Università di Pisa

Corso di laurea specialistica in informatica per l'economia e l'azienda

Real Time Analyzer per plugin http di nProbe.

Sommario

Introduzione

Lo scopo di questo progetto è quello di realizzare un'applicazione che fornisca una panoramica in tempo reale dell'andamento del traffico http di una rete IP.
L'oggetto dell'analisi è il traffico HTTP in termini di byte, numero dei flussi e tempo di utilizzo della risorsa (tempo totale dei flussi) in transito sul router, che viene catturato dalla sonda, sia dal punto di vista degli host della rete, che dal punto di vista dei server web contattati dagli host.
Trattandosi di un applicazione che dovrà funzionare su PC server, generalmente sprovvisti di interfaccia grafica e più facilmente raggiungibili da remoto, è stato scelto di implementare l'interfaccia grafica in jsp+Servlet su tomcat 6 application server ed è quindi facilmente accessibile da un qualsiasi web browser.
L'applicazione RTA, sfrutta il dump del plugin HTTP di nProbe come fonte dei dati per l'analisi. nProbe li salva sul disco al termine di ogni minuto, e in seguito vengono caricati sul Database per renderli disponibili per le analisi delle Servlet.

Struttura dell'applicazione

RTA è suddiviso due parti, la prima è LogServer, un programma stand-alone, da installare e lanciare sul router dove è in esecuzione nProbe, la seconda è un insieme di Servlet e pagine jsp la visualizzazione dei risultati delle analisi, che possono anche essere dislocate su un server diverso.

LogServer

LogServer è costituito da un'unica classe LogServer.java.
La sua funzione è quella di leggere i file del dump di nprobe, quando questi vengono creati, e scriverli sul Database.
I campi dei log che interessano ai fini delle analisi sono:
timestamp : formato gg/mm/aaaa hh:mm:ss
client
server

flowid: Flow Hash

bytes

time: calcolato come tend -tstart

Server Latency

Su LogServer è possibile settare la variabile “hours” che stabilisce la durata massima della finestra temporale dei log su DB. Al termine di queste ore i dati obsoleti vengono cancellati per non sovraccaricare il DB con informazioni non più rilevanti.

Pagine JSP+Servlet

Le Servlet utilizzate da RTA sono 4: HostsAnalysis ServersAnalysis HostDetail e ServletChart.

Le Prime 3 eseguono delle query sul DB e stampano i risultati su una tabella nella pagina web, invece ServletChart restituisce un immagine con il grafico dell’andamento temporale del traffico http in termini di byte a seconda della pagina da cui viene chiamata.

HostsAnalysis.java:

Stampa la tabella con le analisi degli host in ordine di bytes ricevuti:

Host Server bytes FTime Nflow

FTime indica il tempo totale di tutti I flussi dell’host

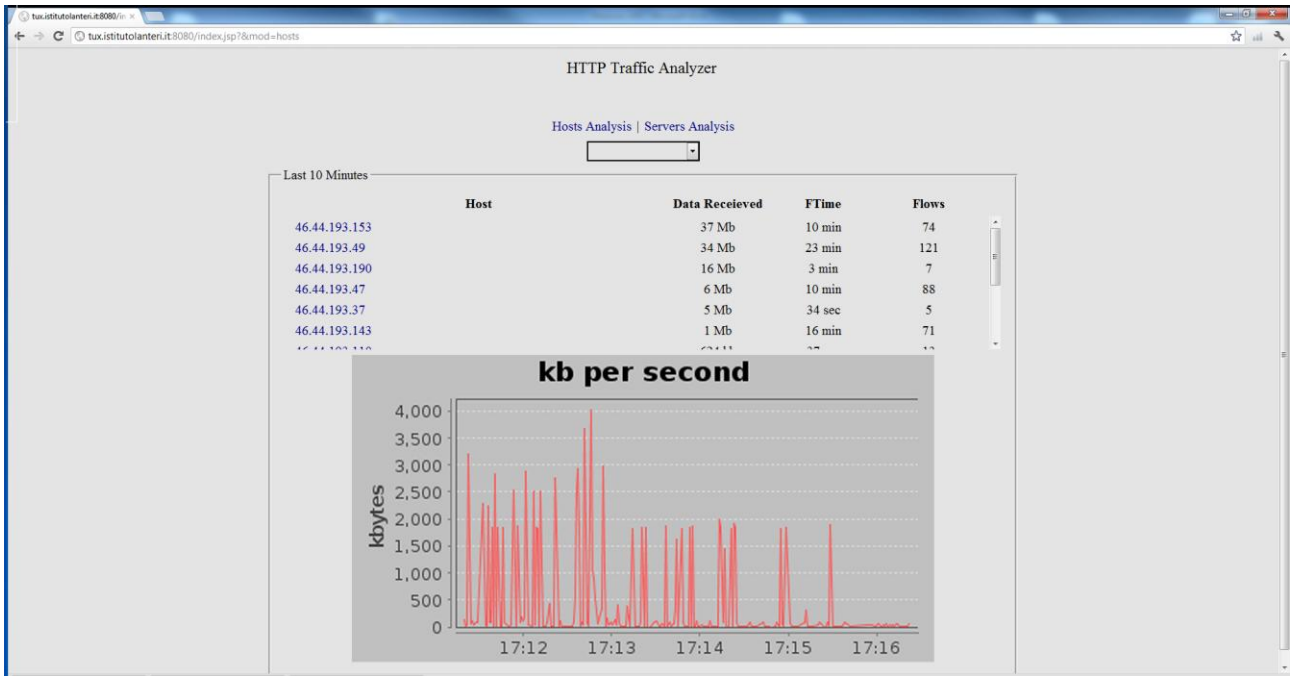
NFlows, il numero dei diversi flussi

Query:

```
SELECT client, SUM(bytes) AS nbytes, SUM(time) AS time
FROM pages
WHERE timestamp>'"STARTTIME"'
GROUP BY 1, ORDER BY 2
```

Per ogni client:

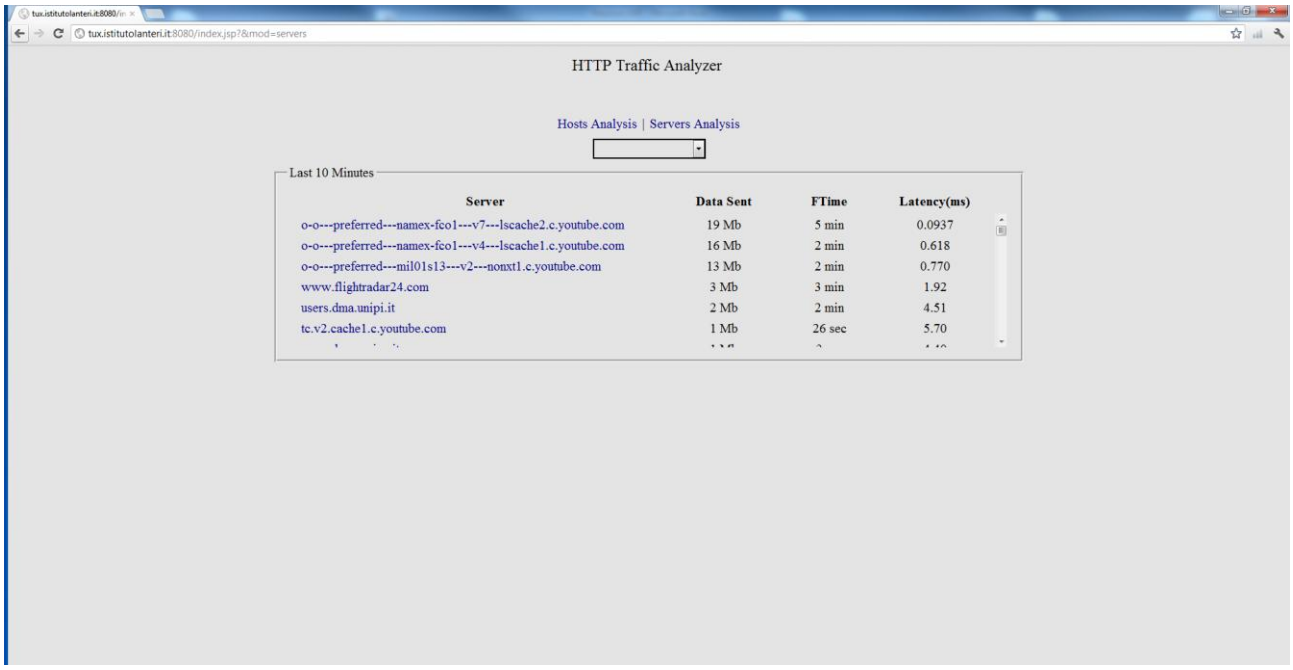
```
SELECT count(*) AS flows
FROM (SELECT flowid, count(*)
        FROM pages
        WHERE client='"CLIENT"' AND timestamp >'"STARTIME"'
        GROUP BY 1) AS g
```



ServersAnalysis.java:

Stampa la tabella con le analisi dei server per le richieste complessive di tutti gli host in ordine di bytes trasmessi:

Server bytes FTime LatenzaMedia



Ad ogni campo server dei record è associato un collegamento ipertestuale che punta alla pagina con il grafico dell'andamento temporale dei bytes del relativo serve.

HostsDetails.java:

Stampa la tabella con le analisi di tutti i server del singolo host in ordine di bytes ricevuti:

Server bytes FTime LatenzaMedia

ServletChart.java:

Questa servlet utilizza le librerie JFreeChart per disegnare il grafico dell'andamento temporale in ordine di kb del traffico http.

Con parametri diversi viene chiamata da tutte le pagine jsp che includono varie servlet dell'applicazione per stamparne il relativo grafico.

Eseguita la giusta query e ottenuti i dati dal DB (kbyte, time) crea un oggetto di tipo TimeSeries con i dati da plottare che viene passato come parametro al metodo ChartFactory.createTimeSeriesChart() per costruire il grafico. Infine viene restituito un content type image/png alla pagina jsp che l'ha chiamata.

La granularità minima dell'andamento temporale dei grafici è il secondo.