

Sistemi e Applicazioni Cloud - Esercizio

Simulatore reti sbilanciate [Tempo consegna: 2h 30m]

Parte 1: rete base

Si simuli una rete a code che implementa modelli M/M/1 come quella in figura.

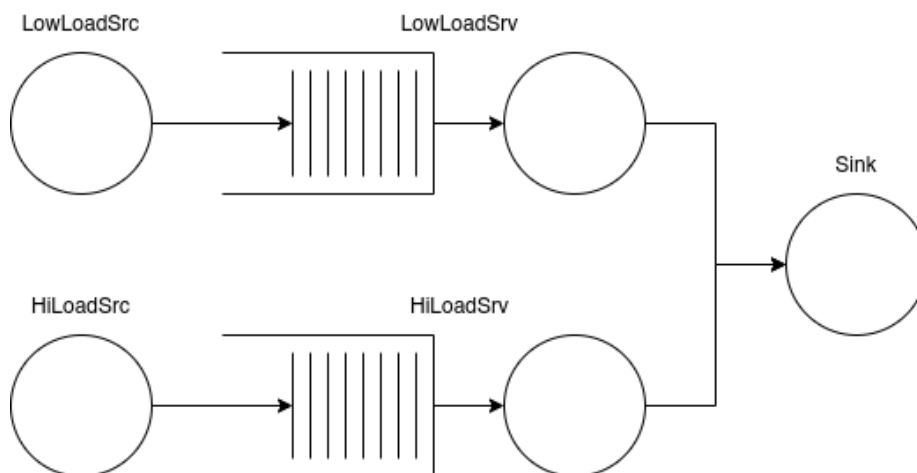


Figure 1: Modello di rete

La rete ha due generatori di carico:

- LowLoadSrc che invia richieste con un tasso $\lambda - \delta$
- HiLoadSrc che invia richieste con un tasso $\lambda + \delta$

Con $\lambda = 6$ richieste al secondo

Entrambi i server hanno un processing rate $\mu = 10$ richieste al secondo. Si chiede di mostrare il tempo di risposta T_r e l'utilizzazione dei server ρ per diversi valori di δ (espresso in richieste al secondo):

δ	Avg(T_r)	StdDev(T_r)	Max(T_r)	ρ (LowLoadSrv)	ρ (HiLoadSrv)
0	0.251	0.004631	2.115	0.600	0.600
1	0.2767	0.006955	3.0568	0.559	0.740
2	0.3947	0.009026	4.6420	0.399	0.8022
3	0.7934	0.02457	7.20429	0.299	0.90157

Parte 2: problema duale

Si consideri la stessa rete di prima. Questa volta però le sorgenti producono lo stesso volume di richieste λ , mentre la capacità di processing è diversi per i due

server, rispettivamente:

- LowLoadSrv che processa dati a un tasso $\mu + \delta$
- HiLoadSrv che processa dati a un tasso $\mu - \delta$

δ	Avg(T_r)	StdDev(T_r)	Max(T_r)	ρ (LowLoadSrv)	ρ (HiLoadSrv)
0	0.254	0.00463	2.1245	0.6004	0.5990
1	0.3642	0.007265	5.3565	0.58462	0.66465
2	0.3335	0.00584	4.2400	0.5802	3.74017
3	0.5852	0.0248	1.0796	0.46047	0.8582

Parte 3: configurazione avanzata

Valutare la possibilità di prendere i dati relativi alle due sorgenti dati. Per farlo:

- Studiare l'uso della proprietà `jobType` nel componente `Source`
- Studiare il componente `Classifier` con particolare riferimento alla proprietà `dispatcherField`
- Modificare la rete per aver due Sink e un Classifier che smista il traffico tra i due
- Ri-valutare i tempi di risposta medi dei due server nei due problemi precedenti

```

# Fasse_1.data
# esame_hk_zhiko F Classifier.ned () esame_json F fase_1.data X
# Fasse_1.data
# signal LifetimeTime_max_low LifetimeTime_max_high signalLifetime_max_low signalLifetime_max_high signalLifetime_max_low signalLifetime_max_high signalBusyTime_low signalBusyTime_high
# 0.250649789999180 0.00380392743548377 0.250974786233576 0.003553464931378665 2.3087680662256 2.306232130282498 2.82158776455996 0.356487791879905 0.6004554626086 0.6004743918148858 0.59999131549438 0.60299908752276816
# 0.1906851982187798 0.002627603181213 0.19106453463864 0.002136631197803 2.1366318179803 2.13608092367234 2.83078684615996 0.416776281375901 0.60266197312437 0.60266145458911 0.780226380319786 0.805517428739925
# 0.022269839464646 0.002269839464646 0.022269839464646 0.002269839464646 2.426983946464646 2.426983946464646 2.426983946464646 0.3997787647628 0.602573797323045 0.602573797323045 0.602573797323045 0.602573797323045
# 0.14936346023256 0.001136322135281 1.031246495380658 0.0046476377408211 4.00866193463 0.132942732488016 7.28029513119 0.111644461182182 0.299529929179516 0.3011160077613618 0.3855747244865 0.3855747244865
# FASE 2
# signal LifetimeTime_max_low LifetimeTime_max_high signalLifetime_max_low signalLifetime_max_high signalLifetime_max_low signalLifetime_max_high signalBusyTime_low signalBusyTime_high
# 0.188881909218 0.00380392743548377 0.250974786233576 0.003553464931378665 2.3087680662256 2.306232130282498 2.82158776455996 0.356487791879905 0.6004554626086 0.6004743918148858 0.59999131549438 0.60299908752276816
# 0.160881890218 0.00166457137359 0.16108143828014 0.00136287732396 2.1366318179803 2.13608092367234 2.83078684615996 0.416776281375901 0.602441535843 0.602441535843 0.80393945432774 0.80393945432774
# 0.022269839464646 0.002269839464646 0.022269839464646 0.002269839464646 2.426983946464646 2.426983946464646 2.426983946464646 0.3997787647628 0.602573797323045 0.602573797323045 0.602573797323045 0.602573797323045
# 0.142467427253418 0.001136322135281 1.027291908764 0.004647637740821 4.15657732777618 0.132942732488016 7.876235861696 0.111644461182182 0.299529929179516 0.3011160077613618 0.3855747244865 0.3855747244865

```