

Progettazione ed implementazione di un ambiente di emulazione di sistemi di edge computing e reti mobili 5G

Riccardo Carissimi - Matricola 962766

1 Introduzione

Le attività di tirocinio e tesi sono state svolte presso il laboratorio "Connets" del Dipartimento di Informatica dell'Università degli Studi di Milano e hanno riguardato la realizzazione di un ambiente di emulazione per lo studio di sistemi di edge computing e reti mobili 5G.

2 Contesto

L'evoluzione delle tecnologie di edge computing e delle reti mobili 5G sta rivoluzionando il panorama dell'elaborazione distribuita e dell'interconnessione delle risorse di calcolo. Queste tecnologie offrono una maggiore capacità di calcolo e connettività, consentendo l'esecuzione di applicazioni e servizi avanzati vicino ai dispositivi finali e riducendo la latenza nelle comunicazioni.

La comprensione del funzionamento e delle prestazioni di tali sistemi è fondamentale per garantire la progettazione e l'implementazione efficace di queste architetture. In particolare, l'ottimizzazione degli algoritmi di scheduling, responsabili della gestione delle risorse

e del bilanciamento del carico, è cruciale per garantire prestazioni elevate e un utilizzo efficiente delle risorse disponibili.

3 Obiettivo della tesi

Questa tesi si propone di progettare ed implementare un ambiente di emulazione per reti 5G e sistemi di edge computing, analizzando inoltre i vari algoritmi di scheduling proposti per capirne le caratteristiche e valutarne l'impatto sull'architettura.

4 Lavoro svolto e tecnologie adottate

Abbiamo progettato l'ambiente di emulazione per sfruttare le capacità dei sistemi di orchestrazione per la gestione delle risorse e il bilanciamento del carico nei nodi. È stato scelto di adottare Kubernetes, che ha fornito una soluzione scalabile e flessibile per l'esecuzione di applicazioni distribuite.

Successivamente, la piattaforma di virtualizzazione Proxmox, basata su KVM, è stata utilizzata per creare efficientemente macchine virtuali che emulassero l'ambiente di edge computing.

È stata rivolta particolare attenzione alle performance di rete e come queste impattassero lo scheduling delle richieste sui nodi, a cui abbiamo dedicato una trattazione particolare in sede di analisi.

5 Risultati raggiunti

È stata valutata la bontà dell'architettura progettata e implementata tramite simulazioni di applicazioni reali, grazie a cui siamo riusciti a confermare il raggiungimento degli obiettivi prefissati.

Inoltre abbiamo posto grande attenzione all'analisi degli algoritmi di scheduling implementati da IPVS, a cui Kubernetes si affida.

Siamo quindi riusciti a ricavare caratteristiche e peculiarità dei principali algoritmi di scheduling, al fine di permetterne un uso corretto a seconda del comportamento desiderato dall'ambiente di emulazione.

Sulla base dei risultati dei test e dai dati raccolti possiamo affermare che in un ambiente geograficamente distribuito l'algoritmo *shortest expected delay* è generalmente più performante. Al contrario gli algoritmi *Round-Robin* e *least connections* sono più indicati in realtà dove i nodi di computazione sono in una singola area geografica, in -quanto a parità di richieste e delay tra nodi offrono prestazioni migliori.

Tuttavia, è importante notare come l'ambiente consenta l'analisi di una qualsiasi configurazione di rete e algoritmo di scheduling. Questo è un risultato molto importante che mostra la grande flessibilità dell'ambiente di emulazione progettato e implementato.

Un'importante evoluzione del lavoro svolto finora sarebbe data dall'uso del sistema di simulazione OMNeT++ e del suo framework Simu5G. Grazie a questi sarebbe possibile generare le richieste da user equipment simulati verso l'applicazione reale, permettendo di considerare nella simulazione anche la rete 5G