Sensor mapping (parte 1)

1. Modelli di ottimizzazione

- Step per risolvere il problema del sensor mapping
  - **Definizione di un modello di performance del sistema distribuito**
    - Modellazione di un problema di ottimizzazione
    - Definizione di una notazione formale
  - **Definizione di un'euristica per risolvere il problema**
    - Modellazione tramite algoritmi genetici
  - **Effettuare una valutazione delle performance**
    - Come si avvicinano le performance dell'euristica alla soluzione ottima?
    - Quali sono le differenze nei tempi di esecuzione per trovare la soluzione di mapping?
- Modello di performance
  - Nodi
    - Sensori
    - Nodi fog
    - Cloud data centers
  - Delay di rete
    - Da sensore a fog node
    - Da fog node a cloud data center
    - Dovuti a — Connessioni non cablate — Lente e maggiormente error prone
  - **Obiettivo** — Mappare i data flow da sensori a fog nodes per minimizzare i tempi di risposta
  - **Assunzioni**
    - Un sensore manda i suoi dati a un nodo fog
      - Ogni sensore manda tutti i dati a solo un fog node
      - Nessuna ridondanza
    - Non consideriamo il problema simile per i fog nodes
      - Mappare i fog nodes ai cloud data centers
      - Problema banale
        - Assumiamo non ci sia rischio di sovraccaricare il cloud
        - Il nodo fog può mandare i dati al DC più vicino
- **Notazione formale**
  - **Descrizione del comportamento di un nodo**
    - Datarate dei dati in uscita dal sensore i — λi — Req/s
    - Processing rate al nodo fog j — μj — Req/s
  - **Delay della rete**
    - Dal sensore i al nodo fog j — δij — s
    - Dal nodo fog j al datacenter cloud k — δjk — s
  - **Variabili decisionali**
    - Mapping tra sensori e fog nodes — xij
    - Mapping tra fog nodes e cloud datacenters — yjk
    - Booleane
      - Ex. xij =
        - 1 se il sensore i sta mandando dati al nodo fog j
        - 0 altrimenti
      - Stessa cosa per yjk — Nel modello finale il mapping sul cloud è semplificato
  - Conclusioni

Symbol	Meaning/Role
Model parameters	
$\mathcal{S}$	Set of sensors
$\mathcal{F}$	Set of fog nodes
$\mathcal{C}$	Set of cloud data centers
λ_i	Outgoing data rate from sensor i
λ_j	Incoming data rate at fog node j
$1/\mu_j$	Processing time at fog node j
$\delta_{i,j}$	Communication latency between sensor i and fog node j
$\delta_{j,k}$	Communication latency between fog j and cloud k
Model variables	
i	Index of a sensor
j	Index of a fog node
k	Index of a cloud data center
Data flows description	
$x_{i,j}$	Data flow from sensor i to fog node j
$y_{j,k}$	Data flow from fog node j to cloud data center k
Scenario description	
δ	Average network delay
$\delta\mu$	Network delay to processing time ratio
ρ	Infrastructure load

- **Problema di ottimizzazione**
  - **Funzione obiettivo**
    - Minimizzare il **response time**
    - Assunzione semplificante — Non consideriamo il ritorno della risposta al sensore, ma solo il completamento del servizio
  - **Contributi al response time**
    - Delay della rete dai sensori ai fog nodes — T_{netSF}
      - $T_{netsf} = \frac{1}{\sum_{i\in\mathcal{S}} \lambda_i} \sum_{i\in\mathcal{S}} \sum_{j\in\mathcal{F}} \lambda_i x_{i,j} \delta_{i,j}$
      - Il ritardo medio dal sensore i al nodo fog j (δij) è pesato con il quantitativo di traffico che subisce quel ritardo (λi)
    - Delay da fog nodes al cloud — T_{netFC}
      - $T_{netfc} = \frac{1}{\sum_{j\in\mathcal{F}} \lambda_j} \sum_{j\in\mathcal{F}} \sum_{k\in\mathcal{C}} \lambda_j y_{j,k} \delta_{j,k}$
      - Il ritardo medio dal nodo fog j al cloud k (δjk) è pesato con il quantitativo di traffico che subisce quel ritardo (λj)
      - λj definito come data rate in input al nodo fog j — $\lambda_j = \sum_{i\in\mathcal{S}} x_{i,j} \cdot \lambda_i$ — Ovvero la somma dei data rates dei sensori allocati al nodo j
    - Processing time ai nodi fog — T_{proc}
      - Modellato secondo la teoria delle code
      - Una stima di questo componente, consistente con altri risultati in letteratura corrisponde a questa formula — $T_{proc} = \frac{1}{\sum_{j\in\mathcal{F}} \lambda_j} \sum_{j\in\mathcal{F}} \frac{\lambda_j}{\mu_j - \lambda_j}$
    - Response time — $T_R = T_{netSF} + T_{netFC} + T_{proc}$
  - **Vincoli**
    - $\sum_{j\in\mathcal{F}} x_{i,j} = 1 \quad \forall i \in \mathcal{S},$ — Ogni sensore deve mandare dati a un solo nodo fog — Un vincolo simile c'è anche per yjk
    - $\lambda_j < \mu_j \quad \forall j \in \mathcal{F},$ — I nodi fog non dovrebbero essere sovraccaricati

2. Definizione dello scenario

- Assunzioni semplificatrici per lo scenario globale
  - Il delay è proporzionale alla distanza tra nodi
  - Il datarate di input e il processing rate sono omogenei
- Parametri che descrivono scenari differenti
  - **Topologia di rete**
  - Requirements — **Tipi di applicazioni**
- Delay della rete medio
  - δ
  - $\delta = \frac{\sum_{i\in\mathcal{S}} \sum_{j\in\mathcal{F}} \delta_{i,j} + \sum_{j\in\mathcal{F}} \sum_{k\in\mathcal{C}} \delta_{j,k}}{|\mathcal{S}| \cdot |\mathcal{F}| + |\mathcal{F}| \cdot |\mathcal{C}|}$
  - Collegato alle caratteristiche **statiche** del sistema distribuito
- Carico del sistema
  - ρ
  - Carico medio dei nodi fog
  - $\rho = \frac{\sum_{i\in\mathcal{S}} \lambda_i}{\sum_{j\in\mathcal{F}} \mu_j}$
  - Collegato alle caratteristiche **dinamiche** del sistema
- Rapporto tra network delay e processing time
  - $\delta\mu$ — $\delta/(1/\mu)$
  - Identifica scenari
    - CPU-bound
    - Network-bound
  - $\delta\mu = \delta \cdot \frac{\sum_{j\in\mathcal{F}} \mu_j}{|\mathcal{F}|}$
  - Collegato alle caratteristiche **statiche** del sistema distribuito
- Diversi scenari corrispondono a diverse combinazioni di δμ e ρ
  - Si considerino questi range
    - $0.01 \leq \delta\mu \leq 10$
    - $0.2 \leq \rho \leq 0.9$
  - δμ = 0.01, ρ = 0.8
    - Rappresenta un caso dove il ritardo della rete è molto più basso del tempo di processing di un job medio
    - E la domanda di processing è alta
    - Questo processo è **CPU-bound** — Gestire le richieste computazionali è il punto principale per ottimizzare la funzione obiettivo
  - δμ = 8, ρ = 0.2
    - Bassa intensità di carico di lavoro
    - Network delay maggiore del tempo di servizio di un job
    - Questo processo è **Network-bound** — I ritardi della rete sono il punto principale per ottimizzare la funzione obiettivo
  - δμ = 1, ρ = 0.2
    - Bassa intensità di carico di lavoro
    - Network delay comparabile al tempo di servizio di un job
    - Diventa importante ottimizzare il contributo della rete nella funzione obiettivo

3. Problema semplificato

- Gestire il mapping fog to cloud è banale
  - Per ogni nodo fog mandiamo i dati al datacenter più vicino
  - Non c'è rischio di congestionare o sovraccaricare il cloud
- Semplificazione del nostro problema
  - Eliminiamo tutti gli elementi legati al mapping fog to cloud
  - Dunque
    - T_{netFC} — Delay da fog nodes al cloud
    - Delay nodo fog j al datacenter cloud k — δjk — s
    - Mapping tra fog nodes e cloud datacenters — yjk
- Da qui per trovare la soluzione dobbiamo eseguire il problema di ottimizzazione e trovare xij — Ovvero il mapping ottimo tra sensore i e fog node j
- Non è comunque un problema facile
  - Necessitiamo di forti assunzioni
  - La soluzione funziona solo per problemi di piccole dimensioni
  - Abbiamo inoltre
    - Lunghi tempi di esecuzione
    - Problema NP-hard con funzioni non lineari
    - Soluzione non scalabile