

# Sistemi e Applicazioni Cloud

Appello del 10 settembre 2024 [Tempo consegna: 2h 30m]

## Parte 1: rete base

Si usa un simulatore per studiare il comportamento di un sistema di Edge computing che:

- esegue localmente una parte delle elaborazioni
- scarica una parte delle elaborazioni verso un sistema Cloud remoto che opera secondo un paradigma FaaS

Il sistema è mostrato nella figura.

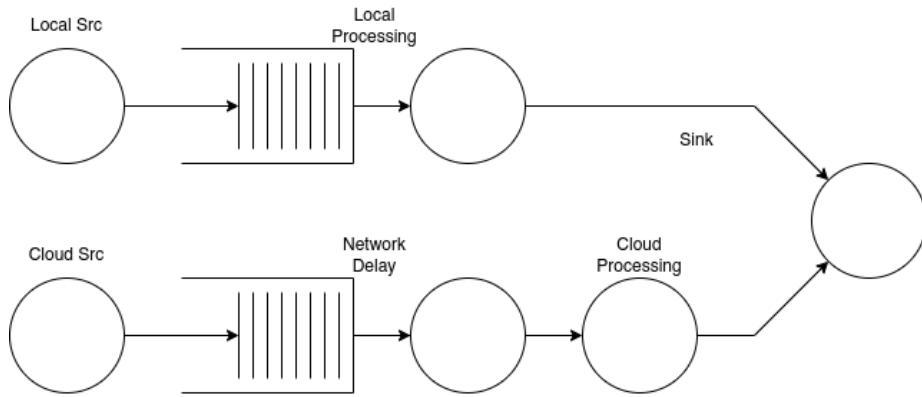


Figure 1: Modello di rete

Il carico in ingresso è  $\lambda = 10$  richieste al secondo. Una frazione  $f_l$  delle richieste sono processate localmente (quindi  $\lambda_l = \lambda f_l$  dove  $\lambda_l$  è il tasso di arrivo nel ramo *Local* della figura), con un server locale che ha un processing rate  $\mu$  di 10 richieste al secondo. Le rimanenti richieste sono inviate al sistema Cloud ( $\lambda_c = \lambda(1 - f_l)$  dove  $\lambda_c$  è il tasso di arrivo nel ramo *Cloud* della figura). Il sistema Cloud opera in modalità FaaS e il tempo di risposta si può modellare come un server con coda che simula la rete (che è suscettibile di congestione) e ha un tempo di servizio di 70 ms per ogni messaggio e da un delay center che modella il tempo di esecuzione della funzione Cloud  $T_c = 1/\mu$ .

Identificare il tempo di risposta complessivo  $T_r$  nel caso in cui  $f_l = 0.7$ . Indicare anche l'intervallo di confidenza del 67% per tali valori [ Valore atteso: 0.292 sec ].

**Nota:** Se non si svolge correttamente questo punto del compito, l'esame non è superato.

## Parte 2: dimensionare il bilanciamento

Identificare mediante la teoria delle reti di code il valore di  $f_l$  tale per cui il tempo di esecuzione del ramo *Local* è pari al tempo di esecuzione del ramo *Cloud*.

## Parte 3: verifica

Eseguire un'analisi del tempo di risposta  $T_r$  per un range di valori di  $f_l \in [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]$ .

Punto bonus: plottare il grafico adattando lo script python fornito.

