Performance Modeling of Computer Systems and Networks

Prof. Vittoria de Nitto Personè

Multiserver and Priority scheduling

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

1

Assumptions:

- Arrival rate 1 j/s random
- Average demand Z=4x10^5 oxat, expo, do not know size (astratto)
  Z = quanto job chiede, op/job

Possible configurations:

- 1 server of capacity C=10^6 oxat/s — capacità server, non è v.a.
- Dual-core of C/2 each one — dual core equivalente, ciascun proc ha capacità dimezzata.

QoS requirements:

- Average waiting $T_Q < 0.15$ s
- For at least 35% of arrivals average response time $T_S < 0.5$ s
  la percentuale viene fornita dal testo

Def.

$E(S) = Z/C = 0.4$ s   operazioni richiesta/operazioni server nell'unità di tempo

Z e C sono indipendenti, poichè C è una caratteristica fisica dell'hardware, costante; Z è una variabile, è quanto chiede un singolo job (varia da job a job), e mediamente è Z.

2

QoS requirements:

- Average waiting $T_Q < 0.15$ s

Analytical models
priority scheduling

$\lambda = 1$ j/s, $E(S) = 0.4$ s $\Rightarrow$ $\rho = 0.4$

- 1 server of capacity C=10^6 oxerat/s

KP-M/M/1

$E[T_q] = \frac{\rho \cdot E[S]}{1-\rho}$ $E(T_Q) = 0.26$ s $E(T_Q)^{\text{Abstract-P}} = 0.2243$ s

- Dual-core of C/2 each one

$$E(S_i) = \frac{Z}{\frac{C}{2}} = 2\frac{Z}{C} = 2E(S) = 0.8\ s$$

$$E(T_Q)_{Erlang} = \frac{P_Q E(S)}{1-\rho} = 0.15238 \text{ s}$$

PARTE SINGLE CORE

prof. Vittoria de Nitto Personè 3

3

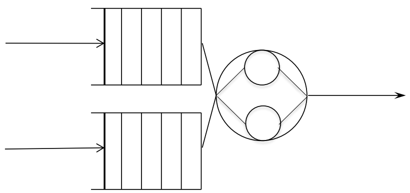
QoS requirements:

- Average waiting $T_Q < 0.15$ s

Analytical models
priority scheduling

$\lambda = 1$ j/s, $E(S) = 0.4$ s $\Rightarrow$ $\rho = 0.4$

- Dual-core of C/2 each one



prof. Vittoria de Nitto Personè 4

4

[Spiegato bene sul quaderno degli esercizi] Tento con abstract priority Preemptive (l'unica che porta miglioramenti generali e non solo locali).
Setto la probabilità p1 = 0.35 (perchè lo chiede il secondo requisito, che può essere rispettato se rispetto almeno il primo):

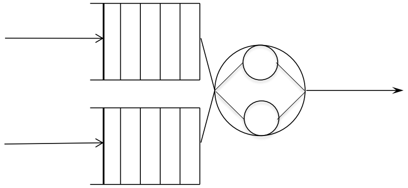
E[Tq1] = $\frac{p_1 \cdot \rho \cdot E[S]}{1 - \rho \cdot p_1}$ = 0.065, E[Tq2] = $\frac{\rho E[S]}{(1-\rho_1)(1-\rho)}$ = 0.31, pesandole con le rispettive percentuali trovo

E[Tq]= 0.35 * E[Tq1] + 0.65 * E[Tq2] = 0.2243 s, che non rispetta. Se cambiassi percentuali?
Trovando la probabilità p1 tale che E[Tq1] = 0.15 trovo p1 = 0.68, ma applicandolo a E[Tq1] ed E[Tq2] non rispetto il QoS.

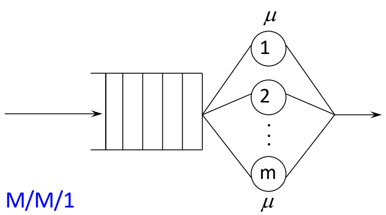
PARTE DUAL CORE

Essendo in Erlang C, bisogna calcolare p(0) = probabilità che sistema sia vuoto,
poi Pq = probabilità che sistema sia pieno, e poi applicare E[Tq_erlang] in figura.
Si usa rho = 0.4 e E[S] = 0.4, perchè l'utilizzazione rho è ENTRATA MAX/USCITA MAX = lambda/2*u_ridotto (u_ridotto è il mu del server dimezzato),
cioè al massimo lavorano entrambi i server, e in questo caso il tempo che si libera uno dei due server è E[S].

Multiserver with priority classes



The *Erlang* formula



M/M/m

$$E(T_Q)_{Erlang} = \frac{P_Q E(S)}{1-\rho}$$

M/M/1

$$E(T_Q)_{KP} = \frac{\rho E(S)}{1-\rho} = \frac{E(S_{rem})}{1-\rho}$$

$$E(S) = \frac{E(S_i)}{m}$$

osservo similitudini

Multiserver with priority classes

$$E(T_Q) = p_1 \frac{\rho_1 E(S)}{(1-\rho_1)} + p_2 \frac{\rho E(S)}{(1-\rho)(1-\rho_1)}$$

$$\Downarrow$$

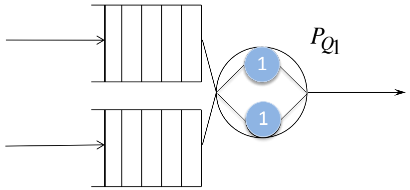
$$E(T_Q) = p_1 \underbrace{\frac{P_{Q1} E(S)}{(1-\rho_1)}} + p_2 \underbrace{\frac{P_Q E(S)}{(1-\rho)(1-\rho_1)}}$$

Devo calcolare p(0), sostituendo a TUTTI i rho
il valore rho_1;
ottenuto p(0) in funzione di rho_1
lo metto in Pq_1, anche qui calcolandolo con TUTTI rho_1
In questo "Pezzo" NON USO MAI IL RHO NORMALE.

non devo ricalcolare nulla,
uso i "vecchi pezzi"

7

Multiserver with priority classes



m=2
p_1=0.35
p_2=0.65
ρ = 0.4
ρ_1=0.14
ρ_2=0.26

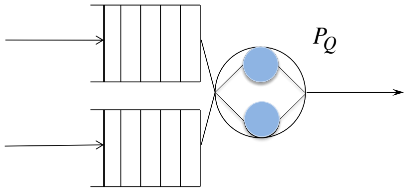
$$P_{Q1} = Erlang(\rho_1) = 0.03438$$

$$\frac{(m\rho_1)^m \cdot p(0)}{m(1-\rho_1)}$$

8

La prima classe (la componente E[Tq_1] in E[Tq]) vede solo se stessa, in quanto c'è prelazione.

Multiserver with priority classes



$P_{Q1} = Erlang(\rho_1) = 0.03438 \qquad P_Q = 0.22857$

$$E(T_Q) = p_1 \frac{P_{Q1} E(S)}{(1-\rho_1)} + p_2 \frac{P_Q E(S)}{(1-\rho)(1-\rho_1)} = 0.12077$$

QoS requirements:

- Average waiting $T_Q < 0.15$ s **!!**

globalmente bound rispettato, sia per classe 1 che classe 2.

9

QoS requirements:

- For at least 35% of arrivals average response time $T_S < 0.5$ s

Analytical models
priority scheduling

$\lambda = 1$ j/s, $E(S) = 0.4$ s $\Rightarrow$ $\rho = 0.4$

- 1 server of capacity C=10[6] oxerat/s

$E(T_Q) = 0.26$ s

- ~~Dual-core of C/2 each one~~

$$\cancel{E(S_i) = \frac{Z}{\frac{C}{2}} = 2\frac{Z}{C} = 2E(S) = 0.8\ s}$$

Il secondo approccio non potrà mai funzionare, in quanto già solo E[Si] è superiore al bound, e mi manca ancora considerare E[Tq]

prof. Vittoria de Nitto Personè 10

10

QoS requirements:

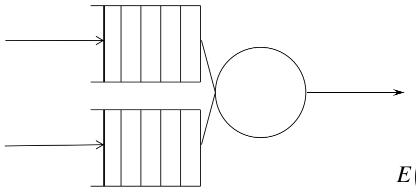
Analytical models
priority scheduling

- For at least 35% of arrivals average response time $T_S < 0.5$ s

$\lambda = 1$ j/s, $E(S) = 0.4$ s $\Rightarrow$ $\rho = 0.4$

- 1 server of capacity C=10^6 oxerat/s

Abstract-P



p_1=0.35
p_2=0.65
$\rho = 0.4$
ρ_1=0.14
ρ_2=0.26

$$E\left(T_{S_1}\right) = 0.4651162$$

prof. Vittoria de Nitto Personè 11

11

Ad E[Tq_1] ho sommato E[S] = 0.4 e NON E[S_i] = 2*0.4 = 0.8, perchè essendoci prelazione non viene mai interrotto un job di classe 1.
Discorso diverso per job di classe 2, che vengono sostituiti (dovrei calcolare E[S_virt_2]).

Non vuol dire che multiserver multicoda sia un silver bullet, possono esistere casi in cui tale tecnica risulta svantaggiosa.