# Performance Modeling of Computer Systems and Networks

*Prof. Vittoria de Nitto Personè*

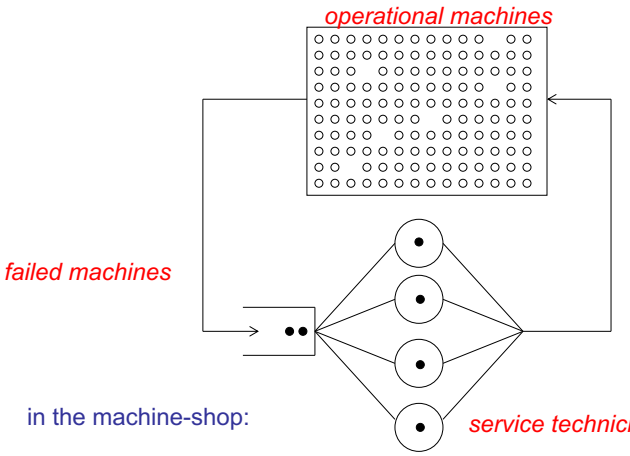
## Trace-driven simulation
Case study 1

Università degli studi di Roma Tor Vergata
Department of Civil Engineering and Computer Science Engineering



1

Discrete-event simulation
*conceptual model*




in the machine-shop:

- The "jobs" are the failed machines and the "servers" are the technies

- If there is only one service technician, the conceptual model includes a single-server queue



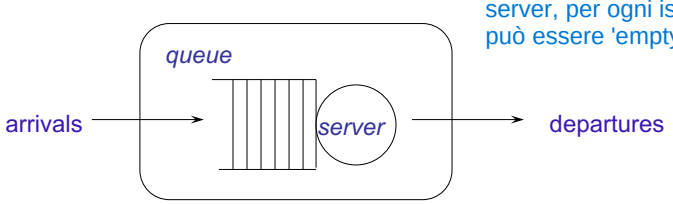
2

vedi p.22 di "Discrete"



## Single server queue

server, per ogni istante 't', può essere 'empty' o 'non empty'.




*Service node*

I job arrivano a punti random del tempo. Anche il tempo di servizio, dentro il server, è random (non tutti i job sono uguali). Se un job arriva, e la coda è vuota, verrà servito. Se job non arriva e coda vuota, server è idle.

- *def. 1* a *single server service node* consists of a *server* plus its *queue*



3

## terminology

synonymous

- queue/center/node
- job/user/request

| usual | | in the book |
|---|---|---|
| • waiting time | → | delay |
| • response/sojourn time | → | wait |

Il libro "Discrete" assume per default che la politica della coda sia FIFO, la coda sia infinita.
Inoltre, il server è conservativo, quindi se ci sono job da eseguire, non rimarrà mai idle.



4

Quali sono le grandezze in gioco? p.23 "Discrete"

delay/tempo attesa in coda/waiting time = quando macchina guasta deve attendere che le altre vengano riparate.

begins: momento in cui arrivo + tempo che attendo

tempo attesa = tempo risposta = quanto attendo in coda + tempo riparazione

quando riparto, departure, tempo completamento = quando arrivo + tempo che attendo per essere riparato

Discrete-event simulation
*specification model*

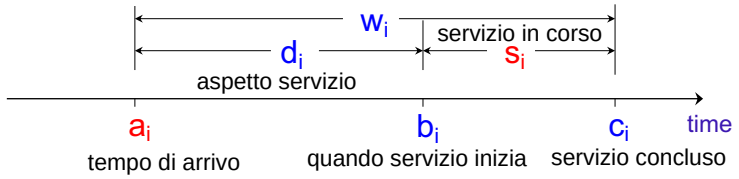
For a job *i*:

- The *arrival time* is $a_i$
- The *service time* is $s_i$

} input variables

- The *delay* in the queue is $d_i$ (delay, usually this is known as "waiting time")
- The time that service begins is $b_i = a_i + d_i$ (begin)
- The *wait* in the node is $w_i = d_i + s_i$ (wait, aka response time)
- The departure time is $c_i = a_i + w_i$ (completion)

} output variables






5

molto spesso si preferisce specificare il tempo di interrarrivo tra due nodi, piuttosto che specificare quando arrivano.

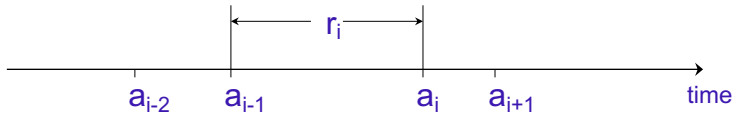
Discrete-event simulation
*specification model*

tempo di interarrivo

The *interarrival time* between jobs $i - 1$ and $i$ is

$$r_i = a_i - a_{i-1}$$

where, by definition, $a_0 = 0$




Assume only one arrival per time instant

$r_i > 0, \forall\ i$ ad ogni istante di tempo c'è un solo arrivo per volta. Non ci sono arrivi multipli nello stesso momento (bulk arrivals).

NO *bulk*

A volte ha senso considerare i bulk, altri no. Nel nostro caso potrei considerarli, perchè è possibile che si rompano due macchine nello stesso momento.



6

# *trace-driven simulation*

Ho i dati, come calcolo le misure di prestazione?

- The model is driven by external data:

  Given the arrival times $a_i$ and service times $s_i$, can the delay times $d_i$ be computed?

  ovvero, noti tempi servizio e tempi di arrivo, come faccio a sapere l'attesa in coda dei singoli job?
- For some queue disciplines, this question is difficult to answer

  Parliamo di ULTIMO COMPLETAMENTO, solo un job davanti a me.
- If the queue discipline is FIFO, $d_i$ is determined by when $a_i$ (the arrival) occurs relative to $c_{i-1}$ (the previous departure)

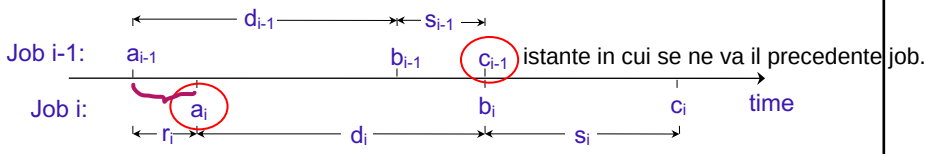
Nell'esempio, l'ordine di riparazione è FIFO, altre volte è più complesso.
Se l'arrivo è dopo il completamento del job precedente a me (non ho attesa), il delay è 0.





Case 1. The job arrives *before* the previous job completes

$a_i < c_{i-1}$



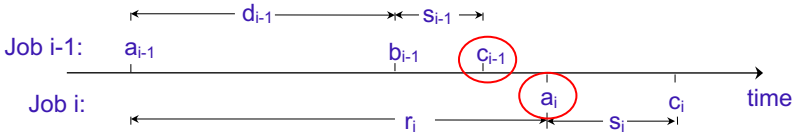

Se io arrivo e il job precedente è ancora presente, attendo un tempo pari a:
"tempo di completamento job precedente - tempo in cui sono arrivato io"

$d_i = c_{i-1} - a_i = [\, a(i-1) + w(i-1) \,] - a(i) =$
$a(i-1) - a(i) + w(i-1) =$
$r(i) + [\, s(i-1) + d(i-1) \,]$

r1 = tempo interrarrivo tra i due job.
Inoltre in job (i-1) c'è un tempo "b(i-1)", perchè anche il job precedente avrà atteso il suo predecessore.

Case 2. The job arrives *after* the completion of the previous job

$a_i \geq c_{i-1}$




In questo caso, è come se arrivassi quando la fila è vuota, non aspetto.

→ $d_i=0$



9

p.27 output statistics



# Output statistics

- The purpose of simulation is insight — gained by looking at statistics
- The importance of various statistics varies on perspective:
  - User perspective (job): wait time is most important
  - Manager perspective: utilization is critical
- Statistics are broken down into two categories
  - Job-averaged
  - Time-averaged

Facciamo ciò per ottenere indici, che possono esser più o meno importanti a seconda dell'attore coinvolto: ciò che vede l'utente è diverso da ciò che vede un manager.



10

Osservo 1000 job, di ciascuno mi calcolo tempo risposta (istante arrivo, istante che esce), voglio la media. Sommo e divido per 1000.
E' una statistica JOB average. Uguale per interarrivi e tempi servizi medi.
Se volessi calcolarmi popolazione media allora parlo di statistiche time-average.
esempio, popolazione in una scuola:
per 8 ore ci sono 100 persone (studenti + insegnanti), per 8 ore ci sono 20 persone (bidelli), per 8 ore non c'è nessuno.
allora avrò popolazione media: [100*8 + 20*8 + 0*8]/24, dove 24 sono le ore della giornata completa, e io voglio popolazione per ogni ora, ho deciso io così, potrei ragionare anche in giorni!

# Job-averaged statistics

• *Average interarrival time* $\bar{r} = \frac{1}{n}\sum_{i=1}^{n} r_i = \frac{a_n}{n}$

(a1 - a0) + (a2 - a1) +..... + [a(N) - a(N-1)], ma vedo che molti elementi si annullano. Restano aN - a0, ma a0 è 0. quindi resta aN, che divido per N.

è l'inverso, frequenza arrivo media. *Arrival rate* $\frac{1}{\bar{r}}$

• *Average service time* $\bar{s} = \frac{1}{n}\sum_{i=1}^{n} s_i$ qui non ho semplificazione, sommo e basta.

*Service rate* $\frac{1}{\bar{s}}$

Attenzione:
Service time e interarrival time sono espressi come secondi/job. I rate, essendo reciproci, sono jobs al secondo.

esempio: se ultimo job arriva ad a(N) = 320, e N = 10, allora r = a(N)/N = 32 secondi/Job
Allora arrival rate è $1/\bar{r} = 0.031$, ovvero mi arrivano 0.031 jobs al secondo.
Se, per esempio, service rate $1/\bar{s} = 0.029$ jobs al secondo, il server non riesce a processare tutti i job che arrivano (service rate < arrival rate) in media.





# Job-averaged statistics

• The *average delay* and *average wait* are defined as

$$\bar{d} = \frac{1}{n}\sum_{i=1}^{n} d_i \qquad \bar{w} = \frac{1}{n}\sum_{i=1}^{n} w_i$$

Recall $w_i = d_i + s_i \ \forall$ i, hence

tempo risposta medio $\bar{w} = \frac{1}{n}\sum_{i=1}^{n} w_i = \frac{1}{n}\sum_{i=1}^{n}(d_i + s_i) = \frac{1}{n}\sum_{i=1}^{n} d_i + \frac{1}{n}\sum_{i=1}^{n} s_i = \bar{d} + \bar{s}$ attesa media + servizio medio

Sufficient to compute any two of $\bar{w}, \bar{d}, \bar{s}$

note due, è facile calcolare la terza.
inoltre servizio medio è input, quindi basta delay da calcolare.



E' buona norma, in fase di controllo, non limitarsi a dire "ho due variabili, calcolo la terza in funzione di quelle che ho" bensì calcolare tutto in modo indipendente, e DOPO verificare se abbiamo rispettato il controllo di consistenza.

p.28 "Discrete"



# time-averaged statistics

For SSQ, need three additional functions

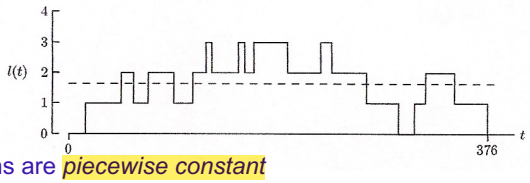
l(t) e q(t) sono >0, potenzialmente illimitati, a meno che non metta io dei paletti.

- l(t): number of jobs in the service node at time t service node = nel sistema
- q(t): number of jobs in the queue at time t
- x(t): number of jobs in service at time t

o opera, o non opera il server. (per questo 0 o 1).

By definition l(t)=q(t)+x(t) ∀ t

l(t) = 0, 1, 2, …
q(t) = 0, 1, 2, …
x(t) = 0, 1




The three functions are *piecewise constant*



13



Over the time interval (0, τ):

time-averaged number in the node: $\bar{l} = \frac{1}{\tau}\int_0^{\tau} l(t)dt$ alla fine sto calcolando l'area, dividendo per il tempo tau.

time-averaged number in the queue: $\bar{q} = \frac{1}{\tau}\int_0^{\tau} q(t)dt$

time-averaged number in service: $\bar{x} = \frac{1}{\tau}\int_0^{\tau} x(t)dt$

tempo è tau, x(t) è 0 o 1.
Se sempre pieno, integro su 1, ovvero trovo tau/tau = 1
Se sempre vuoto, integro su 0, 0/tau = 0

Def. *Utilization*
The proportion of time that the server is busy

Since l(t)=q(t)+x(t) ∀ t $\bar{l} = \bar{q} + \bar{x}$

Quindi x medio mi dice la proporzione a livello di tempo in cui il server è occupato.



14

Che relazione c'è tra queste statistiche?

How are job-averaged and time-average statistics related?

Little's Law (1961) ha un'applicabilità molto più alta di quella che vediamo ora.

If (a) queue discipline is FIFO,

(b) service node capacity is infinite, and (anche se non vale, posso applicare Little)

(c) server is idle both at the beginning and end of the observation interval ($t = 0$, $t = c_n$) istante iniziale e finale vuole server vuoto.

then

$$\int_0^{c_n} l(t)dt = \sum_{i=1}^{n} w_i$$ popolazione centro - tempo risposta

$$\int_0^{c_n} q(t)dt = \sum_{i=1}^{n} d_i$$ popolazione coda - tempo attesa

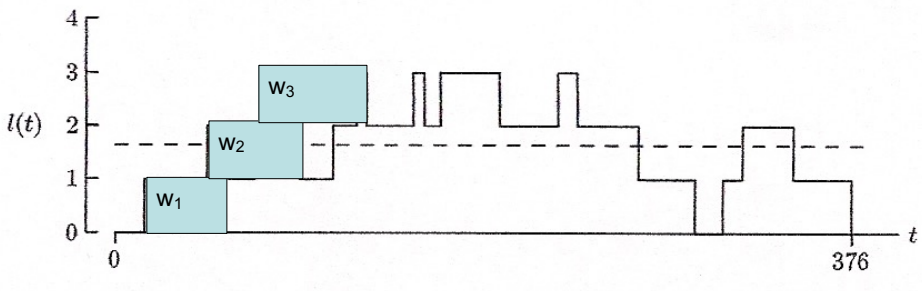
$$\int_0^{c_n} x(t)dt = \sum_{i=1}^{n} s_i$$ popolazione server - tempo servizio



15

A livello intuitivo, ha senso confrontare in QUESTI CASI DI STUDIO integrali e sommatorie, in quanto alla fine trattiamo funzioni stepwise. Infatti per dimostarlo basta scrivere l(t) come sommatoria di una funzione indicatrice del singolo job "i", da mettere dentro l'integrale. Questa funzione indicatrice vale 1 se job sta nel service node (nel sistema), 0 altrimenti. Porto fuori la sommatoria, mi rimane l'integrale su questa funzione indicatrice del job "i". Il job "i" sta nel sistema dal suo arrivo "a(i)" al suo completamento "c(i)", quindi c(i) - a(i). Ma questo è "w(i)", cioè il tempo di risposta.

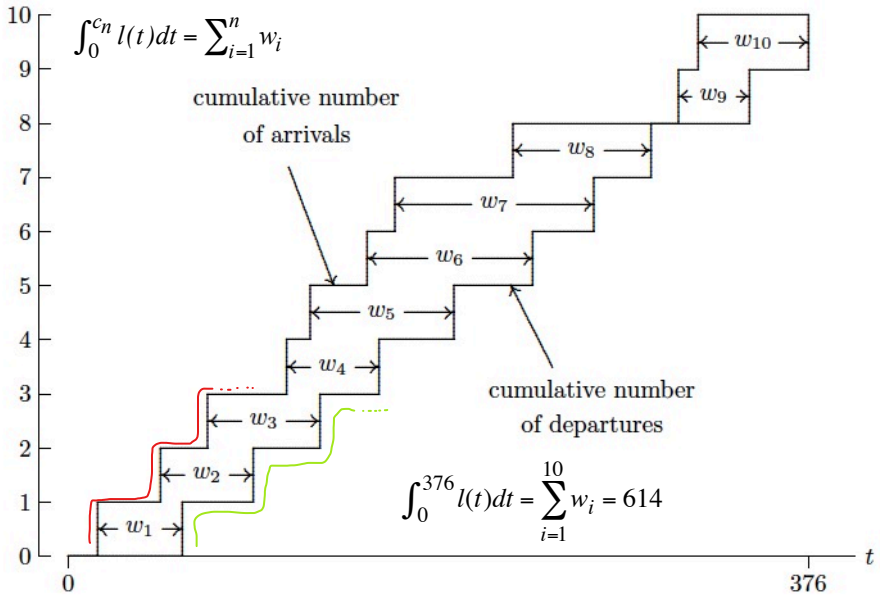
Parto da tempo 0, a tempo 376. W(i) è tempo risposta.






16

Questa è la cumulativa del numero di arrivi (bordo superiore),
e cumulativa del numero di partenze (bordo inferiore), Little dimostra che l'integrale
che l'integrale e la somma coincidono. Noi l'abbiamo visto da un punto di vista puramente grafico.






17

Discrete-event simulation
*Output statistics*

Using $\tau = c_n$ in $\bar{l} = \frac{1}{\tau}\int_0^{\tau} l(t)dt$ definizione di media, se metto tau = istante ultimo completamento c(n)

along with Little's Theorem, we have:

$$c_n \bar{l} = \int_0^{c_n} l(t)dt = \sum_{i=1}^{n} w_i = n\bar{w}$$

As a consequence: $\bar{l} = \frac{n}{c_n}\bar{w}$ forma che usiamo noi, relazione tra medie.

Same holds for: $\bar{q} = \frac{n}{c_n}\bar{d}$ $\bar{x} = \frac{n}{c_n}\bar{s}$

$\frac{n}{c_n}$ represents the *average system throughput* in $c_n$
Note that, for infinite queue, this corresponds to the average arrival rate

n° job/ tempo
throughput medio.
Mi dice quanti job serve nell'unità di tempo, per quel periodo di osservazione.



18

Ho sistema sottoposta a frequenza di arrivo, smaltisce con frequenza di servizio, il rapporto freq. arrivo/ freq. servizio.

## *Def.* Traffic intensity

The ratio of the arrival rate to the service rate

intensità di traffico può essere >1

$$\frac{1/\bar{r}}{1/\bar{s}} = \frac{\bar{s}}{\bar{r}} = \frac{\bar{s}}{a_n/n} = \left(\frac{c_n}{a_n}\right)\bar{x}$$

Ho utilizzazione <= 1, perchè vista come proporzione rispetto al tempo, quindi non può essere >1

$$\bar{x} = \frac{n}{c_n}\bar{s}$$ sfrutto Little

When $c_n/a_n$ is close to 1.0, the traffic intensity and utilization will be nearly equal

Per tempi estesi può effettivamente tendere a 1, allora in questi casi l'intensità di traffico e utilizzazione sono CIRCA la stessa cosa, ma generalmente non è detto!