

UNIVERSITÀ DEGLI STUDI DI MILANO  
FACOLTÀ DI SCIENZE E TECNOLOGIE

DIPARTIMENTO DI INFORMATICA  
GIOVANNI DEGLI ANTONI



Corso di Laurea triennale in Informatica

PATTERN DI ASSEGNAIMENTO OTTIMIZZATI PER  
L'EFFICIENZA ENERGETICA IN EDGE COMPUTING

Relatore: Prof. Alberto Ceselli  
Correlatore: Prof. Marco Premoli

Tesi di Laurea di:  
Manuel Parati  
Matr. 958584

ANNO ACCADEMICO 2021-2022

# Ringraziamenti

Questa sezione, contiene i ringraziamenti.

# Indice

<b>Ringraziamenti</b>	<b>i</b>
<b>Indice</b>	<b>ii</b>
<b>1 Introduzione</b>	<b>1</b>
1.1 Obbiettivi . . . . .	1
1.2 Struttura dell’elaborato . . . . .	1
<b>2 Modellazione del sistema</b>	<b>2</b>
2.1 L’infrastruttura Mobile Edge Computing . . . . .	2
2.2 Modello di orchestrazione proposto . . . . .	3
2.2.1 Esempio del funzionamento . . . . .	3
2.2.2 Implementazione . . . . .	5
<b>3 Modelli di ottimizzazione</b>	<b>6</b>
3.1 Modello completo . . . . .	6
3.2 Modello con migrazioni a costo 0 . . . . .	7
3.3 Modello con migrazioni a costo infinito . . . . .	7
<b>4 Algoritmi euristici</b>	<b>8</b>
4.1 Semplificazione del modello . . . . .	8
4.2 Aggregamento dei time-slot . . . . .	8
<b>Bibliografia</b>	<b>9</b>

# Capitolo 1

## Introduzione

...

### 1.1 Obbiettivi

...

### 1.2 Struttura dell'elaborato

...

# Capitolo 2

## Modellazione del sistema

Questo capitolo fornisce una panoramica riguardo all'infrastruttura del Mobile Edge Computing ed introduce il modello di orchestrazione proposto.

### 2.1 L'infrastruttura Mobile Edge Computing

All'interno di un'infrastruttura Mobile Edge Computing (MEC) si trovano i cluster di virtualizzazione, spesso chiamati 'MEC facility' o più semplicemente 'facility', e gli access point (AP). Le facility sono il luogo in cui avviene la virtualizzazione, e sono composte dalle macchine virtuali (VM) su cui si eseguono le applicazioni degli utenti finali. Gli AP sono invece i dispositivi (per esempio le antenne wireless) a cui gli end point si collegano per poter ricevere il servizio. Ogni AP è associato ad una facility, a cui inoltra tutto il traffico che riceve: questo significa che ogni facility avrà in esecuzione nelle proprie VM tutte le applicazioni utilizzate dagli utenti connessi agli AP a lui associati.

Data la natura mobile degli end point, che sono per esempio smartphone o laptop, il traffico a cui sono sottoposti gli AP varia nel tempo e di conseguenza cambia la domanda rivolta alle facility. Per questo motivo l'assegnamento viene effettuato dinamicamente, e lo strumento incaricato di tale compito è l'orchestratore, che implementa la logica definita dal modello di orchestrazione. Le azioni che svolge vengono chiamate 'orchestrazioni' e consistono nell'assegnare un AP ad una facility diversa da quella attuale, provocando il ridimensionamento della potenza delle VM in termini del numero di processori e memoria utilizzata, e la migrazione del loro stato. Il ridimensionamento è dovuto alla variazione di domanda da soddisfare, mentre la migrazione dello stato è necessaria per avere in esecuzione in ogni facility le applicazioni utilizzate dagli utenti connessi agli AP che le sono assegnati. Il costo prodotto dalla migrazione prende il nome di 'costo di migrazione'.

Nel sottocapitolo successivo viene illustrato il modello di orchestrazione proposto in questo lavoro.

## 2.2 Modello di orchestrazione proposto

Il modello di orchestrazione proposto è una variante di quello presentato negli articoli [1], [2] in cui nell'effettuare le scelte di orchestrazione si tiene in considerazione anche l'utilizzo dell'energia da parte delle facility.

Come illustrato in [2], all'interno del modello viene introdotta una discretizzazione temporale che permette di effettuare migrazioni solo in determinati istanti, per esempio ogni 15 minuti. Questa scelta è ragionevole, dato che in caso contrario si pagherebbe un costo troppo elevato in termini di migrazione e spese generali dovute alla gestione del traffico. L'orizzonte temporale viene quindi rappresentato da un insieme di time-slot della stessa durata.

L'obiettivo del modello è quello di effettuare assegnamenti che permettano di ottenere un buon compromesso tra: una buona qualità del servizio (cioè bassa latenza per l'utente finale), costi di migrazione contenuti e buona gestione dell'energia da parte delle facility. Nell'effettuare le scelte bisogna inoltre tenere in considerazione che la domanda degli AP varia nel corso del tempo e che ogni facility possiede un limite massimo di domanda che è in grado di soddisfare simultaneamente.

Dal punto di vista energetico, le facility utilizzano in ogni slot time una quantità di energia direttamente proporzionale alla domanda complessiva che gli viene rivolta. A tale scopo, ognuna di esse possiede un certo numero di pannelli fotovoltaici che le permettono di generare energia variabile nel tempo, che può essere direttamente utilizzata oppure immagazzinata all'interno delle proprie batterie per i time-slot successivi. Le batterie hanno una capacità limitata e nel caso in cui l'energia prodotta non basti a soddisfare la domanda, è possibile acquistarla ad un prezzo che varia in base all'istante temporale.

### 2.2.1 Esempio del funzionamento

Una semplice applicazione di esempio è presentata nella Figura 1. Come si può vedere, è presente una rete MEC con due facility (K1 e K2) e quattro AP (A, B, C, D) considerata in due time-slot consecutivi ( $t=0, 1$ ). Gli end point connessi agli AP sono rappresentati con dei piccoli cerchi. Si supponga che in ogni time-slot per gestire la domanda di ogni utente sia necessario utilizzare una singola unità di energia e che inizialmente le batterie siano vuote in entrambe le facility. Nel primo time-slot ( $t=1$ ) gli AP A e B sono associati alla facility K1 mentre C e D a K2, quindi tutti gli utenti che sono connessi ad A e B avranno le proprie applicazioni

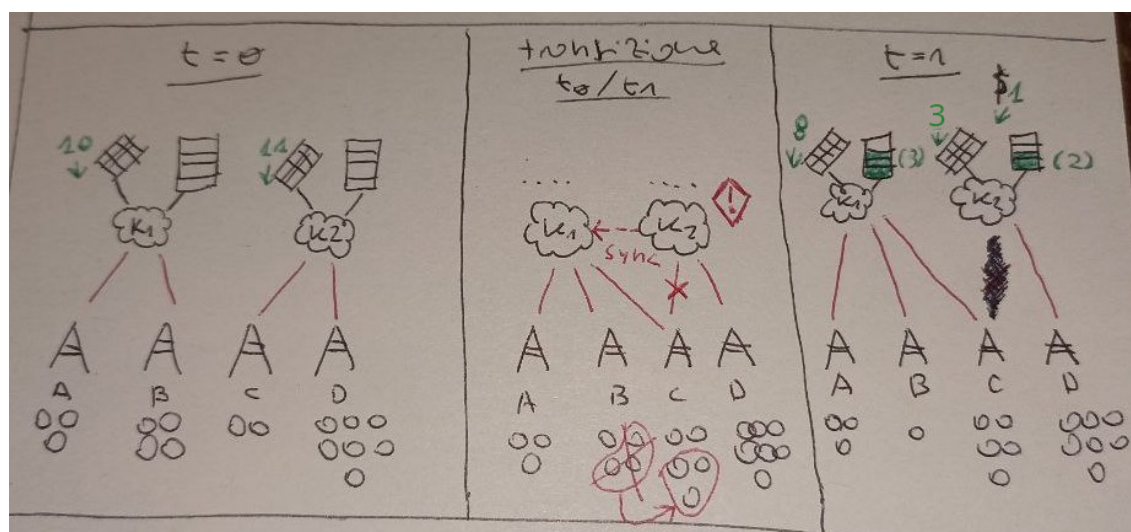


Figura 1: Esempio funzionamento del modello di orchestrazione.

in esecuzione su una VM presente in K1, mentre quelli connessi a C e D le avranno in K2. Questi assegnamenti permettono di avere una buona latenza, dato che ogni AP è connesso alla facility più vicina, ma anche una buona gestione energetica, in quanto nella facility K1 vengono prodotte 10 unità di energia ed utilizzate solo 7 mentre in K2 se ne producono 11 e utilizzano 9. In entrambi i casi avanzano delle unità (3 in K1 e 2 in K2) che vengono immagazzinate nelle batterie per poter essere spese nei prossimi time-slot. Successivamente tre utenti connessi all'AP B si spostano e vengono agganciati da C: a questo punto K2 riceve una richiesta di domanda eccessiva che non riesce a gestire e di conseguenza è necessario effettuare un'azione di orchestrazione per ribilanciare il traffico. Questo comporta assegnare C a K1, ridimensionare le VM (aumentarne la potenza in K1 ed abbassarla in K2) e sincronizzare le facility trasferendo lo stato delle VM riguardanti tutti gli utenti connessi a C da K2 a K1. In un contesto reale, il modello deve prevenire una situazione di questo tipo effettuando orchestrazioni che tengano in considerazione la futura domanda dei vari AP, dato che questa circostanza va ad inficiare la qualità del servizio offerto. Nel secondo time-slot ( $t=1$ ) si avranno quindi gli AP A, B e C assegnati alla facility K1, mentre D a K2. Dal punto di vista energetico, la facility K1 riesce a soddisfare la domanda di traffico (9 unità) utilizzando l'energia presente nella batteria e quella prodotta in questo time-slot (8 + 3 unità complessive). Per quanto riguarda invece K2, l'energia disponibile (3 + 2 unità) non è sufficiente per colmare la domanda, e quindi è necessario acquistare una ulteriore unità. Da notare come l'energia potrebbe avere costi diversi nel corso dei time-slot, e di conseguenza per il modello potrebbe risultare conveniente acquistarla quando il costo è basso per immagazzinarla nelle batterie ed utilizzarla successivamente.

### 2.2.2 Implementazione

Il comportamento del modello di orchestrazione fin qui descritto viene implementato attraverso un modello di programmazione lineare misto-intera (MILP), a cui d'ora in poi ci si riferirà con il nome di 'modello completo'. Il modello completo, descritto dettagliatamente nella sezione 3.1, rappresenta la miglior metodologia per risolvere il problema in quanto garantisce di ottenere la soluzione ottima, ma data la sua natura combinatoria potrebbe essere difficilmente utilizzabile in alcune situazioni. Infatti quando l'arco temporale preso in considerazione oppure il numero di AP e facility tendono ad accrescere, il tempo di calcolo aumenta esponenzialmente fino a diventare proibitivo.

Per questo motivo vengono successivamente proposte due euristiche che semplificano il problema con l'obiettivo di abbassare i tempi di calcolo ed ottenere una soluzione non troppo peggiorata rispetto a quella migliore. Nella sezione 4.1 viene rappresentata la prima idea euristica, che consiste nell'utilizzare una semplificazione del modello completo, mentre nella sezione 4.2 la seconda, che aggrega i time-slot in modo tale da semplificare l'istanza su cui il modello completo opera.



# Capitolo 3

## Modelli di ottimizzazione

In questo capitolo vengono introdotti i modelli di ottimizzazione utilizzati.

### 3.1 Modello completo

...

$$\min \quad \alpha \sum_{t \in T} \sum_{i \in A} \sum_{(j,k) \in \substack{K \times K \\ K \times K}} d_i^t l_{jk} y_{ijk}^t + \beta \sum_{t \in T} \sum_{i \in A} \sum_{k \in K} d_i^t m_{ik} x_{ik}^t + \gamma \sum_{t \in T} \sum_{k \in K} c_k^t z_k^t \quad (1)$$

$$\text{s.t.} \quad \sum_{i \in A} d_i^t x_{ik}^t = v_k^t \quad \forall k \in K, \forall t \in T \quad (2)$$

$$\sum_{k \in K} x_{ik}^t = 1 \quad \forall i \in A, \forall t \in T \quad (3)$$

$$x_{ik}^t = \sum_{l \in K} y_{ilk}^t \quad \forall i \in A, \forall k \in K, \forall t \in T \setminus \{1\} \quad (4)$$

$$x_{ik}^t = \sum_{l \in K} y_{ikl}^{t+1} \quad \forall i \in A, \forall k \in K, \forall t \in T \setminus \{T\} \quad (5)$$

$$z_k^1 + e_k^1 + p_k \geq v_k^1 + g_k^1 \quad \forall k \in K \quad (6)$$

$$z_k^t + e_k^t + g_k^{t-1} \geq v_k^t + g_k^t \quad \forall k \in K, \forall t \in T \setminus \{1\} \quad (7)$$

$$g_k^t \leq G_k \quad \forall k \in K, \forall t \in T \quad (8)$$

$$v_k^t \leq C_k \quad \forall k \in K, \forall t \in T \quad (9)$$

$$x_{ik}^t \in \{0, 1\} \quad \forall i \in A, \forall k \in K, \forall t \in T \quad (10)$$

$$y_{ik'k''}^t \in \{0, 1\} \quad \forall i \in A, \forall k', k'' \in K, \forall t \in T \quad (11)$$

$$x, y, g, v, z \geq 0 \quad (12)$$

...

### 3.2 Modello con migrazioni a costo 0

...

$$\min \quad \beta \sum_{i \in A} \sum_{k \in K} d_i m_{ik} x_{ik} + \gamma \sum_{k \in K} c_k z_k \quad (13)$$

$$\text{s.t.} \quad \sum_{i \in A} d_i x_{ik} = v_k \quad \forall k \in K \quad (14)$$

$$\sum_{k \in K} x_{ik} = 1 \quad \forall i \in A \quad (15)$$

$$z_k + e_k + p_k = v_k + g_k \quad \forall k \in K \quad (16)$$

$$v_k \leq C_k \quad \forall k \in K \quad (17)$$

$$x_{ik} \in \{0, 1\} \quad \forall i \in A, \forall k \in K \quad (18)$$

$$a, x, p, v, z \geq 0 \quad (19)$$

...

### 3.3 Modello con migrazioni a costo infinito

...

$$\min \quad \beta \sum_{t \in T} \sum_{i \in A} \sum_{k \in K} d_i^t m_{ik} x_{ik} + \gamma \sum_{t \in T} \sum_{k \in K} c_k^t z_k^t \quad (20)$$

$$\text{s.t.} \quad \sum_{i \in A} d_i^t x_{ik} = v_k^t \quad \forall k \in K, \forall t \in T \quad (21)$$

$$\sum_{k \in K} x_{ik} = 1 \quad \forall i \in A \quad (22)$$

$$z_k^1 + e_k^1 + p_k \geq v_k^1 + g_k^1 \quad \forall k \in K \quad (23)$$

$$z_k^t + e_k^t + g_k^{t-1} \geq v_k^t + g_k^t \quad \forall k \in K, \forall t \in T \setminus \{1\} \quad (24)$$

$$g_k^t \leq G_k \quad \forall k \in K, \forall t \in T \quad (25)$$

$$v_k^t \leq C_k \quad \forall k \in K, \forall t \in T \quad (26)$$

$$x_{ik} \in \{0, 1\} \quad \forall i \in A, \forall k \in K \quad (27)$$

$$x, g, v, z \geq 0 \quad (28)$$

...

# Capitolo 4

## Algoritmi euristici

4.1 Semplificazione del modello

4.2 Aggregamento dei time-slot

# Bibliografia

- [1] Alberto Ceselli, Marco Fiore, Marco Premoli, and Stefano Secci. Optimized assignment patterns in mobile edge cloud networks. *Computers & Operations Research*, 106:246–259, 2019.
- [2] Alberto Ceselli, Marco Fiore, Angelo Furno, Marco Premoli, Stefano Secci, and Razvan Stanica. Prescriptive analytics for mec orchestration. In *2018 IFIP Networking Conference (IFIP Networking) and Workshops*, pages 1–9, 2018.