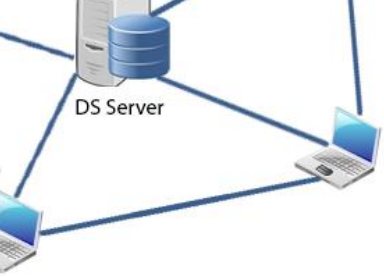
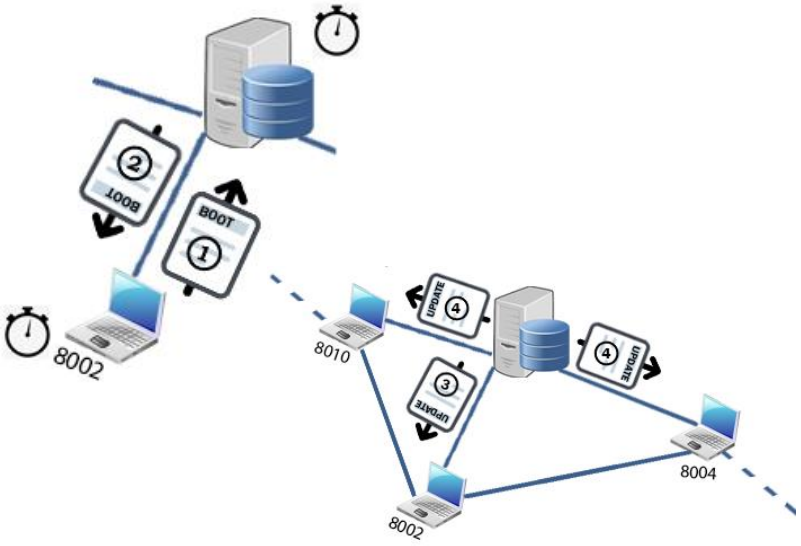


Progetto di Reti Informatiche

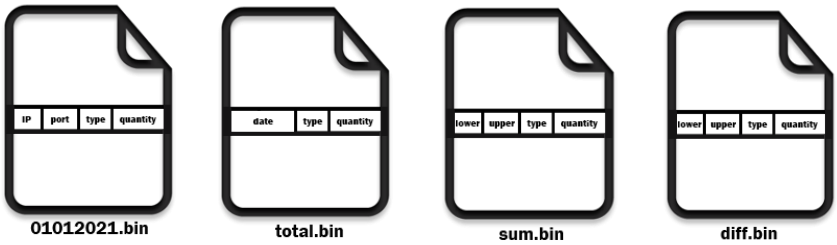
Terranova Franco

Boot E Update

Il DS server dispone di una lista circolare, mantenuta ordinata in base al numero di porta, nella quale memorizza le informazioni sui peer una volta effettuato il boot. Inserito il peer nella giusta posizione all'interno della lista, i due vicini assegnati al peer saranno l'elemento precedente e l'elemento successivo nella lista, realizzando quindi un meccanismo di prossimità basato sul numero di porta. L'inserimento del peer nella lista causa quindi sempre, dopo l'inserimento in lista, l'aggiornamento dei vicini di altri peer, integrando così correttamente il peer nel network (a meno che questo non sia il primo peer ad entrare nel network). Il registro di prossimità dei peer implementato mediante coda circolare evita eventuali problemi di clustering e di isolamento.



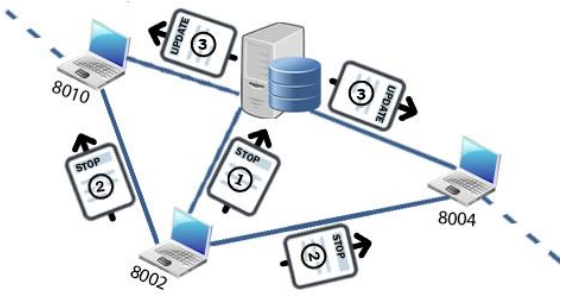
Registro giornaliero



Il registro giornaliero del peer viene mantenuto in memoria. Alla disconnessione il peer somma il numero di tamponi e di nuovi casi registrati ottenendo così due entry che vengono memorizzate all'interno del file relativo alla data odierna insieme all'indirizzo IP e alla porta del peer. Si aggiungono queste due ultime informazioni poiché all'interno dello stesso file verranno memorizzate anche le entry (relative a quel giorno) di altri peer. Tra le quali, quindi, quelle dei vicini quando questi invocheranno la stop e le entry di altri peer ottenute attraverso il flooding. Il compattamento dei valori registrati permetterà inoltre di ridurre la quantità di traffico nel network.
Se il file già esiste (il peer è già entrato in esecuzione nella stessa giornata) i dati vengono opportunatamente aggiornati. Ogni peer dispone poi di altri file (total.bin, sum.bin, aggr.bin) necessari per la memorizzazione di informazioni calcolate durante le aggregazioni. Il file dei totali permette di evitare calcoli successivi inutili.

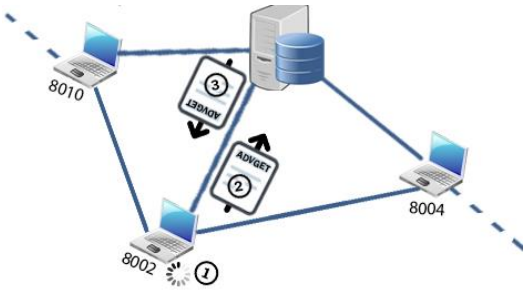
Stop

Quando il peer richiede la disconnessione dal network memorizza le proprie entry odierne su file e le invia ai suoi vicini. Il DS server aggiorna opportunatamente il registro di prossimità e fornisce i nuovi neighbors ai suoi vicini. Poichè la prossimità è ottenuta mediante l'utilizzo di una lista circolare la disconnessione del peer non potrà mai causare l'isolamento di una parte del network.

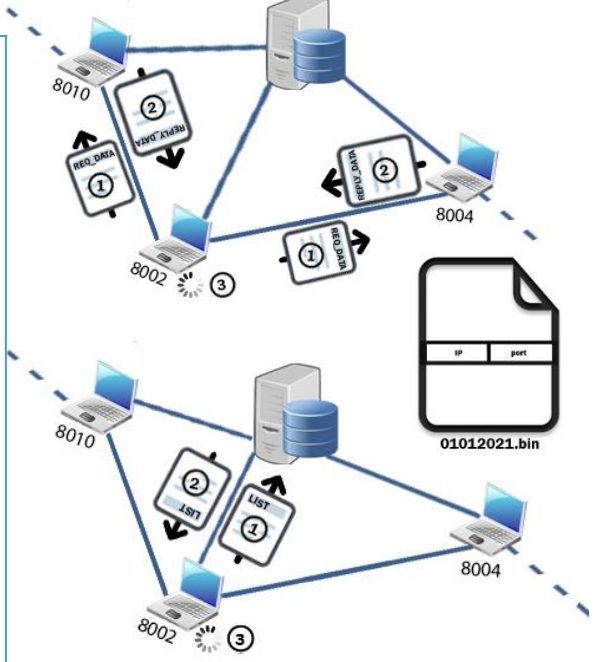


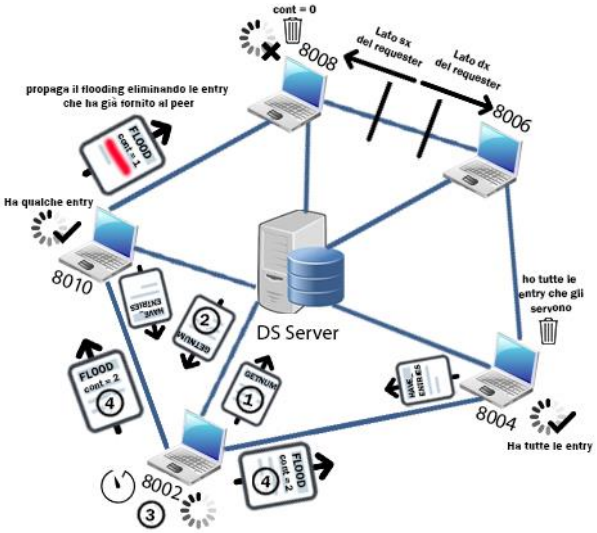
Get

(1) Alla richiesta di una elaborazione il peer verifica se ha già calcolato il dato richiesto, consultando rispettivamente i file sum.bin e diff.bin.
(2) Se non ha già calcolato il dato invia innanzitutto una notifica al DS server informandolo che sta per calcolare un dato aggregato. Durante il calcolo dell'aggregazione il DS server non permetterà a nessun peer di terminare (stop), di entrare a far parte del network (boot), non potrà esso stesso terminare causando la terminazione di tutti i peer del network (esc) e non permetterà ad altri peer di effettuare il calcolo di un'aggregazione.

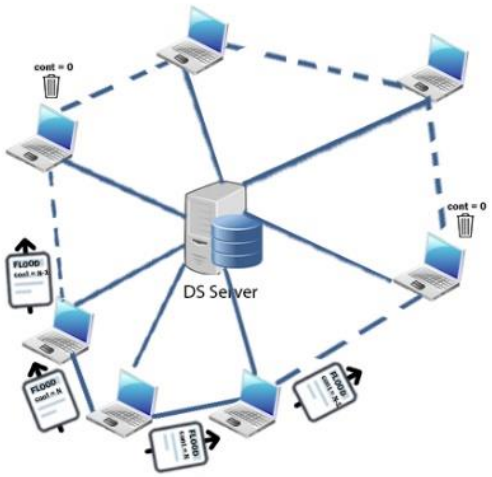


In questo modo, durante la richiesta del dato ai neighbors e durante l'eventuale flooding l'associazione tra neighbors rimane statica insieme al numero dei peer costituenti il network (necessario successivamente per caricare il contatore del messaggio) e inoltre in questo modo durante un flooding non potrà verificarsi un altro flooding o la richiesta di dati calcolati tra vicini.
(3) Il peer chiede successivamente ai suoi neighbors se hanno a disposizione il dato.
(4) Se questi non hanno a disposizione il dato chiede al DS server la lista dei peer connessi nei giorni coinvolti nel calcolo dell'aggregazione così da determinare se ha disposizione tutte le entry.
Il DS server oltre alla lista circolare mantiene una lista degli accessi in cui memorizza indirizzo IP e porta dei peer che sono entrati a far parte del network in quella giornata e quindi dei peer che hanno registrato almeno una entry. Questa a differenza della lista circolare viene memorizzata su file e risulta essere molto meno dinamica, per esempio se un peer lascia il network viene rimosso dalla lista circolare ma non dalla lista degli accessi di quella giornata.
Ricevuta la lista, il peer è in grado di determinare se ha a disposizione tutte le entry.
Se non ha a disposizione tutte le entry richiede al DS server il numero dei peer attualmente connessi al network così da poter impostare correttamente il valore del contatore nel messaggio inviato in flooding e realizzare un meccanismo di **query ad orizzonte limitato**. In questo modo riduciamo l'overhead, nel caso in cui abbiamo un elevato numero di peer connessi al network (scalabilità), a discapito però del fatto che una data entry possa non venire mai trovata. Nel caso in cui il numero di peer connessi al network non è elevato non si utilizza il meccanismo di query ad orizzonte limitato ma si utilizza comunque il campo contatore, avendo a disposizione il numero di peer attualmente connessi al network, per poter indicare ai peer quando smettere di propagare il flooding. Sfruttando infatti la disposizione ad anello avremo quindi che l'ultimo peer del lato sinistro del requester porterà il contatore a 0, non propagando più la query e lo stesso valrà per il suo vicino che fa invece parte del lato destro del requester. In questo modo un peer non potrà ricevere più volte lo stesso messaggio inviato in flooding e di conseguenza non sarà necessario che questo memorizzi informazioni di stato sui flooding già gestiti.
Nella mia implementazione il peer non sfrutta l'anello inviando un solo messaggio in flooding bensì invia il messaggio verso entrambi i lati (neighbor destro e sinistro) così da ridurre il tempo necessario per entrare in possesso delle entry.
Se un peer che riceve il messaggio ha alcune delle entry specificate lo comunica all'indietro fino al requester e avverrà il trasferimento. Se ha a disposizione tutte le entry, smette inoltre di propagare il flooding, altrimenti rimuove dal messaggio le entry che ha fornito al peer e continua a propagare il flooding. Poiché il requester invia il messaggio verso entrambi i lati potrebbe ricevere più volte le stesse entry e se riceve più volte la stessa entry allora non memorizza la copia.
Grazie al valore del contatore il peer imposta un timer, e rimane in attesa di eventuali contatti da peer nel network, per un intervallo di tempo proporzionale al numero di peer coinvolti nel flooding.





Query ad orizzonte limitato



Offline Get

E' stato previsto anche un meccanismo che permetta di calcolare un'aggregazione senza essere connessi al network. Poiché in questi casi non si ha la possibilità di consultare il DS server per ottenere la lista dei peer connessi in un dato giorno si fa uso dei soli file total.bin, sum.bin, diff.bin mantenuti consistenti durante il calcolo delle aggregazioni in modalità online. Se l'aggregazione può essere calcolata mediante questi file il calcolo viene effettuato, altrimenti si invita l'utente a connettersi al network.

Formato dei messaggi

Tipo	sp	Codice di stato	\n	riga di intestazione
nome campo:	sp	valore	\n	corpo
...				
nome campo:	sp	valore	\n	

Lower & Upper Bound

Se il lower bound e/o l'upper bound non vengono specificati viene selezionata una finestra temporale per il peer. La finestra temporale viene determinata in funzione del file relativo alla data più remota/più recente in cui il peer ha registrato qualche dato, determinato scorrendo la cartella dei registri del peer. I file non verranno coinvolti se riguardano registers non ancora chiusi.
Quando un registro viene chiuso e memorizzato su file e non dovrà più essere modificato, vengono inoltre rimossi i permessi di scrittura sul file.