

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

DOCUMENTAZIONE PROGETTO A.A. 2020/2021

ANTONIO PATIMO MAT:578464

STRUTTURE DATI UTILIZZATE

I dati posseduti da un peer sono organizzati in un registro generale che a sua volta è composto da una lista di registri giornalieri ordinata per data, ma anche da una serie di altre liste utili durante lo scambio di informazioni con gli altri Peer, quali una lista di date necessarie e una lista di registri richiesti.

I registri giornalieri tengono traccia dei tamponi effettuati e dei nuovi casi registrati in tale data rispettivamente nelle variabili `tamponi` e `nuovi_casi`. Ho scelto di rappresentare i dati in forma aggregata e non di singole entry per diminuire il numero complessivo di dati da scambiare tra i peer per ottenere una maggiore efficienza di trasferimento ed elaborazione. Un registro giornaliero inoltre possiede un valore booleano per determinare se quel registro è completo oppure no.

Per quanto riguarda la gestione delle aggregazioni, ogni peer possiede una lista delle aggregazioni da esso possedute. Ogni aggregazione possiede tre stringhe: `aggr`, `type`, `period`, tramite le quali è possibile identificare univocamente un'aggregazione. Ogni aggregazione possiede inoltre un intero che rappresenta la dimensione del risultato (1 per il totale, variabile per la variazione), un vettore di interi che rappresenta i valori del risultato e un vettore di stringhe che contiene i periodi rispetto ai quali i risultati fanno riferimento. Per esempio, un'operazione del tipo `get variazione tampone 01:12:2020-03:12:2020` darà un risultato della forma: `01:12:2020-02:12:2020: 5, 02:12:2020-03:12:2020: 7`.

In fine è presente anche una struttura dati `IOController` che contiene una serie di informazioni utili alla comunicazione con gli altri peer, per esempio un array contenente le porte dei neighbor peer e un array contenente le porte dei peer da contattare in seguito ad una operazione di flooding.

SCAMBIO DELLE INFORMAZIONI TRA PEER

Quando un peer riceve una richiesta per un'aggregazione da parte dell'utente, dopo aver verificato la validità del periodo, crea una lista di date necessarie utili a calcolare tale aggregazione. Per esempio, se l'aggregazione richiesta è `get totale tampone 01:12:2020-03:12:2020` la lista di date necessarie conterrà 3 elementi aventi data rispettivamente `01/12/2020`, `02/12/2020`, `03/12/2020` di tipo `struct tm`.

Il Peer verificherà quindi la presenza di queste date nel registro generale ed in particolare se nel registro generale esiste un registro giornaliero avente una di queste date e il campo completo ad 1 il peer provvederà a rimuovere dalla lista di date necessarie la data corrispondente, in quanto il peer possiede già tutte le informazioni circa quella data. In questo modo il Peer inoltrerà una richiesta di date che gli sono effettivamente necessarie, in modo da ridurre in maniera significativa lo scambio di dati tra peer (Supponiamo che un utente richieda il totale dei tamponi tra il `1/12/2020` e il `31/12/2020` se il peer possiede tutti i registri giornalieri completi dal `1/12/2020` al `28/12/2020` il peer inoltrerà solamente la richiesta per le date dal `29` al `31`).

Se un Peer dopo avere verificato la presenza dei dati "svuota" la lista delle date necessarie allora esso possiede tutti i dati necessari per calcolare l'aggregazione richiesta, altrimenti richiede ai neighbor se essi possiedono tale aggregazione. In caso contrario il Peer invia ai neighbor un

messaggio di FLOOD, un messaggio contenente l'ID della richiesta e la lista delle date necessarie al Peer.

Quando un Peer riceve una richiesta di flood riceve anche la lista delle date necessarie al requester e verifica la presenza di tali date all'interno del suo registro generale. Se trova una data richiesta si aggiunge alla lista dei peer da contattare ed inoltre se il registro giornaliero trovato ha la variabile completo uguale a 1 elimina la data dalle date necessarie perché quel peer possiede già tutte le informazioni per quella data. In questo modo ad ogni inoltro della richiesta di flood la lista delle date necessarie si ridurrà in modo significativo andando a ridurre lo scambio di dati nella rete. Così facendo è molto probabile che le date richieste vengano trovate prima che il messaggio raggiunga tutti i peer della rete in quanto basterà trovare quei peer che hanno i registri completi per le date richieste.

Se un peer verificando la presenza dei dati "svuota" la lista delle date necessarie non inoltrerà più la richiesta e restituirà la lista dei peer da contattare (che in quel caso sarà composta solo dal peer stesso). Altrimenti inoltra la richiesta con la lista delle date necessarie aggiornata ai suoi neighbor escluso il peer da cui è arrivata la richiesta. Il peer in questione avrà una lista dei peer da contattare che sarà aggiornata ogni volta che gli arriva una risposta da un peer a cui aveva inoltrato la richiesta. Quando saranno arrivate tutte le risposte attese, il peer inoltrerà la risposta (contenente la lista dei peer da contattare) con un messaggio FLOORE al peer che gli aveva a sua volta inoltrato la richiesta. Quando il requester riceverà la risposta da parte di tutti i suoi neighbor avendo la lista dei peer da contattare potrà inviare loro la richiesta dei registri relativi alle date necessarie tramite un messaggio REQRE. I peer così contattati creeranno una lista di registri giornalieri con i valori richiesti e la inoltreranno indietro al requester. Quando il requester avrà ricevuto i registri richiesti dai peer da contattare potrà modificare via via il suo registro generale in base ai valori ricevuti. In fine il requester setterà la variabile completo a 1 per tutti i registri compresi nel periodo richiesto, in quanto dopo una richiesta di flood egli possiederà sicuramente tutti i dati relativi a quelle date presenti all'interno della rete.

Per quanto riguarda il formato dei messaggi scambiati, per esempio nell'invio da parte di un peer dei registri giornalieri richiesti al requester, si invia per prima cosa il numero dei registri da inviare utilizzando il tipo uint16_t. Successivamente previa comunicazione della lunghezza del messaggio, si invia una stringa contenente le informazioni sotto forma di interi separati da spazi. Per esempio, l'invio di un registro giornaliero con data 01/12/2020 avente 5 tamponi 6 nuovi casi e il valore di completo a 1 verrà inviato come stringa nel formato: "1 12 2020 5 6 1"

TOPOLOGIA DELLA RETE

La lista dei peer connessi al network è gestita dal DS come una lista circolare ordinata per numero di porta. In particolare, quando un peer entra nella rete sarà sufficiente inserirlo nella posizione corretta della lista e quando un peer si disconnetterà dal network sarà sufficiente rimuoverlo da tale lista. Questa scelta di integrazione dei peer permette di organizzare i peer per località del numero di porta ed impedisce la formazione di zone separate della rete. In particolare, ogni peer ha 2 neighbor peer, quello precedente nella lista e quello successivo. Questa scelta porta la rete ad assumere una topologia ad anello ma ci tengo a sottolineare come il progetto sia del tutto indipendente dalla topologia della rete. Infatti, la totalità delle operazioni svolte dai peer è completamente indipendente dal numero di neighbor e dalla topologia della rete. Implementando diversamente la funzione che stabilisce l'insieme dei vicini di un particolare peer potremmo dare alla nostra rete la topologia desiderata.