

## Osservazioni

- Dopo varie prove il fattore **cores** si e' rivelato molto importante in quanto, posizionando una stampa prima della terminazione della simulazione, ho avuto una misura dei task generati ma non eseguiti che, in assenza di indicazioni contrarie, vengono interpretati come task eseguiti con successo.  
Questo avviene solamente con l'opzione `wait_for_all_tasks=false` (se e' settato a true la simulazione non termina).  
Estremizzando il numero di cores (come osservato anche nel report precedente) la quasi totalita' dei task risulta eseguita con successo.  
Negli esempi gli autori mantengono il numero di cores richiesti a 1, continuo' su questa strada anche io.
- Nel file [Orchestratori\\_Notion.pdf](#) ho riassunto gli step della vita di un task con tutti i suoi passaggi tra le varie classi del simulatore.
- In base al comportamento del metodo `offloadingIsPossible` della classe `Orchestrator.java` si ha una situazione chiara di come venga stabilita la possibilita' di offload:
  - Su Cloud e' sempre possibile perche' si presume che tutti i dispositivi possano raggiungere il cloud.
  - Su Edge Datacenter e' subordinato alla sua distanza del dispositivo che genera il task (o, nel caso di orchestrazione attiva, dal suo orchestratore).
  - Su Mist (non piu' edge datacenter ma solo potenza di calcolo degli edge devices) e' permesso solo se il dispositivo che ha la VM che stiamo valutando non sia morto e sia in range (stavolta quello degli edge devices) del dispositivo che ha generato il task (o, nel caso di orchestrazione attiva, dal suo orchestratore).
- Il concetto di Fog inteso come un avvicinamento del Cloud all'Edge non e' presente esplicitamente nel simulatore quindi lo escluderei dal "vocabolario" delle mie prove. Dato che gli edge devices, qualora previsto, possano sempre fare offload su Cloud penso che si possa dare per scontata una vicinanza logica e fisica tipica del Fog Computing.  
Discorso a se stante sono le RSU che sono veri e propri dispositivi (gli unici con questa funzione nei miei test) utilizzati come datacenter e individuati dal simulatore specificatamente come **EdgeDataCenters** e sfruttabili con architettura di calcolo Edge.

- L'utilizzo degli orchestratori e' risultato non vantaggioso nelle mie prove in quanto introduce dell'overhead di comunicazione con il dispositivo che fa questa funzione (a meno che non sia esso stesso il generatore di tasks nel caso in cui gli orchestratori non siano abilitati).  
Questa degradazione di prestazione e' data dal vincolo stringente di 1 secondo di delay massimo sull'applicazione dei sensori.  
Ho provato a monitorare la banda prima dell'assegnazione dell'orchestratore in modo da non assegnarlo qual'ora la rete non potesse garantire i requisiti di delay ma ho notato che l'utilizzo delle due reti e' sempre molto basso, non avrebbe senso fare una valutazione del genere.  
Riporto nella tabella seguente i risultati che ho ottenuto rendendo la RSU un orchestratore (modificando il file `edge_datacenter.xml`) e simulando 150 devices.

Architettura	Orchestratore	Percentuale successo tasks
Cloud	Nessuno	54%
Cloud	Edge	12.41 %
Cloud	Cloud	33.24%
Edge	Nessuno	54%
Edge	Edge	33.37%
Edge	Cloud	12.54%

Ho provato anche a raddoppiare la potenza di calcolo dei datacenters (sia cloud che edge) ma il risultato non cambia.

Allentando il requisito di delay massimo sull'applicativo dei sensori fino a 5s (valore che utilizzano anche gli autori come minimo delay) i risultati sono in linea con la precedente tabella, cioe' abbiamo un'alta percentuale di successo senza orchestratori (circa l'80%) che si abbassa fino al 30% in presenza di orchestratori.

Prima di passare alla manipolazione degli orchestratori vorrei stabilire una baseline in modo da poter avere dati per fare confronti tra le diverse strategie e la loro efficacia .

## Dubbi

- Non ho capito il criterio con il quale viene utilizzato l'intero **phase**. (e.g. *We also started with stage 2 and then stage 1, as this decreases the algorithm complexity*)