

Report simulazione "Prima definizione"

Tommaso Lencioni

June 1, 2021

Ho cercato di simulare una prima situazione di deployment di applicazioni su veicoli intelligenti usando architettura Cloud e Fog.
Dopo varie prove ho ottenuto i risultati presentati in questo report.

Infrastruttura

Cloud

Ho lasciato le specifiche del cloud invariate.

Ha due hosts ciascuno con con:

- 8 core
- 2000000 mips
- 16 GB RAM
- 1000000 MB(?) di storage

Fog

Il dispositivo edge datacenter simula una *road side unit* (RSU).

Ho lasciato invariate le prestazioni dell'edge datacenter (2 hosts da 8 core, 800000 MIPS, 16GB RAM, 200000 MB(?) ciascuno) modificando solo la sua posizione (125 x, 125 y cioè al centro della mia simulazione)

Edge

Ho utilizzato un solo tipo di device trattando un veicolo come un unico dispositivo impostando valori che rispecchiano il comportamento di un veicolo in un contesto urbano:

- L'architettura del processore è ARM.¹
- È dotato di mobilità e di una velocità di 14 m/s (50 Km/h).
- I parametri di pausa e durata del movimento sono approssimativamente quelli di traffico scorrevole.
- Anche se non tutta l'energia è dedicabile al calcolo ho inserito come capacità della batteria quella della Nissan Leaf² (per fare gioco sull'efficienza questo numero adrebbe sicuramente ridotto)

¹<https://www.arm.com/solutions/automotive>

²en.wikipedia.org/wiki/Nissan_Leaf

- Come consumo massimo e in idle ho utilizzato i valori di circa 2 Raspberry Pi.

Applicazioni

Ho ipotizzato la presenza di 2 applicativi (originariamente solo 1, si vedano i [Dubbi](#)):

1. SENSOR_DATA_COMMUNICATION

Applicazione che simula la raccolta e comunicazione dei dati forniti dai sensori della vettura.

- Il rate è di 20, questo dovrebbe significare una chiamata ogni 3 secondi.
- Ho provato a settare l'user percentage a 100 (in quanto tutti i dispositivi hanno bisogno di comunicare dati) ma la simulazione non finiva di salvare le chart così ho deciso di diminuirla (questo sicuramente non è corretto ma non ho trovato un modo per ovviare al problema).
- La latenza massima permessa è di un secondo.
- Le dimensioni (container size 15000 KB, request size 100 KB, results size 500 KB e task length 10000 MI) sono indicative.
Ho cercato di dare più peso alla risposta immaginandomi una situazione nella quale ricevo lo status della rete circostante.
- Ho lasciato il numero dei core richiesti a 1 (si vedano i [Dubbi](#)).

2. ENTERTAINMENT

Applicazione che simula l'utilizzo dell'infotainment di una vettura (musica streaming, webradio, etc)

- Il rate è di 10, questo dovrebbe significare una chiamata ogni 6 secondi.
- Per quanto riguarda l'user percentage stessa cosa di sopra.
- Il delay è di 5 secondi (ipotizzando che un contenuto audio sia trasferito in una cache interna al device)
- La dimensione del Container è di 50MB.
- La richiesta è di 100KB e la risposta è di 40MB.

- La lunghezza del task è di 15000 MI.
- Per quanto riguarda i core stessa cosa di sopra.

Scenario

Per questa simulazione ho previsto (riporto le modifiche sostanziali alla simulazione di default):

- Una durata di 60 minuti.
- Una dimensione dell'area di simulazione di $250\text{m} \times 250\text{m}$.
- Il range degli edge devices di 50m
- il range degli edge datacenter di 180m (circa $125 \times \sqrt{2}$ così da inscrivere tutta l'area di simulazione essendo posizionato al centro)
- Il numero minimo di devices è 50 che crescono fino a 150 in scaglioni di 50.
- Ho utilizzato come architettura di orchestrazione CLOUD_ONLY e MIST_ONLY per fare un confronto.

Osservazioni

Dai grafici ho tratti le seguenti conclusioni:

- Il Cloud garantisce un delay di esecuzione minore rispetto all'offload su RSU ([Figure 1](#)).
- Nonostante le applicazioni vengano eseguite in meno tempo in Cloud il tempo di attesa è maggiore e, soprattutto, crescente ([Figure 2](#)).
Ci possiamo aspettare che aumenti con l'aumento del numero dei devices fino a rendere nulla il vantaggio che ha il Cloud sull'execution delay.
- Il consumo di energia è inferiore per l'architettura Mist con una differenza crescente rispetto a Cloud ([Figure 3](#)).
- Il numero dei task falliti a causa del delay è maggiore nell'architettura Cloud (con crescita più rapida rispetto a Mist) ([Figure 4](#)).

Figure 1: Delay medio di esecuzione

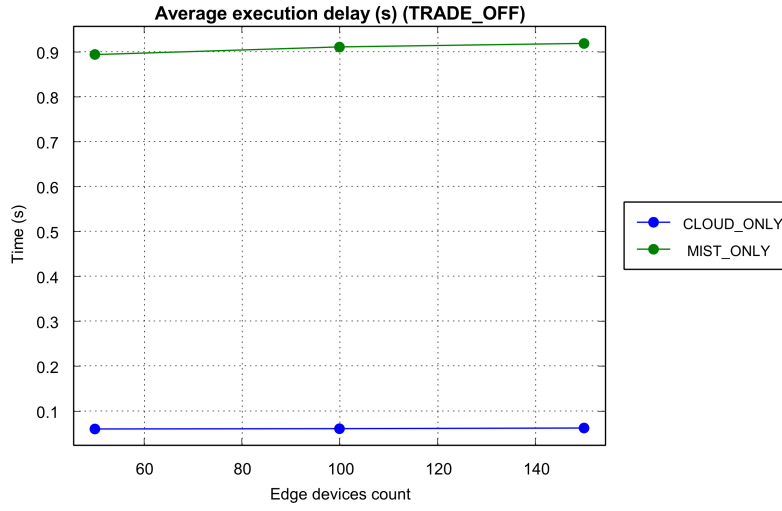
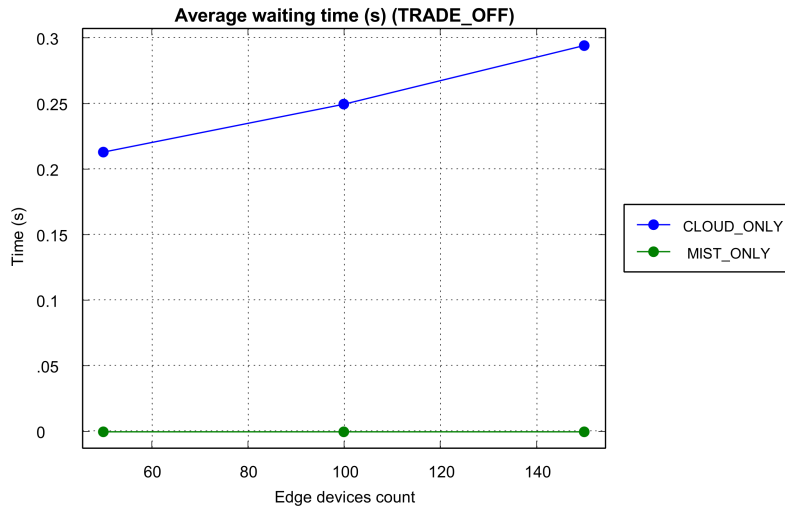


Figure 2: Tempo di attesa medio



- Il numero dei task falliti a causa della mobilità è, come ci si può aspettare, presente solo nell'architettura Mist ([Figure 5](#)).
- I dati dei due grafici precedenti ci portano a concludere che il rischio di un offload su Mist sia palesato in termini di task falliti ([Figure 6](#)).

Figure 3: Consumo di energia

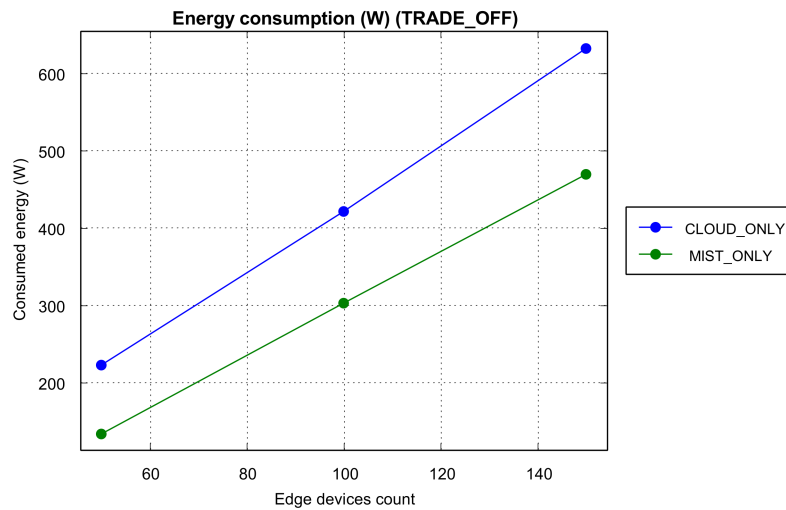


Figure 4: Task falliti a causa dell'elevato delay

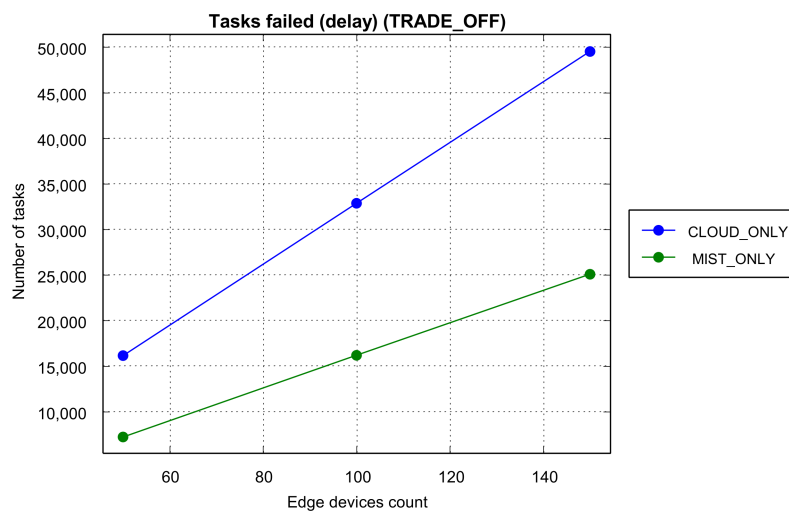


Figure 5: Task falliti a causa della mobilità dei devices

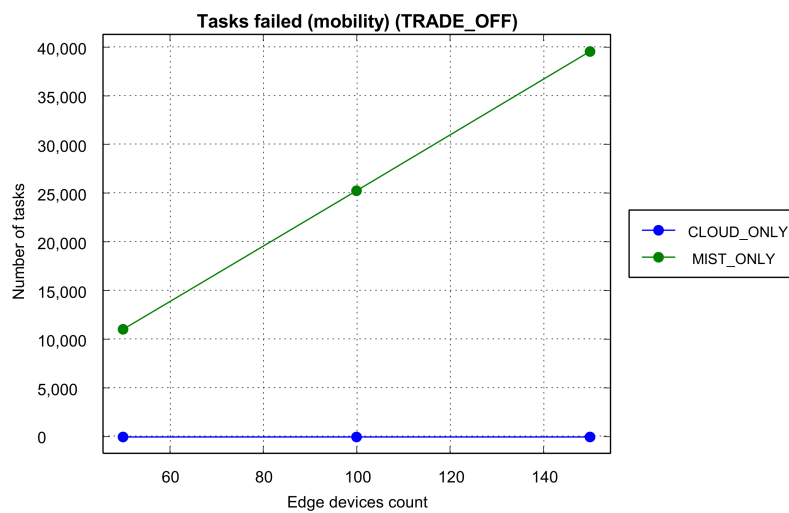
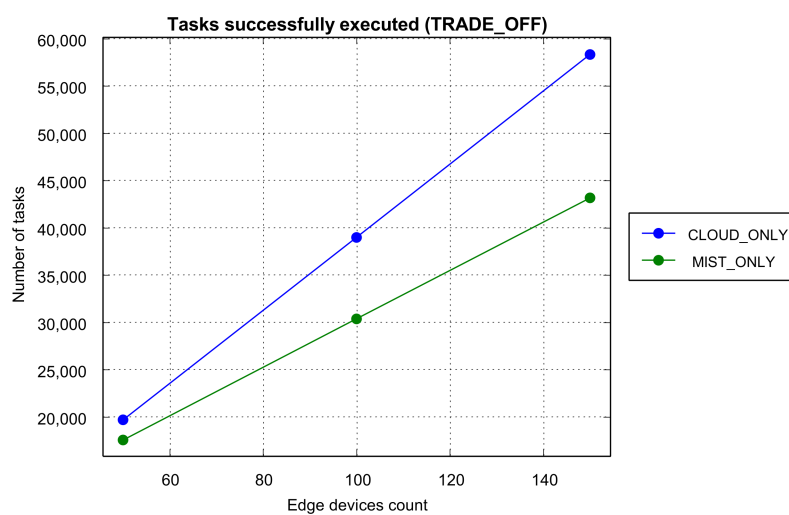


Figure 6: Task eseguiti con successo



- Nonostante i parametri siano "realistici" ho riscontrato un tasso elevato di fallimento che ho cercato di quantificare controllando il file **Sequential_simulation.csv**.

Con il comando

```
awk -F"," '{print $1,$8,$9}' Sequential_simulation.csv
```

ho ottenuto i seguenti dati che evidenziano come circa la metà delle tasks fallisca (percentuale che aumenta con l'architettura Mist)

Architecture	Tot Tasks	Con successo	Percentuale
CLOUD_ONLY	36000	19773	54.93%
CLOUD_ONLY	72000	39057	54.24%
CLOUD_ONLY	108000	58404	54.07%
MIST_ONLY	36000	17639	48.99%
MIST_ONLY	72000	30445	42.28%
MIST_ONLY	108000	43244	40.04%

Table 1: Percentuale di successi

Anche raddoppiando gli edge datacenter la situazione rimane pressoché invariata.

Provando a impostare il delay massimo del servizio di SENSOR_DATA_COMMUNICATION a 5 secondi (secondo me troppo) sono riuscito a ottenere circa l'80% di successo.

- Nei test successivi ho aumentato il range della RSU a 250m in modo che coprisse l'intera area della simulazione ma non ho riscontrato benefici (si vedano i [Dubbi](#)).

Dubbi

Numero di applicazioni

- Inizialmente ho ipotizzato una sola applicazione (quella di comunicazione dei dati raccolti) ma la simulazione non riusciva a salvare i grafici (impiegava molto tempo rispetto ad altre simulazioni, ero costretto a terminare brutalmente l'esecuzione).

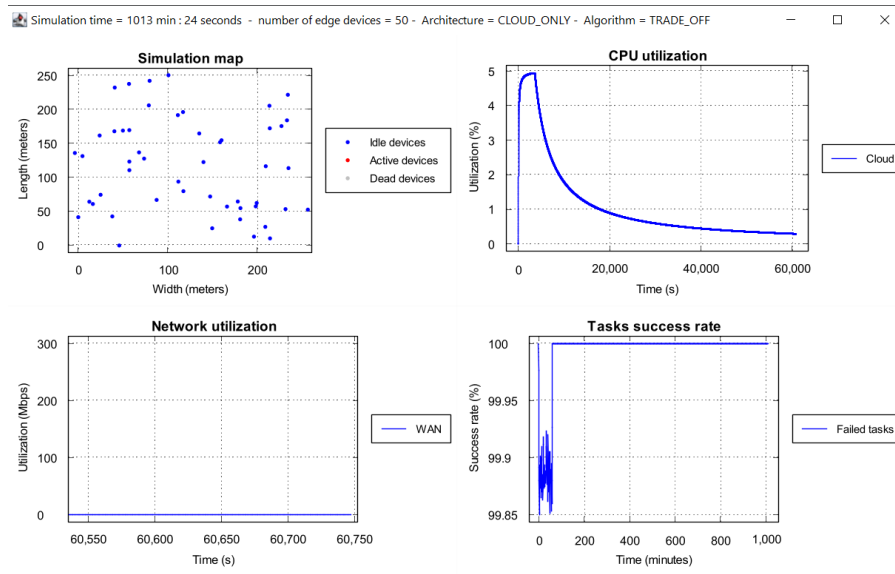


Figure 7: Screen della GUI dopo circa 1000 minuti di simulazione

- Solo dopo aver sperimentato aumentando il delay massimo a 5s ho ottenuto un risultato soddisfacente (come già osservato sopra). Non riesco a capire perché avendo una sola applicazione questa non dovrebbe essere portata a termine.

Numero di core

- Aumentando il numero di core richiesti da un'applicazione (ad esempio 2) ho riscontrato che la simulazione non termina riportando un completamento parziale (e immutabile) nel log di PES (si veda [Screen della GUI dopo circa 1000 minuti di simulazione](#)).
- Modificando *wait_for_all_tasks* a **false** questo problema non si presenta e il numero di tasks falliti diminuisce addirittura con l'architettura Cloud (si veda [Incremento % successi Cloud](#)).

Architecture	Tot Tasks	Con successo	Percentuale
CLOUD_ONLY	36000	31624	87.84%
CLOUD_ONLY	72000	62707	87.09%
CLOUD_ONLY	108000	94175	87.2%
MIST_ONLY	36000	17564	48.79%
MIST_ONLY	72000	30549	42.43%
MIST_ONLY	108000	43070	39.88%

Table 2: Incremento % successi Cloud

Range degli edge datacenter

- Aumentato il range degli edge datacenter (nella mia simulazione uno solo) all'intera area di simulazione non ho riscontrato un miglioramento nel fallimento dei task per mobility (che rimane il motivo principale per il quale i tasks falliscono).

Probabilmente ho frainteso questa metrica.