Performance Modeling of Computer Systems and Networks

Prof. Vittoria de Nitto Personè

Priority scheduling

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

1

Service classes

- (Multimedia traffic)
- Quality of Service (QoS)
- Penalties

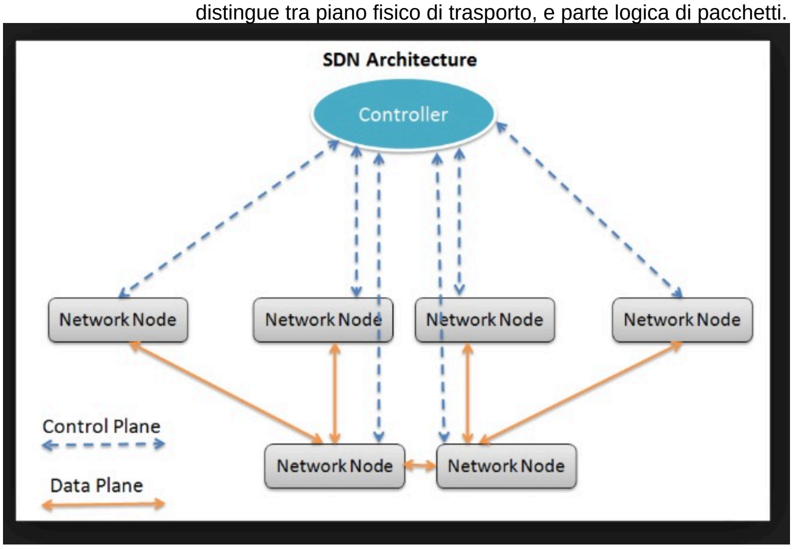
The proper scheduling policy can improve performance of a server tremendously.
It costs nothing to alter your scheduling policy (no money, no new hardware), so the performance gain comes for free.

2

SDN

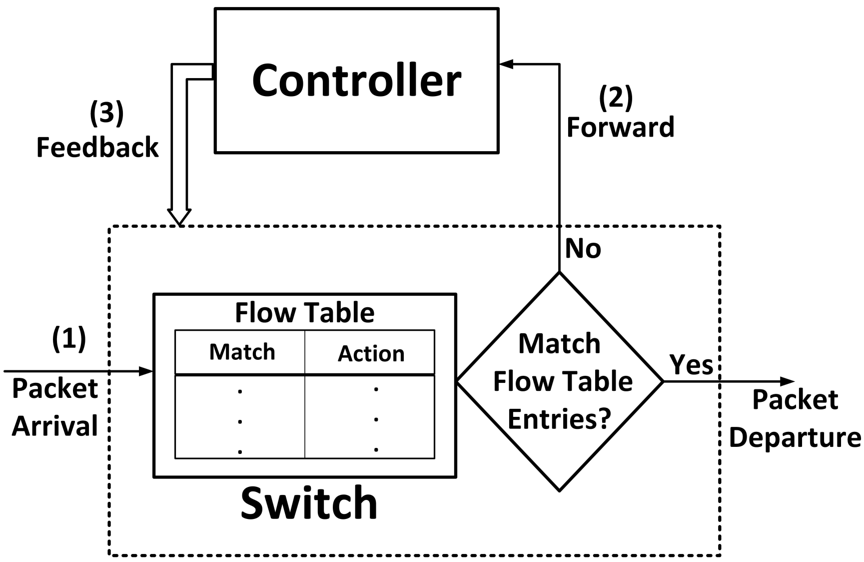
architettura per la realizzazione di reti di telecomunicazioni
il piano di controllo della rete e quello di trasporto dei dati sono separati logicamente

distingue tra piano fisico di trasporto, e parte logica di pacchetti.



3

Se da (1) arriva pacchetto non presente in tabella, lo mando al controllore (2), che fa operazioni di aggiornamento tabella, rimandandolo allo switch (3). Quando lo rimanda allo switch ha priorità maggiorata (poiché non essendo inviato ha accumulato ritardo).



4

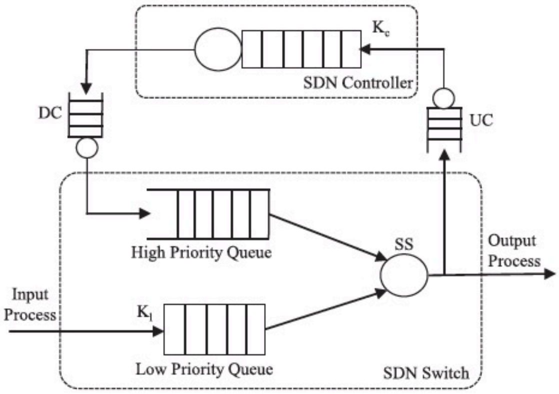
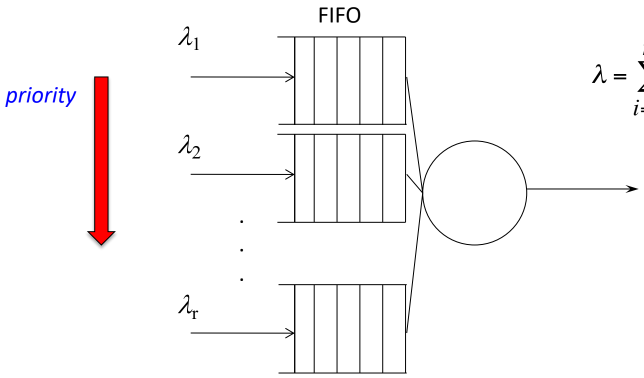


Fig. 1. The PQ-based SDN system architecture.

Miao W., Min G., Wu Y., Wang H. and Hu J., 2016. Performance Modelling and Analysis of Software-Defined Networking under Bursty Multimedia Traffic. *ACM Trans. Multimedia Comput. Commun. Appl.*, Vol. 12, No. 5s, Article 77.

5

Priority classes

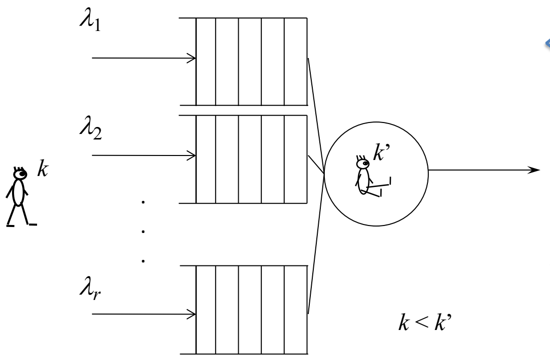


$$\lambda = \sum_{i=1}^{r} \lambda_i$$

Coda 1 priorità max, coda 'r' priorità minima. Il flusso totale lamda è la somma di questi 'r' flussi, e il servente qui è SINGOLO.

6

Priority classes



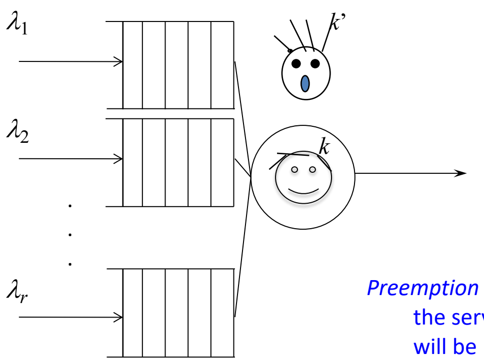
- *abstract*
  (non dipende da quanta è la size del job)
- *size based*
  (danno ordine in base alla quantità di lavoro chiesto)

- *preemptive*
  servizio interrompibile o meno!
- *not-preemptive*

In servizio c'è " k' ", questo vuol dire che tutte le code da 1 a " k' - 1" sono vuote, sennò toccava a loro in quell'istante. Se arriva 'k' più importante, e c'è PRELAZIONE, interrompo k' in favore di k.

Priority classes with preemption

$k < k'$

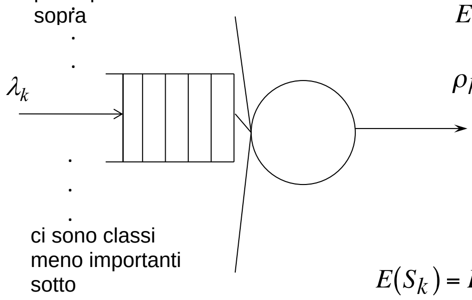


Preemption without loss:
the service of the interrupted job will be resumed from the interruption point

interrompo solo se posso salvare stato e riprenderlo da istante interruzione.
SI parla di "interruzione senza perdita".

Abstract priority without preemption

ci sono classi
più importanti
sopra



ci sono classi
meno importanti
sotto

S_k in "generale"

$$E(S_k) = \frac{1}{\mu_k} \qquad \sigma^2(S_k)$$

$$\rho_k = \lambda_k E(S_k) \qquad \rho = \sum_{i=1}^{r} \rho_i \text{ totale}$$

quanto arriva* quanto si ferma

abstract

tempo di servizio uguale
per tutti.

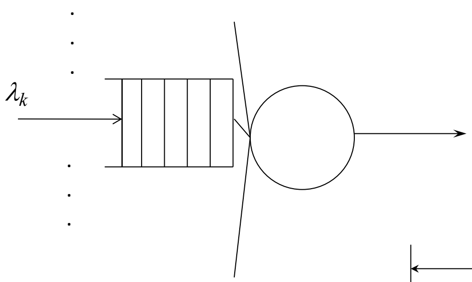
$$E(S_k) = E(S) = \frac{1}{\mu}, \quad \sigma^2(S_k) = \sigma^2(S), \forall k$$

$$\rho_k = \lambda_k E(S)$$

Nel caso size-based,
la coda-k ha un certo tempo di risposta
diverso da altre code (ordino in base alla size).

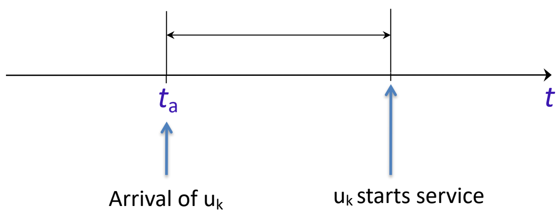
Abstract priority without preemption

local performance measures



tempo di attesa
specifico per una classe k?

$$E(T_{Q_k})?$$

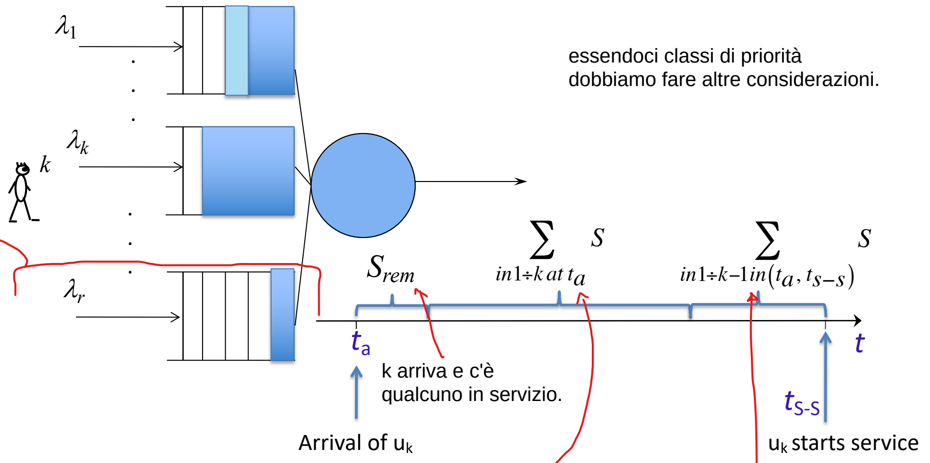


Abstract priority without preemption

local performance measures

essendoci classi di priorità
dobbiamo fare altre considerazioni.

per "k", quello che c'è
sotto non interessa.



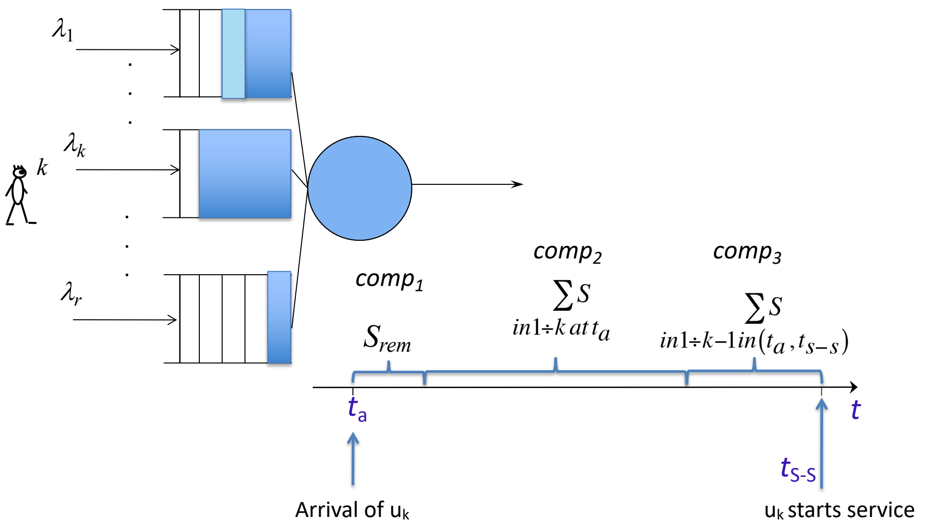
k arriva e c'è
qualcuno in servizio.

attendo TUTTI quelli da classe '1' a classe k
arrivati prima di me. Un job di classe k che arriva prima di
me (sempre classe k) ha precedenza, essendo FIFO.

attendo i job che sono più importanti (classi 1 a k-1)
di me, ma che sono arrivati dopo di me e prima
che io prenda servizio. Un job di classe k che arriva
dopo di me (io sono classe k) non ha precedenza,
essendo FIFO.

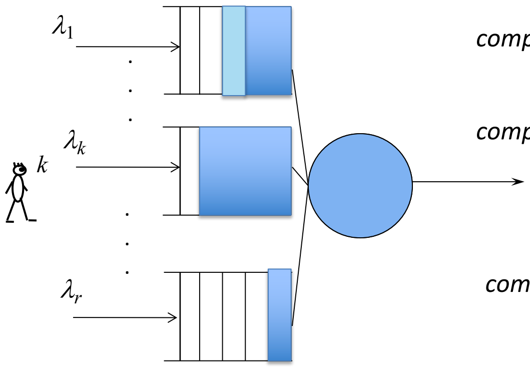
Abstract priority without preemption

local performance measures



Abstract priority without preemption

local performance measures



comp₁: $E\left(S_{rem}\right) = \frac{\lambda}{2} E\left(S^2\right)$

comp₂: proportional to the load of queues 1:k $= \dfrac{1}{1 - \sum_{i=1}^{k} \rho_i}$

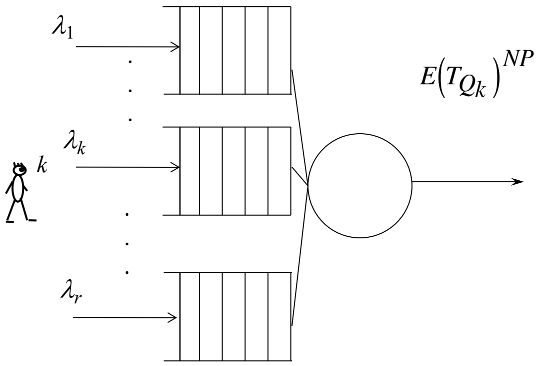
nel caso singolare era: $\dfrac{1}{1 - rho}$

comp₃: proportional to the load of queues 1:k-1 $= \dfrac{1}{1 - \sum_{i=1}^{k-1} \rho_i}$

13

Abstract priority without preemption

local performance measures



$$E\left(T_{Q_k}\right)^{NP\_priority} = \frac{\frac{\lambda}{2} E\left(S^2\right)}{\left(1 - \sum_{i=1}^{k} \rho_i\right)\left(1 - \sum_{i=1}^{k-1} \rho_i\right)}$$

classi di priorità senza prelazione!

$$E\left(T_{Q_k}\right) \le E\left(T_{Q_{k+1}}\right)$$

abbastanza banale.

14

Abstract priority without preemption

local performance measures

$$\boxed{E\left(T_{Q_k}\right) \le E\left(T_{Q_{k+1}}\right)}$$

$$\frac{\frac{\lambda}{2}E\left(S^2\right)}{\left(1-\sum_{i=1}^{k}\rho_i\right)\left(1-\sum_{i=1}^{k-1}\rho_i\right)} \le \frac{\frac{\lambda}{2}E\left(S^2\right)}{\left(1-\sum_{i=1}^{k}\rho_i\right)\left(1-\sum_{i=1}^{k+1}\rho_i\right)}$$

$$\frac{1}{\left(1-\sum_{i=1}^{k}\rho_i\right)\left(1-\sum_{i=1}^{k-1}\rho_i\right)} \le \frac{1}{\left(1-\sum_{i=1}^{k}\rho_i\right)\left(1-\sum_{i=1}^{k+1}\rho_i\right)}$$

prendo solo i denominatori:

$$\left(1-\sum_{i=1}^{k}\rho_i\right)\left(1-\sum_{i=1}^{k-1}\rho_i\right) \ge \left(1-\sum_{i=1}^{k}\rho_i\right)\left(1-\sum_{i=1}^{k+1}\rho_i\right)$$

15

Abstract priority without preemption

local performance measures

$$\cancel{\left(1-\sum_{i=1}^{k}\rho_i\right)}\left(1-\sum_{i=1}^{k-1}\rho_i\right) \ge \cancel{\left(1-\sum_{i=1}^{k}\rho_i\right)}\left(1-\sum_{i=1}^{k+1}\rho_i\right)$$

$$\cancel{1}-\sum_{i=1}^{k-1}\rho_i \ge \cancel{1}-\sum_{i=1}^{k+1}\rho_i$$

$$\sum_{i=1}^{k+1}\rho_i \ge \sum_{i=1}^{k-1}\rho_i \qquad \rho_i \ge 0, \forall i$$

$$\boxed{E\left(T_{Q_k}\right) \le E\left(T_{Q_{k+1}}\right)}$$

16

Abstract priority without preemption

local performance measures

Prestazioni locali, rispetto alla classe

indipendente dalla classe

$$E\left(T_{S_k}\right) = E\left(T_{Q_k}\right) + E(S) \qquad E\left(T_{S_k}\right) \le E\left(T_{S_{k+1}}\right)$$

usando Little:

$$E\left(N_{Q_k}\right) = \lambda_k E\left(T_{Q_k}\right)$$

$$E\left(N_{S_k}\right) = \lambda_k E\left(T_{S_k}\right) \qquad E\left(N_{S_k}\right) = E\left(N_{Q_k}\right) + \rho_k$$

rho specifico
della classe

Abstract priority without preemption

global performance measures

And the "global" performance? Devo fare una media pesata.

probabilità della classe k rispetto alla totale

$$E\left(T_Q\right)^{NP\_priority} = E\left(E\left(T_{Q_k}\right)\right) = \sum_{k=1}^{r} p_k E\left(T_{Q_k}\right) \quad =$$

$$p_k = \frac{\lambda_k}{\lambda}$$

% di traffico su una coda / tutto il traffico

and similarly for $E(T_S)^{NP\_priority}$

$$E(T_S)^{NP\_priority} = E\left(T_Q\right)^{NP\_priority} + E(S)$$

Abstract priority without preemption

probabilità

$\lambda_k = p_k \lambda$

rho: $\rho_k = \lambda_k E(S) = p_k \lambda E(S) = p_k \rho$

rho_k = probabilità di essere di quella classe * rho_totale

priority vs no-priority

How are the performance improved in respect of a simple abstract scheduling not-considering the priority classes? quante classi vanno meglio? quali peggio?

$$E(T_{Q_k})^{NP\_priority} = \frac{\frac{\lambda}{2} E(S^2)}{\left(1 - \sum_{i=1}^{k} \rho_i\right)\left(1 - \sum_{i=1}^{k-1} \rho_i\right)} \quad ? \quad E(T_Q)^{KP} = \frac{\frac{\lambda}{2} E(S^2)}{1 - \rho}$$

The highest priority class:

$$E(T_{Q_1})^{NP\_priority} = \frac{\frac{\lambda}{2} E(S^2)}{(1 - \rho_1)} \leq E(T_Q)^{KP}$$

1

prima classe va sicuramente meglio (vede solo se stessa), meglio rispetto a vedere tutti mischiati.
rho_1 è più grande di rho???

Analytical models
priority scheduling

priority vs no-priority

How are the performance improved in respect of a simple abstract scheduling not-considering the priority classes?

The lowest priority class:

$$E\left(T_{Q_r}\right)^{NP\_priority} = \frac{\frac{\lambda}{2}E\left(S^2\right)}{(1-\rho)\left(1-\sum_{i=1}^{r-1}\rho_i\right)} \geq E\left(T_Q\right)^{KP}$$

And what about the "global" performance?

$$E\left(T_Q\right)^{NP\_priority} = E\left(T_Q\right)^{KP}$$

se vado a fare somme pesate, non ho vantaggi. Le classi più basse annullano i vantaggi delle classi più alte.

$$\Downarrow$$

$$E(T_S)^{NP\_priority} = E(T_S)^{KP}$$

21

Se pensiamo a KP, scheduling astratti. Ma anche NP_priority ha uno scheduling astratto, e quindi globalmente non è nient'altro che una KP.

Analytical models
priority scheduling

priority vs no-priority

$$E\left(T_Q\right)^{NP\_priority} = E\left(E\left(T_{Q_k}\right)\right) = \sum_{k=1}^{r} p_k E\left(T_{Q_k}\right) = E\left(T_Q\right)^{KP}$$

$r=2$

$$E\left(T_Q\right) = p_1 E\left(T_{Q_1}\right) + p_2 E\left(T_{Q_2}\right) = p_1 \frac{\frac{\lambda}{2}E\left(S^2\right)}{(1-\rho_1)} + p_2 \frac{\frac{\lambda}{2}E\left(S^2\right)}{(1-\rho)(1-\rho_1)}$$

$$= \frac{\lambda}{2}E\left(S^2\right)\left[\frac{p_1}{(1-\rho_1)} + \frac{p_2}{(1-\rho)(1-\rho_1)}\right] = \frac{\lambda}{2}E\left(S^2\right)\frac{p_1(1-\rho)+p_2}{(1-\rho)(1-\rho_1)} = \frac{\frac{\lambda}{2}E\left(S^2\right)}{1-\rho}$$

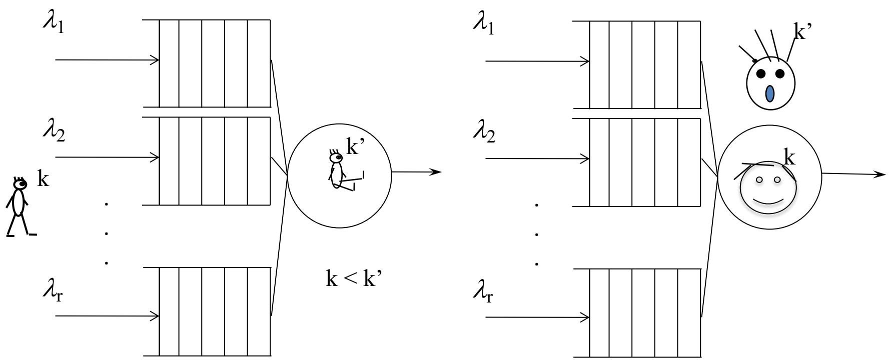
p1 - p1*ρ + p2

$\downarrow$

p1 + ρ_1 + p2 = 1 + ρ_1 , si annulla con denominatore.

22

Preemptive Priority

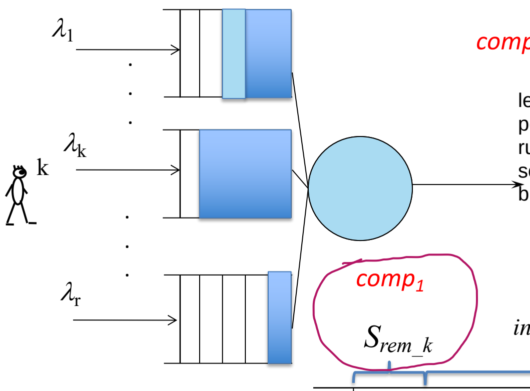


Preemption without loss:
the service of the interrupted job will be resumed from the interruption point

23

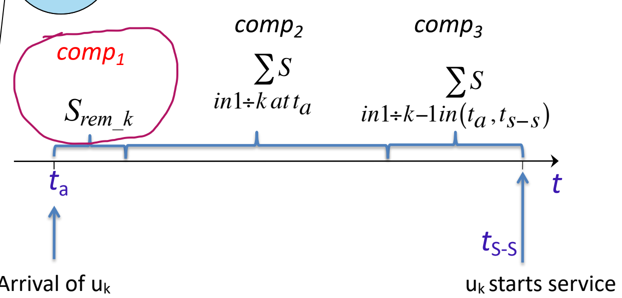
Preemptive Priority

local performance measures



comp₁: $E\left(S_{rem_k}\right) = \frac{1}{2}\sum_{i=1}^{k}\lambda_i E\left(S^2\right)$

le classi da 1 a k,
perchè se arrivano e sono più importanti, gli rubano il posto. Prima non era una sommatoria, perchè nessuno poteva bloccarmi.

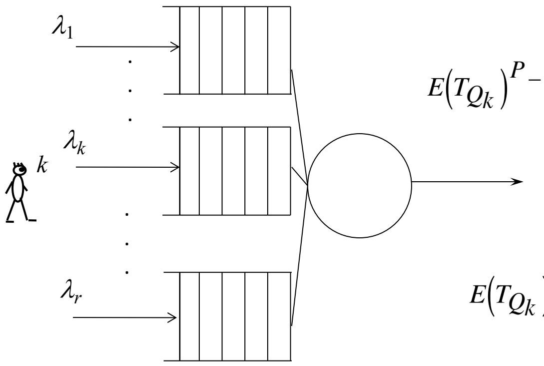


24

Analytical models
priority scheduling

Preemptive Priority

quando non c'è prelazione, c'è tutto lambda.



$$E(T_{Qk})^{P\_priority} = \frac{\frac{1}{2}\sum_{i=1}^{k}\lambda_i E(S^2)}{\left(1-\sum_{i=1}^{k}\rho_i\right)\left(1-\sum_{i=1}^{k-1}\rho_i\right)}$$

$$E(T_{Qk})^{P\_priority} \le E(T_{Qk+1})^{P\_priority}$$

$$E(T_{Qk})^{P\_priority} \le E(T_{Qk})^{NP\_priority}$$

$$E(T_Q)^{P\_priority} \le E(T_Q)^{NP\_priority} = E(T_Q)^{KP}$$

per P_priority ho al numeratore una sommatoria di lambda da 1 a k.

per NP_priority ho tutto lambda, quindi è sempre maggiore del primo caso (sono uguali se prendo l'ultima classe di importanza).

prof. Vittoria de Nitto Personè 25

25

Ho guadagnato qualcosa per alcune classi, perchè il tempo con prelazione globale è minore uguale rispetto alla variante senza prelazione. Ho finalmente guadagnato sulla KP. Nel modello SENZA Prelazione, se c'è in servizio 'k', e arriva un 'k-1', deve aspettare. Qui, con prelazione, sostituisco subito, quindi posso dire ancora meglio che NON VEDO CLASSI DI PRIORITÀ INFERIORI. Per questo c'è guadagno.

Analytical models
priority scheduling

Preemptive Priority

stai sereno che lo vedo

$$\frac{\frac{1}{2}\sum_{i=1}^{k}\lambda_i E(S^2)}{\left(1-\sum_{i=1}^{k}\rho_i\right)\left(1-\sum_{i=1}^{k-1}\rho_i\right)} \le \frac{\frac{1}{2}\sum_{i=1}^{k+1}\lambda_i E(S^2)}{\left(1-\sum_{i=1}^{k}\rho_i\right)\left(1-\sum_{i=1}^{k+1}\rho_i\right)}$$

$$\frac{\sum_{i=1}^{k}\lambda_i}{\left(1-\sum_{i=1}^{k}\rho_i\right)\left(1-\sum_{i=1}^{k-1}\rho_i\right)} \le \frac{\sum_{i=1}^{k+1}\lambda_i}{\left(1-\sum_{i=1}^{k}\rho_i\right)\left(1-\sum_{i=1}^{k+1}\rho_i\right)}$$

$$\frac{\sum_{i=1}^{k}\lambda_i}{\left(1-\sum_{i=1}^{k}\rho_i\right)\left(1-\sum_{i=1}^{k-1}\rho_i\right)} \le \frac{\sum_{i=1}^{k+1}\lambda_i}{\left(1-\sum_{i=1}^{k}\rho_i\right)\left(1-\sum_{i=1}^{k-1}\rho_i\right)}$$

$$\left(1-\sum_{i=1}^{k}\rho_i\right)\left(1-\sum_{i=1}^{k-1}\rho_i\right) \ge \left(1-\sum_{i=1}^{k}\rho_i\right)\left(1-\sum_{i=1}^{k+1}\rho_i\right)$$

prof. Vittoria de Nitto Personè 26

26

Preemptive Priority

$$E(T_Q)^{X\_priority} = E\left(E(T_{Qk})\right) = \sum_{k=1}^{r} p_k E(T_{Qk})$$
$$= p_1 E(T_{Q1}) + p_2 E(T_{Q2}) + \ldots + p_r E(T_{Qr})$$

$$E(T_Q)^{NP\_priority} = p_1 E(T_{Q1}) + p_2 E(T_{Q2}) + \ldots + p_r E(T_{Qr})$$

= VI = VI = VI

$$E(T_Q)^{P\_priority} = p_1 E(T_{Q1}) + p_2 E(T_{Q2}) + \ldots + p_r E(T_{Qr})$$

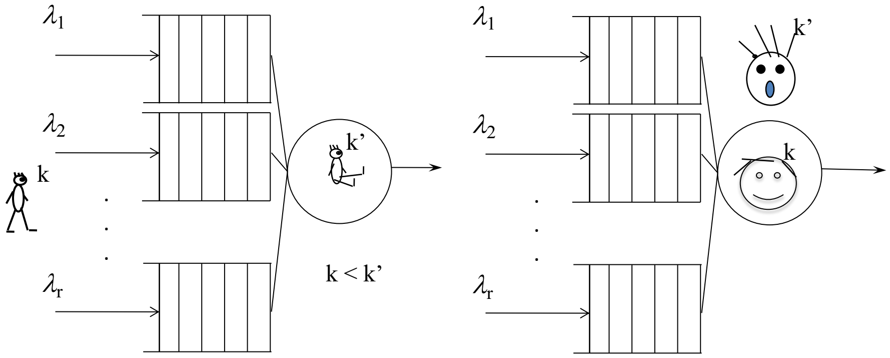
$$E(T_Q)^{P\_priority} \le E(T_Q)^{NP\_priority} = E(T_Q)^{KP}$$

27

Se parliamo di servizio, un job che viene buttato fuori e poi ritorna, perde più tempo. La sua classe è sempre k, quindi perde molto tempo.
Tempo attesa: tempo arrivo - tempo presa servizio. Adesso specifichiamo "tempo presa servizio" con "tempo presa servizio la prima volta".

Preemptive Priority

Due to preemption, the service time of a job of class k may increase for the arrivals of higher priority classes



28

Preemptive Priority

Due to preemption, the service time of a job of class k may increase for the arrivals of higher priority classes

Essere buttato fuori dal servizio perchè arriva un job prioritario, rallenta il mio tempo di servizio.
Lo posso immaginare come E[s] + attese per le interruzioni

Virtual service time $E(S_{virt\_k})$

$$E\left(S_{virt\_k}\right) = \frac{E(S)}{1 - \sum_{i=1}^{k-1} \rho_i}$$

viene allungato di un fattore proporzionale a tutti i rho delle classi che mi possono interrompere. La sommatoria arriva a "k-1", una classe "k" non può infatti essere interrotta da sè stessa.

$$E\left(T_{S_k}\right)^{P\_priority} = E\left(T_{Q_k}\right)^{P\_priority} + E\left(S_{virt\_k}\right)$$

$$\wedge| \qquad \qquad |\vee \qquad ?$$

$$E\left(T_{S_k}\right)^{NP\_priority} = E\left(T_{Q_k}\right)^{NP\_priority} + E(S)$$

29 Con priority miglioro l'attesa in coda, ma peggioro il tempo di servizio.

Non posso dire a priori chi sia più grande, c'è chi va meglio e chi peggio.
Per la classe 1, E(S_virtu_k) = E[S], perchè classe 1 non può essere buttata fuori.
A parte classe 1, le altre non posso dire nulla: quante classi ci sono? quanto sono grandi i flussi?

Preemptive Priority

global response time

And what about the "global" performance?

$$\boxed{E\left(T_{Q_k}\right)^{P\_priority} + E\left(S_{virt\_k}\right)}$$

$$E(T_S)^{P\_priority} = E\left(E\left(T_{S_k}\right)\right) = \sum_{k=1}^{r} p_k E\left(T_{S_k}\right)$$

$$E(T_S)^{P\_priority} = \sum_{k=1}^{r} p_k \left[E\left(T_{Q_k}\right) + E\left(S_{virt\_k}\right)\right]$$

$$= \sum_{k=1}^{r} p_k E\left(T_{Q_k}\right) + \sum_{k=1}^{r} p_k E\left(S_{virt\_k}\right)$$

$$= E\left(T_Q\right)^{P\_priority} + \sum_{k=1}^{r} p_k \underbrace{E\left(S_{virt\_k}\right)}$$

non è più E[S], lo è solo per la classe 1.

30

preemption vs no-preemption

$$E(T_S)^{P\_priority} = E(T_Q)^{P\_priority} + \sum_{k=1}^{r} p_k E(S_{virt\_k})$$

$$\wedge| \qquad\qquad |\vee$$

$$E(T_S)^{NP\_priority} = E(T_Q)^{NP\_priority} + E(S) = E(T_S)^{KP}$$

?

anche in termini globali non posso confrontarli, perchè per i singoli "pezzi" posso dire chi è più piccolo, ma mettendoli insieme non so dire se ho un guadagno o meno.

In general

$$E(T_S)^{P\_priority} \quad ? \quad E(T_S)^{KP}$$

For exponential service time

$$E(T_S)^{P\_priority} = E(T_S)^{KP}$$

!!!

La memoryless annulla tutto il guadagno, ovvero ciò che guadagno in attesa è compensato da ciò che perdo nel servizio.

31

preemption vs no-preemption

$r = 2$

tutto viene dall'ESPONENZIALITA'.

$$E(T_S)^{P\_priority} = p_1 E(T_{S_1}) + p_2 E(T_{S_2})$$

tempo virtuale

$$= p_1\left[\frac{\frac{\lambda_1}{2}E(S^2)}{(1-\rho_1)} + E(S)\right] + p_2\left[\frac{\frac{\lambda}{2}E(S^2)}{(1-\rho)(1-\rho_1)} + \frac{E(S)}{1-\rho_1}\right]$$

prima classe, non ritardabile

seconda classe

from the expo assumption

aspetta tutti

$$= p_1\left[\frac{\rho_1 E(S)}{(1-\rho_1)} + E(S)\right] + p_2\left[\frac{\rho E(S)}{(1-\rho)(1-\rho_1)} + \frac{E(S)}{1-\rho_1}\right]$$

$$= E(S)\left\{p_1\left[\frac{\rho_1 + 1 - \rho_1}{(1-\rho_1)}\right] + p_2\left[\frac{\rho + (1-\rho)}{(1-\rho)(1-\rho_1)}\right]\right\}$$

32

preemption vs no-preemption

$r = 2$

$$E(T_S)^{P\_priority} = p_1 E(T_{S_1}) + p_2 E(T_{S_2})$$
$$= E(S)\left[\frac{p_1}{(1-\rho_1)} + \frac{p_2}{(1-\rho)(1-\rho_1)}\right]$$
$$= E(S)\frac{p_1(1-\rho)+p_2}{(1-\rho)(1-\rho_1)} = \frac{E(S)}{1-\rho} = E(T_S)^{KP}$$

La proprietà Memoryless semplifica moltissimo i fenomeni.

Priorità astratte: criteri indipendenti da quanto chiedono di servizio.

Tempi di attesa in coda:
Se c'è prelazione, ho vantaggi locali e globali su E[Tq] rispetto alla NonPreemptive e rispetto a KP.
Senza prelazione, localmente ho vantaggi tra due classi, ma globalmente ciò si traduce in vantaggio per una classe e svantaggio per un'altra, ottenendo stesse prestazioni della KP.

33

Tempi di risposta:
Sia preemptive che Non, sia locale che non, non posso mai dire con certezza se preemptive vada meglio, anche se è meglio rispetto E[Tq] perdo questo vantaggio nel calcolo del tempo di servizio.
(sia localmente che globalmente).

Con tempo esponenziale torno globalmente al caso della KP.

20/04/2023 osservazione: perchè l'occupazione delle classi si vede usando i "rho"?
Il tempo di servizio rimanente È GENERALMENTE E[S_rim] = lambda/2 * E[S^2].
Abbiamo visto, per la classe k, che: E[S_rim] * 1/[1 - sum from i = 1 to k di rho(i)] * 1/[1 - sum from i =1 to k-1 di rho(i)]
Quante classi ci guadagnano? cioè quante volte questa media del tempo di attesa è < E[Srim]/(1-rho), ovvero caso coda unica KP. Stiamo lavorando nel caso senza prelazione. Tutto si riduce a vedere il denominatore, ovvero:
[1 - sum from i = 1 to k di rho(i)] * [1 - sum from i =1 to k-1 di rho(i) > (1 - rho) = 1 - sum from i = i to r di rho(i) (somma di tutte le r classi). Divido questa ultima sommatoria in: sum from i = 1 to k di rho(i) + sum from i = k +1 to r di rho(i), ho semplicemente spezzato.
Svolgendo il prodotto delle componenti a sinistra otteniamo:
~~1~~ - sum from i = 1 to k-1 di rho(i) - sum ~~from~~ i =1 to k di rho(i) + [sum from i = 1 to k di rho(i) * sum from i = 1 to k-1 di rho(i)]
> ~~1~~ - sum from i ~~=~~ 1 to k di rho(i) - sum from i = k+1 to r di rho(i).

Porto viola a sinistra, e tutto il resto a destra: sto confrontando a sinistra le ultime k+1 classi con quelle più importanti.
sum from i = k+1 to r di rho(i) > sum from i = 1 to k-1 di rho(i) * [1 - sum from i = 1 to k di rho(i)]   cioè:
se livello occupazione classi più basse è > livello occupazioni classi più alte, allora le prestazioni delle prime k classi sono migliori. (1 - sum from i = 1 to k di rho(i)) è <1, quindi stiamo parlando in "percentuale", perchè questo valore è moltiplicato per le prime k-1 classi.
Tanto più sono le "k classi" considerate, tanto più quelle sotto vengono svantaggiate.