1</i>	<i>Disk2</i>
2	2	0	0

- 3) chiedo alla DST (CPU) quanto impieghera' ad eseguire il job (secondo la sua distribuzione di probabilita' – ipotizziamo 200) e quale sara' il DST (in base al suo routing – ipotizziamo T);
- 4) creo l'evento di ID=13 corrispondente a quanto ai punti 2) e 3), dove $1040 = 940+200$;
- 5) aggiorno lo stato dell'evento di ID=11 a "processato":

<i>LISTA degli EVENTI</i>					
<i>ID</i>	<i>FK</i>	<i>SRC</i>	<i>DST</i>	<i>CLK</i>	<i>stato</i>
1	0	T	CPU	210	processato
6	1	CPU	Disk1	330	processato
2	0	T	CPU	431	processato
8	2	CPU	T	533	processato
7	6	Disk1	CPU	650	processato
3	0	T	CPU	682	processato
10	7	CPU	CPU	730	processato
11	3	CPU	CPU	820	processato
4	0	T	CPU	884	non processato
5	0	T	CPU	932	non processato
12	10	CPU	T	940	non processato
9	0	T	CPU	1135	non processato
13	11	CPU	T	1140	non processato

NB: vedi sopra per il tempo di attesa nella coda di Disk1

Esecuzione - 9

proseguo processando il prossimo evento, quello di ID=4:

- 1) imposto la proprieta' "clock" a 884;
- 2) processo l'evento di ID=4:
 1. sulla SRC (T) un altro utente ha inviato il comando: aggiungo un job nella coda T;
 2. la DST (CPU) e' gia' occupata (ha 2 job in coda): non la posso occupare, ma posso aggiungere un job in coda;
 3. essendoci ben 2 job in coda, quello che vado ad aggiungere verra' eseguito al termine dei due gia' in coda, ossia ad un CLK pari a quello del "bis-prossimo" evento con SRC CPU (il 13), ossia al tempo 1140 (disciplina di servizio FIFO);
 4. la differenza 1140-884=256 e' il tempo di attesa nella coda CPU;
 5. tenendo conto di quanto sopra, aggiungo un job nella coda di CPU;

<i>UTENTI in ATTESA / JOB nelle CODE</i>			
<i>T</i>	<i>CPU</i>	<i>Disk1</i>	<i>Disk2</i>
3	3	0	0

- 3) chiedo alla DST (CPU) quanto impieghera' ad eseguire il job (secondo la sua distribuzione di probabilita' – ipotizziamo 70) e quale sara' il DST (in base al suo routing – ipotizziamo T);
- 4) creo l'evento di ID=14 corrispondente a quanto ai punti 2) e 3), dove 1210 = 1140+70;
- 5) aggiorno lo stato dell'evento di ID=4 a "processato":

<i>LISTA degli EVENTI</i>					
<i>ID</i>	<i>FK</i>	<i>SRC</i>	<i>DST</i>	<i>CLK</i>	<i>stato</i>
1	0	T	CPU	210	processato
6	1	CPU	Disk1	330	processato
2	0	T	CPU	431	processato
8	2	CPU	T	533	processato
7	6	Disk1	CPU	650	processato
3	0	T	CPU	682	processato
10	7	CPU	CPU	730	processato
11	3	CPU	CPU	820	processato
4	0	T	CPU	884	processato
5	0	T	CPU	932	non processato
12	10	CPU	T	940	non processato
9	0	T	CPU	1135	non processato
13	11	CPU	T	1140	non processato
14	4	CPU	T	1210	non processato

NB: vedi sopra per il tempo di attesa nella coda di Disk1

Esecuzione - 10

proseguo processando il prossimo evento, quello di ID=5:

- 1) imposto la proprieta' "clock" a 932;
- 2) processo l'evento di ID=5:
 1. sulla SRC (T) un altro utente ha inviato il comando: aggiungo un job nella coda T;
 2. la DST (CPU) e' gia' occupata (ha 3 job in coda): non la posso occupare, ma posso aggiungere un job in coda;
 3. essendoci ben 3 job in coda, quello che vado ad aggiungere verra' eseguito al termine dei due gia' in coda, ossia ad un CLK pari a quello del "tris-prossimo" evento con SRC CPU (il 14), ossia al tempo 1210 (disciplina di servizio FIFO);
 4. la differenza $1210-932=278$ e' il tempo di attesa nella coda CPU;
 5. tenendo conto di quanto sopra, aggiungo un job nella coda di CPU;

<i>UTENTI in ATTESA / JOB nelle CODE</i>			
<i>T</i>	<i>CPU</i>	<i>Disk1</i>	<i>Disk2</i>
4	4	0	0

- 3) chiedo alla DST (CPU) quanto impieghera' ad eseguire il job (secondo la sua distribuzione di probabilita' – ipotizziamo 60) e quale sara' il DST (in base al suo routing – ipotizziamo Disk1);
- 4) creo l'evento di ID=15 corrispondente a quanto ai punti 2) e 3), dove $1270 = 1210+60$;
- 5) aggiorno lo stato dell'evento di ID=5 a "processato":

<i>LISTA degli EVENTI</i>					
<i>ID</i>	<i>FK</i>	<i>SRC</i>	<i>DST</i>	<i>CLK</i>	<i>stato</i>
1	0	T	CPU	210	processato
6	1	CPU	Disk1	330	processato
2	0	T	CPU	431	processato
8	2	CPU	T	533	processato
7	6	Disk1	CPU	650	processato
3	0	T	CPU	682	processato
10	7	CPU	CPU	730	processato
11	3	CPU	CPU	820	processato
4	0	T	CPU	884	processato
5	0	T	CPU	932	processato
12	10	CPU	T	940	non processato
9	0	T	CPU	1135	non processato
13	11	CPU	T	1140	non processato
14	4	CPU	T	1210	non processato
15	5	CPU	Disk1	1270	non processato

NB: vedi sopra per il tempo di attesa nella coda di Disk1

Esecuzione - 11

proseguo processando il prossimo evento, quello di ID=12:

- 1) imposto la proprieta' "clock" a 940;
- 2) processo l'evento di ID=12:
 1. la SRC (CPU) ha terminato l'esecuzione, pertanto rimuovo un job dalla coda;
 2. sulla DST (T) un utente ha ricevuto la risposta: rimuovo un job dalla coda T;
 3. calcolo il tempo di risposta ripercorrendo 12->10->7->6->1: $940-210=730$ per il job 1;

<i>UTENTI in ATTESA / JOB nelle CODE</i>			
<i>T</i>	<i>CPU</i>	<i>Disk1</i>	<i>Disk2</i>
3	3	0	0

- 3) chiedo alla DST (T) quanto impieghera' ad inviare un altro comando (secondo la sua distribuzione di probabilita' – ipotizziamo 340) e quale sara' il DST (in base al suo routing e' sempre CPU);
- 4) creo l'evento di ID=16 corrispondente a quanto al punto 3), dove $1280 = 940+340$;
- 5) aggiorno lo stato dell'evento di ID=4 a "processato":

<i>LISTA degli EVENTI</i>					
<i>ID</i>	<i>FK</i>	<i>SRC</i>	<i>DST</i>	<i>CLK</i>	<i>stato</i>
1	0	T	CPU	210	processato
6	1	CPU	Disk1	330	processato
2	0	T	CPU	431	processato
8	2	CPU	T	533	processato
7	6	Disk1	CPU	650	processato
3	0	T	CPU	682	processato
10	7	CPU	CPU	730	processato
11	3	CPU	CPU	820	processato
4	0	T	CPU	884	processato
5	0	T	CPU	932	processato
12	10	CPU	T	940	processato
9	0	T	CPU	1135	non processato
13	11	CPU	T	1140	non processato
14	4	CPU	T	1210	non processato
15	5	CPU	Disk1	1270	non processato
16	0	T	CPU	1280	non processato

NB: al clock 940, due (il primo ed il secondo, ma non e' rilevante) dei 5 utenti che avevano gia' inviato il comando, hanno ricevuto la risposta e hanno potuto inviare un nuovo comando. Gli altri 3 utenti sono ancora in attesa.

Meccanismo di calcolo

Assunto il modello sopra descritto per la simulazione (in particolare per il processamento degli eventi), il calcolo delle grandezze oggetto di misura risulta molto semplice, come già esemplificato in precedenza, per il tempo di risposta ai terminali e il tempo di attesa nella coda Cpu.

In particolare, si ricordano le assunzioni seguenti preliminari:

- la lista degli eventi (il “Calendario”) e' implementato in modo da mantenersi ordinato, quando viene inserito un nuovo evento, rispetto ai tempi simulati in cui avvengono gli eventi;
- un evento e' modellato come un oggetto dotato di:
 - un identificativo ID univoco;
 - una sorgente SRC, da cui l'evento e' originato;
 - una destinazione DST, su cui l'evento termina;
 - un tempo simulato CLK in cui l'evento avviene;
 - un riferimento FK all'eventuale evento “genitore”.

Stanti tali assunzioni, risulta immediato implementare il rilevamento degli eventi di interesse ed il calcolo delle relative misure:

- **TEMPI di ATTESA in CODA:**
 - ogni volta che un Centro di Servizio (di tipo Cpu/Disk, di cui si vogliono misurare i tempi di attesa in coda) e' DST di un nuovo evento A da inserire (che e' il risultato dal processamento dell'evento corrente C), e' facile recuperare il numero k di job nella sua coda; una volta ottenuto il numero k di job in coda, se:
 - $k=0$: il CdS e' immediatamente disponibile, per cui il tempo di attesa in coda e' 0 (per convenzione, la misura corrispondente non viene considerata);
 - $k>0$: basta recuperare il k-esimo evento futuro B di cui il CdS e' SRC: al verificarsi di tale evento, il CdS non avra' piu' job in coda e potra' processare quello relativo all'evento A; sottraendo al clock dell'evento B il clock corrente, si avra' il tempo di attesa in coda per l'evento da inserire nel calendario (la misura corrispondente verra' inserita nell'insieme di misure relativo);

NB: l'eventuale creazione di altri eventi con DST lo stesso CdS, non alterera' quanto sopra, in quanto tali eventi (per la disciplina FIFO) verranno inseriti sempre dopo l'evento A.

- **TEMPI di RISPOSTA:**

- ogni volta che un Terminale e' DST di un evento A, significa che un utente riceve la risposta al suo comando, per cui:
 - si risale, tramite il meccanismo degli “eventi genitore” all'evento G che ha generato tutta la catena (quello in cui il Terminale e' SRC e non ha quindi ulteriori genitori);
 - si recupera il clock di tale evento G e, sottraendolo al clock dell'evento A, si ottiene il tempo complessivo di risposta.

Esempio di Esecuzione

Si riporta un estratto del file “**users-05-run-01.txt**”, contenuto nell'allegato “**archivio risultati.zip**”, relativo al primo run eseguito su N=5 utenti.

Vengono commentate le stampe relative a inizializzazione, stato del calendario alla fine della simulazione e calcolo delle misure di interesse.

Inizializzazione

```
PRINT :: ----- CALENDARIO status at clock : 0 -----
PRINT :: Position  EventId  ParentId  Source   SrcQ  Destinat,  DstQ  Clock
PRINT :: 1          3       NoParent  Terminali 0    CPU        0    77075
PRINT :: 2          2       NoParent  Terminali 0    CPU        0    759336
PRINT :: 3          1       NoParent  Terminali 0    CPU        0    1050918
PRINT :: 4          5       NoParent  Terminali 0    CPU        0    1175889
PRINT :: 5          4       NoParent  Terminali 0    CPU        0    2175390
```

Avendo N=5 utenti ai terminali, si inizializza il calendario degli eventi creando 5 eventi Terminale-->Cpu.

Si noti come:

- gli eventi sono stati creati nell'ordine specificato da “*EventId*”, ma sono riportati nel calendario in posizioni differenti (“*Position*”), essendo il “*Clock*” a cui occorrono generato casualmente (coerentemente con la distribuzione di probabilita' assegnata ai Terminali);
- tali eventi abbiano “*NoParent*” come “genitore” (“*ParentId*”);
- l'ultimo utente arriva nel sistema per la prima volta al clock **2175390**, che viene impostato come “*Edge*” nel meccanismo di scarto delle prime misure, implementato al fine di ridurre la polarizzazione iniziale (“*bias*”).

Stato del calendario alla fine della simulazione

```

PRINT :: ----- CALENDARIO status at clock : 215603715 -----
PRINT :: Position  EventId  ParentId  Source  SrcQ  Destinat,  DstQ  Clock
PRINT :: 1         3         NoParent  Terminali 1   CPU       1     77075
PRINT :: 2         6         3         CPU       1   Disk1     0     113016
PRINT :: 3         7         6         Disk1    0   CPU       1     116859
PRINT :: 4         8         7         CPU      1   Disk1     0     117904
PRINT :: 5         9         8         Disk1    0   CPU       1     120255
PRINT :: 6        10         9         CPU      1   Disk1     0     123267
PRINT :: 7        11        10        Disk1    0   CPU       1     125890
PRINT :: 8        12        11        CPU      1   Terminali 1     146068
PRINT :: 9         2         NoParent  Terminali 1   CPU       1     759336
...
PRINT :: 161       163       162       CPU      1   Disk1     0     2352165
PRINT :: 162       165       164       CPU      1   Disk1     0     2355349
PRINT :: 163       166       163       Disk1    0   CPU       1     2359160
PRINT :: 164       168       166       CPU      1   Disk2     0     2361171
PRINT :: 165       167       165       Disk1    0   CPU       1     2363349
...
PRINT :: 21943     21947     21946     Disk1    0   CPU       1     215561901
PRINT :: 21947     21951     21950     Disk1    0   CPU       1     215603715
PRINT :: 21948     21952     21951     CPU      1   Disk1     0     215612041
PRINT :: 21949     21807     NoParent  Terminali 1   CPU       1     216305108
PRINT :: 21950     21507     NoParent  Terminali 1   CPU       1     216793165
PRINT :: 21951     21943     NoParent  Terminali 1   CPU       1     217589492
PRINT :: 21952     21686     NoParent  Terminali 1   CPU       1     218666523
    
```

Si noti quanto segue:

- SrcQ e DstQ riportano i valori delle code della SRC e della DST dell'evento, ma alla fine dell'esecuzione (una stampa del calendario al processamento di ogni evento, o anche solo al calcolo di una misura, avrebbe reso ingestibile il file di output - che in questo caso e' gia' di 1,8MB e 23.076 righe), per cui non sono da prendere in considerazione, nei ragionamenti che seguono;
- al clock 146068 avviene il primo evento Cpu-->Terminali e relativo calcolo della misura “Tempo di Risposta” (vedere di seguito);
- al clock 2355349 avviene un evento Cpu-->Disk1, il primo che trova Disk1 gia' occupata (con almeno 1 job in coda: il 166, che avverra' al clock 2359160) e di cui si calcola la prima misura “Tempo di attesa in coda” (vedere di seguito);
- la simulazione termina al clock 215603715, l'ultimo evento processato e' il 21951, restano N=5 eventi non processati, di cui:
 - 1 di tipo Cpu-->Disk, relativo a una richiesta utente che deve ancora ricevere risposta;
 - 4 di tipo Terminali-->Cpu, relativi a utenti che hanno gia' ricevuto le risposte ai loro precedenti comandi e ne invieranno dei nuovi ai clock indicati.

Calcolo misure “Tempo di attesa in Coda” a Disk1

PRINT :: Misure object 'Tempo di Attesa in Coda' on 'Disk1' holds 642 elements

PRINT :: Clock	Valore	Skipped
2355349	3811	No
2395725	1655	No
2435161	758	No
3826831	8592	No
3841649	3180	No
3849670	2510	No

Si noti quanto segue:

- al clock 2355349 viene effettuata la prima misura, pari a 3811;
- tale misurazione non viene scartata (“Skipped”=“No”) in quanto viene fatta al clock 2355349, successivo al primo arrivo dell'ultimo utente, che avviene al clock 2175390, precedentemente indicato come “Edge”;
- tale valore e' pari alla differenza tra il clock dell'evento corrente (il 165, vedere in precedenza la stampa del calendario) e il clock del primo evento Disk1-->Cpu (il 166, vedere in precedenza la stampa del calendario);
- cio' e' corretto in quanto, processando l'evento 165, di tipo Cpu-->Disk1, si deve generare un nuovo evento di tipo Disk1-->Cpu (a causa del routing di Disk1); tale evento sara' il 167, che dovra' aspettare che si verifichi l'evento 166, ovvero che il Disk1 si liberi del job che aveva in coda, prima che possa verificarsi a sua volta;
- tale meccanismo di misurazione rispetta sia la gestione FIFO delle code dei CdS Cpu/Disk, che il metodo di scarto delle misure “biased”.

Calcolo misure “Tempo di Risposta” ai Terminali

PRINT :: Misure object 'Tempo Complessivo di Risposta' on 'Terminali' holds 387 elements

PRINT :: Clock	Valore	Skipped
PRINT :: 146068	68993	Yes
PRINT :: 1057641	6723	Yes
PRINT :: 1088185	328849	Yes
PRINT :: 1441649	265760	Yes
PRINT :: 1579599	836	Yes
PRINT :: 1897806	91166	Yes
PRINT :: 2473601	298211	No
PRINT :: 2562620	3786	No
PRINT :: 2976490	724973	No

Si noti quanto segue:

- al clock **146068** viene effettuata la prima misura, pari a **68993**, in quanto (vedere in precedenza la stampa del calendario):
 - a tale clock si verifica l'evento **12**, di tipo Cpu-->Terminali;
 - tale evento, risalendo tramite il “ParentId” (12->11->10->9->8->7->6->3) e' stato generato dall'evento **3**, di tipo Terminali-->Cpu;
 - l'evento **3** avviene al clock **77075**;
 - la misura **68993** e' proprio pari alla differenza dei due clock **146068** e **77075**, dunque rappresenta il tempo (in clock) tra l'invio della richiesta e la ricezione della risposta;
- tale misura viene scartata (“Skipped”=”Yes”) in quanto avviene al clock **146068**, antecedente al primo arrivo dell'ultimo utente, che avviene al clock **2175390**, precedentemente indicato come “Edge”.