UNIVERSITÀ DEGLI STUDI DI TRENTO

Dipartimento di Ingegneria e Scienza dell'Informazione

Corso di Laurea in
Informatica

Elaborato finale

Titolo

Sottotitolo (alcune volte lungo - opzionale)

Supervisore
Montresor Alberto

Laureando
Giust Alberto

Anno accademico 2018/2019

Ringraziamenti

...thanks to...

Indice

Sommario

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sed nunc orci. Aliquam nec nisl vitae sapien pulvinar dictum quis non urna. Suspendisse at dui a erat aliquam vestibulum. Quisque ultrices pellentesque pellentesque. Pellentesque egestas quam sed blandit tempus. Sed congue nec risus posuere euismod. Maecenas ut lacus id mauris sagittis egestas a eu dui. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque at ultrices tellus. Ut eu purus eget sem iaculis ultricies sed non lorem. Curabitur gravida dui eget ex vestibulum venenatis. Phasellus gravida tellus velit, non eleifend justo lobortis eget.

Sommario è un breve riassunto del lavoro svolto dove si descrive l'obiettivo, l'oggetto della tesi, le metodologie e le tecniche usate, i dati elaborati e la spiegazione delle conclusioni alle quali siete arrivati.

Il sommario dell'elaborato consiste al massimo di 3 pagine e deve contenere le seguenti informazioni:

- contesto e motivazioni
- breve riassunto del problema affrontato
- tecniche utilizzate e/o sviluppate
- risultati raggiunti, sottolineando il contributo personale del laureando/a

1 In ante nulla, vestibulum a

1.1 Pellentesque habitant morbi tristique senectus

1.2 Nullam et justo vitae nisi

2 Protocolli epidemici nel dettaglio

Un protocollo epidemico è un modello di comunicazione che trae ispirazione dallo studio della diffusione di epidemie. In modo intercambiabile si può utilizzare il termine protocollo di gossip, derivante dall'omonimo fenomeno studiato nell'ambito delle scienze sociali come metodo efficace per il passaggio di informazioni in una rete sociale.

Il gossip e le epidemie sono stati analizzati e le loro caratteristiche sono state implementate in reti informatiche, nello specifico in sistemi distribuiti. Le regole su cui si basa il loro funzionamento sono semplici, ma allo stesso tempo sono robusti veloci ed affidabili. Inoltre, non è necessario alcun controllo centralizzato, condizione necessaria affinché possano essere utilizzati nei sistemi distribuiti

2.1 Storia

I protocolli epidemici vengono analizzati per la prima volta in un documento scritto da Alan Demers [fonte] nel 1987. Il problema riscontrato da Demers e dai suoi colleghi nella rete interna del centro di ricerca di Xerox a Palo Alto era il seguente: mantenere la consistenza tra più copie di un database presente nelle diverse macchine. Il loro obiettivo consisteva nel progettare algoritmi robusti, efficienti

e con elevata scalabilità, tenendo conto del tempo necessario per la distribuzione di un aggiornamento in una rete ed il traffico generato. Dopo aver esaminato un protocollo best-effort chiamato direct mail, che prevede l'invio in broadcast a tutti i nodi di una rete dell'aggiornamento ricevuto ed aver notato che non era efficiente né affidabile (è possibile che un messaggio venga perso e non è sempre scontato che un nodo conosca tutti i componenti in una rete), le loro attenzioni si sono spostate verso due tipologie di protocolli epidemici: anti-entropy e rumor-mongering. Entrambi prevedono lo scambio periodico di informazioni tra nodi scelti in modo casuale, cercando di risolvere le differenze che intercorrono tra i due database dei rispettivi end-point. Attraverso questi approcci si può ottenere consistenza eventuale (eventual consistency), una forma di consistenza debole, tale per cui un sistema di storagee garantisce che se non ci sono ulteriori aggiornamenti, tutti gli accessi ritorneranno il valore più aggiornato [eventual_consistency].

I protocolli di gossip sono utilizzati oggigiorno in numerosi contesti: in sistemi peer-to-peer che necessitano di scambio di informazioni, come il sistema di condivisione di file BitTorrent [bittorrent] oppure nella rete BitCoin [serve fonte se possibile, ho trovato solo forum che ne parlano].

2.2 Information dissemination

La prima applicazione che verrà analizzata riguarda la distribuzione di informazioni, come avviene naturalmente nel gossip. Gossip nei sistemi distribuiti significa scambiarsi informazioni in modo periodico e probabilistico tra due membri [gossiping_in_distributed_systems]. È importante notare che questo processo viene eseguito ripetutamente e di per se non ha una condizione di terminazione, ma verrà introdotta parlando del modello SIR e degli algoritmi di rumor-mongering.

2.2.1 Componenti e notazione

Si consideri una rete composta da un numero fissato P di nodi. Il grafo generato sarà completo, ovvero ogni nodo potrà comunicare direttamente con tutti gli altri nodi. Ogni nodo contiene una variabile chiamata *value* inizialmente uguale per tutti. I nodi potranno scambiarsi messaggi contenenti *value*, la quale avrà un attributo *value.timestamp* che indicherà la data e l'ora dell'ultimo aggiornamento. L'obiettivo di questi algoritmi sarà: in assenza di ulteriori aggiornamenti, *value* sarà uguale in tutti i nodi.

Ogni nodo possiede anche un attributo status che potrà assumere tre valori, ispirati alla terminologia utilizzata in epidemiologia:

- **Susceptible**(S): un nodo che non è venuto a conoscenza di un aggiornamento
- **Infected**(I): un nodo che è venuto a conoscenza dell'aggiornamento e lo sta distribuendo attivamente
- **Removed**(R): un nodo che è venuto a conoscenza dell'aggiornamento ma non lo distribuisce più

Gli stati *susceptible* e *infected* verranno utilizzati nel modello SI, mentre aggiungendo lo stato *removed* si parlerà di modello SIR.

2.2.2 Epidemie semplici

Il modello SI, chiamato anche anti-entropy o delle epidemie semplici, è il primo protocollo studiato nel centro di ricerca di Xerox. Come già detto, un nodo potrà trovarsi nello stato *susceptible* o *infected*. L'invio periodico di informazioni è scandito da un timer proprio del nodo, impostato inizialmente al valore Δ. Quando il timer scende a zero, il nodo invia il messaggio secondo lo stile scelto ed imposta nuovamente il timer. È interessante notare ogni nodo esegue queste operazioni una sola volta per round.

Nel documento originale vengono esposti e studiati tre stili per il modello SI, che cambiano il modo in cui i componenti della rete comunicano e risolvono le differenze.

Si utilizzerà la seguente notazione: n indica il numero totale dei nodi presenti sulla rete ($|P|$), mentre i valori s e i indicano rispettivamente il rapporto $|S|/n$ e $|I|/n$. Chiaramente $s + i = 1$.

Push

Il primo stile è detto stile push: un nodo infetto sceglierà in modo casuale un vicino a cui inviare informazioni. il nodo destinazione verificherà se le informazioni ricevute sono più aggiornate e, in caso affermativo, cambierà il proprio valore. Un nodo quindi, se infetto, invierà sempre un messaggio ad un altro nodo, indipendentemente dal fatto che il nodo destinazione conosca o meno l'aggiornamento. Si può facilmente intuire che lo stile push è più efficace quando il numero di nodi infetti è basso. In uqesta situazione, infatti, il numero di nodi infetti tenderà a raddoppiare ad ogni round e dopo $O(\log_2 n)$ round il valore i si avvicinerà a $1/2$. Quando invece i supera $1/2$ la situazione cambia: se definiamo s_t il rapporto di nodi suscettibili al roud t, possiamo calcolare il numero atteso di nodi suscettibili al round $t+1$ come:

$$E(s_{t+1}) = s_t\Big(1 - \frac{1}{n}\Big)^{n(1-s_t)} \tag{2.1}$$

Nel dettaglio: un nodo rimane suscettibile se al round t era suscettibile (s_t) e non è stato contattato da nessuno dei nodi infetti ($1-1/n$ indica la probabilità che un nodo non contatti il nodo suscettibile, ripetuto per il numero di nodi infetti $n(1-s_t)$).

Questo valore può essere approssimato per n molto grandi a $s_t e^{-(1-s_t)}$. Come dimostra Pittel [on_spreading_a_rumor] il numero di round atteso $T(n)$ per informare tutti i nodi in una rete è

$$T(n) = \log_2 n + \ln n + O(1) = O(\log n) \tag{2.2}$$

dove $\log_2 n$ deriva dalla prima fase, $\ln n$ da quella finale, mentre la fase intermedia, molto veloce, dura un numero costante di cicli.

Pull

Il secondo stile analizzato è lo stile pull: i nodi chiedono informazioni ad altri nodi, inviando il proprio timestamp. Se questi ultimi possiedono un aggiornamento più recente, invieranno un messaggio in risposta che verrà utilizzato dal nodo iniziale per cambiare il proprio valore.

A differenza dello stile push, quest'ultimo risulta poco efficace quando i nodi infetti sono pochi. Il numero atteso di nodi non ancora infeti dopo $t+1$ round può essere espresso come:

$$E(s_{t+1}) = s_t \cdot s_t = s_t^2 \tag{2.3}$$

in quanto un nodo rimane non informato se nel round precedente era suscettibile ed ha contattato un nodo a sua volta suscettibile. Può accadere che un nodo infetto dovrà aspettare alcuni round prima di venir contattato, rendendo questi round inutili per lo scopo dell'algoritmo. Nonostante ciò, con alta probabilità dopo $O(logn)$ round metà dei nodi sarà infetta. La fase finale invece è molto più rapida in quanto, aumentando il numero di nodi infetti, aumenta la probabilità per un nodo suscettibile di contattare un nodo infetto.

Push-Pull

La soluzione migliore proposta si basa su una combinazione dei due stili precedenti. Lo stile push-pull lavora nel modo seguente: all'azzeramento del timer, un nodo invia un messaggio ad un altro nodo scelto tra i vicini in modo casuale, il quale controllerà il timestamp ed a seconda del risultato invierà una risposta oppure aggiornerà il proprio valore. E' più rapido in quanto sfrutta i punti di forza dei protocolli push e pull (nella fase iniziale sfrutterà il push, nella parte finale il pull). Karp [randomized_rumor_spreading] ha dimostrato che il numero atteso di round per infettare tutti i nodi è $O(\log \log n)$.

Riassumendo, il modello SI è efficace in quanto permette di distribuire su tutta la rete un aggiornamento, in quanto un nodo infetto continuerà (idealmente per sempre) ad inviare o ricevere messaggi. Nonostante ciò questo può rivelarsi un peso non indifferente per la rete in quanto questo modello prevede l'invio del database completo all'interno del messaggio e non il singolo aggiornamento. Se gli aggiornamenti in una rete sono rari, la maggior parte dei messaggi diventa inutile, perché i nodi continueranno a contattarne altri che già sono a conoscenza dell'aggiornamento.

2.2.3 Epidemie complesse

Il modello SIR viene introdotto per risolvere il problema della non terminazione del modello precedente e per aumentare l'efficienza. I nodi potranno assumere lo stato rimosso, ovvero nodi che conoscono l'aggiornamento ma non lo distribuiscono più. Come per le epidemie semplici, inizialmente i nodi sono tutti suscettibili: quando uno di questi viene a conoscenza di un aggiornamento, diventa infetto ed incomincia ad inviare messaggi agli altri nodi. Eventualmente questi nodi potranno "perdere" interesse nel distribuire il proprio aggiornamento, cambiando così il suo stato in rimosso. Quando nella rete non ci sarà più nessun nodo infetto, l'algoritmo termina. Il passaggio dallo stato infetto allo stato rimosso può essere influenzato dai seguenti fattori:

- **Come** (How):
  - **counter**: un nodo passerà allo stato rimosso dopo k contatti
  - **coin**: un nodo passerà allo stato rimosso con probabilità $1/k$
- **Quando** (When)
  - **feedback**: la valutazione avverrà quando un nodo contatta un altro nodo che era già a conoscenza dell'aggiornamento
  - **blind**: la valutazione avverrà ad ogni round

Si ottengono così quattro possibili combinazioni: *feedback/counter, blind/coin, feedback/coin, blind/counter*. Verranno analizzati *feedback/counter* e *blind/coin* utilizzando lo stile push, ma le considerazioni potranno essere applicate allo stesso modo anche per gli altri algoritmi.

Per confrontare i diversi protocolli che si basano sul modello SIR si utilizzano i seguenti criteri:

- **Residuo**: indica il numero di nodi ancora suscettibili al termine dell'algoritmo (viene indicato con s^*). Non è garantito infatti, come invece accade nel modello SI, che tutti i nodi verranno a conoscenza dell'aggiornamento. Può verificarsi una situazione in cui tutti i nodi si trovano nello stato suscettibile oppure rimosso, senza quindi aver ottenuto consistenza in tutta la rete.
- **Traffico**: indica il numero di messaggi inviati. Spesso si utilizza il traffico medio, definito come

$$m = \frac{\text{traffico totale}}{\text{numero di nodi}} \tag{2.4}$$

- **Ritardo**: può essere espresso come ritardo medio t_{avg}, ovvero la differenza tra il momento dell'infezione iniziale e l'arrivo di un aggiornamento, mediato sul numero di nodi, oppure come ritardo totale t_{max}, cioè il tempo necessario affinché l'ultimo nodo riceva l'aggiornamento.

Definiamo, come per il modello SI, s, i e r come la il rapporto di nodi suscettibili, infetti e rimossi rispetto al numero di nodi, in modo tale che $s + i + r = 1$.

L'andamento dei seguenti algoritmi può essere modellato attraverso l'utilizzo delle seguenti equazioni differenziali [The mathematics of infectious diseases]:

$$\begin{aligned} \frac{ds}{dt} &= -\beta is \\ \frac{di}{dt} &= \beta is - \gamma i \\ \frac{dr}{dt} &= \gamma i \end{aligned} \tag{2.5}$$

dove β rappresenta il tasso di contagio mentre γ, chiamato in epidemiologia tasso di recupero, è un valore che dipende da k e dal numero di nodi ancora suscettibili, più precisamente

$$\gamma = \frac{1}{k}(1 - s) \tag{2.6}$$

Possiamo utilizzare le prime due per risolvere il sistema di equazioni differenziali partendo dal loro rapporto e supponendo beta uguale a 1:

$$\begin{aligned}\frac{di}{ds} &= \frac{si - \frac{1}{k}(1-s)i}{-si} \\ &= \frac{\frac{\not{i}(ks-1+s)}{k}}{-\not{i}s} \\ &= \frac{1-ks-s}{ks} \\ &= \frac{1}{ks} - 1 - \frac{1}{k} \\ &= \frac{1}{ks} - \frac{1+k}{k}\end{aligned} \tag{2.7}$$

integrando si ottiene

$$s(i) = \frac{1}{k}\ln s - \frac{1+k}{k} + c \tag{2.8}$$

dove c è costante di integrazione che si può calcolare considerando che inizialmente la funzione è espressa come $i(1-1/n) = 1/n$ che tende a 0 per n molto grandi

$$c = \frac{k+1}{k} \tag{2.9}$$

che porta alla seguente soluzione

$$i(s) = \frac{k+1}{k}(1-s) + \frac{1}{k}\ln s \tag{2.10}$$

Questa equazione può essere utilizzata per calcolare s^* quando $i(s^*) = 0$

$$s^* = e^{(-k-1)(1-s^*)} \tag{2.11}$$

Il risultato è una funzione implicita su s^*, la quale mostra che il residuo diminuisce esponenzialmente all'aumentare di k.

È possibile notare inoltre che tutte le varianti dell'algoritmo condividono la stessa relazione tra traffico e residuo. Considerando ogni messaggio inviato ha probabilità pari a 1/n di contattare un nodo specifico, la probabilità di rimanere suscettibile dopo l'invio di m*n messaggi è pari a:

$$s(m*n) = \left(1 - \frac{1}{n}\right)^{nm} \tag{2.12}$$

che con n grandi può essere approssimato come $s = e^{-m}$. Come si può notare dalla tabella 1, il ritardo è l'unico parametro che distingue le varianti: osservando i dati si può notare che feedback/counter offre ritardo inferiore a parità di k.

2.2.4 Sed pulvinar placerat enim, a

2.3 Vivamus hendrerit imperdiet ex. Vivamus

3 Conclusioni

Bibliografia

Allegato A Titolo primo allegato

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sed nunc orci. Aliquam nec nisl vitae sapien pulvinar dictum quis non urna. Suspendisse at dui a erat aliquam vestibulum. Quisque ultrices pellentesque pellentesque. Pellentesque egestas quam sed blandit tempus. Sed congue nec risus posuere euismod. Maecenas ut lacus id mauris sagittis egestas a eu dui. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque at ultrices tellus. Ut eu purus eget sem iaculis ultricies sed non lorem. Curabitur gravida dui eget ex vestibulum venenatis. Phasellus gravida tellus velit, non eleifend justo lobortis eget.

A.1 Titolo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sed nunc orci. Aliquam nec nisl vitae sapien pulvinar dictum quis non urna. Suspendisse at dui a erat aliquam vestibulum. Quisque ultrices pellentesque pellentesque. Pellentesque egestas quam sed blandit tempus. Sed congue nec risus posuere euismod. Maecenas ut lacus id mauris sagittis egestas a eu dui. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque at ultrices tellus. Ut eu purus eget sem iaculis ultricies sed non lorem. Curabitur gravida dui eget ex vestibulum venenatis. Phasellus gravida tellus velit, non eleifend justo lobortis eget.

A.1.1 Sottotitolo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sed nunc orci. Aliquam nec nisl vitae sapien pulvinar dictum quis non urna. Suspendisse at dui a erat aliquam vestibulum. Quisque ultrices pellentesque pellentesque. Pellentesque egestas quam sed blandit tempus. Sed congue nec risus posuere euismod. Maecenas ut lacus id mauris sagittis egestas a eu dui. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque at ultrices tellus. Ut eu purus eget sem iaculis ultricies sed non lorem. Curabitur gravida dui eget ex vestibulum venenatis. Phasellus gravida tellus velit, non eleifend justo lobortis eget.

Allegato B Titolo secondo allegato

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sed nunc orci. Aliquam nec nisl vitae sapien pulvinar dictum quis non urna. Suspendisse at dui a erat aliquam vestibulum. Quisque ultrices pellentesque pellentesque. Pellentesque egestas quam sed blandit tempus. Sed congue nec risus posuere euismod. Maecenas ut lacus id mauris sagittis egestas a eu dui. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque at ultrices tellus. Ut eu purus eget sem iaculis ultricies sed non lorem. Curabitur gravida dui eget ex vestibulum venenatis. Phasellus gravida tellus velit, non eleifend justo lobortis eget.

B.1 Titolo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sed nunc orci. Aliquam nec nisl vitae sapien pulvinar dictum quis non urna. Suspendisse at dui a erat aliquam vestibulum. Quisque ultrices pellentesque pellentesque. Pellentesque egestas quam sed blandit tempus. Sed congue nec risus posuere euismod. Maecenas ut lacus id mauris sagittis egestas a eu dui. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque at ultrices tellus. Ut eu purus eget sem iaculis ultricies sed non lorem. Curabitur gravida dui eget ex vestibulum venenatis. Phasellus gravida tellus velit, non eleifend justo lobortis eget.

B.1.1 Sottotitolo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sed nunc orci. Aliquam nec nisl vitae sapien pulvinar dictum quis non urna. Suspendisse at dui a erat aliquam vestibulum. Quisque ultrices pellentesque pellentesque. Pellentesque egestas quam sed blandit tempus. Sed congue nec risus posuere euismod. Maecenas ut lacus id mauris sagittis egestas a eu dui. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque at ultrices tellus. Ut eu purus eget sem iaculis ultricies sed non lorem. Curabitur gravida dui eget ex vestibulum venenatis. Phasellus gravida tellus velit, non eleifend justo lobortis eget.