Torino, 11/12/2006

Relazione sulla tesina di "Linguaggi e Traduttori"

Alberto Realis-Luc – Matr. 142119
Leandro Dipietro – Matr. 134841

Obbiettivi del progetto svolto

Dato un un file di log degli accessi al web server Apache nel classico formato di default:

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\""
combined
```

In cui i dati elencati sono:

- **%h** - L'indirizzo IP del client
- **%l** - Il nome dell'utente remoto, se fornito da un identd lookup
- **%u** - Il nome dell'utente eventualmente autenticato sul server
- **%t** - Date e ora della richiesta. Nel formato: `[gg/Mes/aaaa:hh:mm:ss +fuso]`
- **%r** - La prima riga della richiesta HTTP, che contiene all'inizio il metodo usato (GET, POST...)
- **%s** - Lo status code HTTP della risposta
- **%b** - Le dimensioni in byte del file trasferito
- **%{Header}i** - Un header http nella richiesta del client (nel nostro caso: %{User-Agent}i %{Referer}i)

Ogni riga del file access.log descrive un accesso e potrebbe essere ad esempio:

```
217.148.96.21 - - [07/Nov/2002:14:27:05 +0100] "GET /gfx/morecoresisproject.jpg
HTTP/1.1" 304 - "http://www.openskills.info/" "Opera/6.02 (Linux 2.4.17-GANDALF
i686; U) [en]"
```

In questo esempio si nota che i nomi utenti e la dimensione in Byte sono sostituite da un trattino ciò significa che i nomi utenti non sono stati specificati (come nella stragrande maggioranza dei casi) e che la dimensione è nulla probabilmente perché in questo caso non è stato necessario trasferire alcun file.
Il nostro progetto ha l'obbiettivo di filtrare le informazioni contenute in un dato file di log, nel formato descritto sopra, secondo certe condizioni specificate dall'utente in un file.

Funzionamento

LogAnalizer permette di scandire l'intero log riportando in una tabella nel file risultati.html tutti gli accessi che rispettano tutte le condizioni specificate dall'utente in un file. E' possibile impostare al massimo una condizione su ogni campo, in qualunque ordine. In caso di condizioni ripetute sullo stesso campo verrà presa in considerazione solo l'ultima di esse.
Il primo parametro che deve essere passato su linea di comando è il nome del file di log da analizzare, mentre il secondo (opzionale)è il nome del file contenente la query da eseguire sul log.
Il progetto utilizza due scanner e due parser che si occupano di scandire e interpretare il file di log e le condizioni richieste.
Prima vengono utilizzati QueryScanner e QueryParser per acquisire tutte le condizioni desiderate dall'utente, quindi vengono avviati LogScanner e LogParser che si occupano di filtrare gli accessi riga per riga del log selezionando e scrivendo direttamente su risultati.html gli accessi che rispettano tutte le condizioni precedentemente acquisite.

Formato delle condizioni

Ogni condizione è formata da:

- **una parla chiave** che identifica su quale campo viene applicata
- **un carattere di relazione** che a senconda dei casi può essere: `= < > :`
  il carattere **=** indica l'esatta corrispondenza con un valore od un intervallo di valori
  i caratteri **<** e **>** indicano valori minori o maggiori rispetto a quello specificato non possono essere usati per valori testuali
  il carattere **:** permette di specificare su qualsiasi campo un espressione regolare, nel caso in cui si intenda selezionare tutti i valori che iniziano come specificato nell'espressione regolare occorrerà terminare quest'ultima con `.*` per indicare che la stringa può terminare con qualunque altra sequenza di caratteri.
- **un valore od un intervallo di valori** le stringhe e le espressioni regolari devono essere comprese tra apici, gli indirizzi ip devono essere specificati tra parentesi tonde mentre le date devono essere tra parentesi quadre.

Condizione sull'indirizzo IP

Questa condizione viene dichiarata con la keyword: `ip`
Segue il carattere di relazione che può essere: `= < > :`
Infine si ha, contenuto tra parentesi tonde, un l'indirizzo IP o un intervallo di indirizzi IP nel formato: ipStart-ipStop
Il simbolo di relazione = può essere usato per un IP singolo oppure per in intervallo, mentre i simboli < > possono essere usati solo per un ip singolo. Invece il simbolo **:** permette di inserire un'espressione regolare racchiusa tra apici.

Esempi:
`ip=(127.0.0.1)` Seleziona tutti gli accessi avvenuti dall'IP: 127.0.0.1
`ip>(192.168.1.1)` Seleziona tutti gli accessi avvenuti da IP uguali o consecutivi a: 192.168.1.1
`ip<(192.168.1.100)` Seleziona tutti gli accessi avvenuti da IP uguali o precedenti a: 192.168.1.100
`ip=(192.168.1.1-192.168.1.100)` Seleziona tutti gli accessi avvenuti da IP uguali o compresi tra: 192.168.1.1 e 192.168.1.100
`ip=(192.168.1.100-192.168.1.1)` Come prima, funziona anche con l'intervallo invertito.
`ip:'192.168.1.1[13]'` Seleziona tutti gli accessi eseguti dagli indirizzi IP 192.168.1.11 e 192.168.1.13.

Condizioni su utente e utente remoto

Queste condizioni si richiamano con le keyword: `utente` e `utenteremoto`
Si può usare il carattere di relazione = per indicare l'esatta corrispondenza con il nome specificato oppure il carattere : per specifiacre un espressione regolare
Il nome utente specificato o l'espressione regolare devono essere contenute tra apici: '
Esempi:
`utente='paperino'` Seleziona tutti gli accessi fatti dall'utente paperino
`utenteremoto='pluto'` Seleziona tutti gli accessi fatti dall'utente remoto pluto
`utente=''` Seleziona tutti gli accessi in cui non è stato specificato un nome utente

Condizione sul tempo

Con la keyword `tempo` è possibile specificare una condizione sulla data e ora degli accessi che ci interessa filtrare. Questa condizione funziona esattamente come quella sull'IP con la differenza che al posto degli IP specificati tra parentesi tonde ora dobbiamo specificare delle date e ore racchiuse tra parentesi quadre. Le tempistiche vanno specificate in questo formato: `gg/Mes/aaaa:hh:mm:ss` Il mese può essere dato dalle prime tre lettere (la prima maiuscola) del suo nome in italiano o in inglese oppure può essere dato in numero con o senza zero davanti. E' possibile omettere i secondi o i minuti e secondi oppure scrivere solo la data senza specificare l'orario.
Esempi:
`tempo=[21/Nov/2006:13:37:01]` Accessi avvenuti esattamente il 21/11/2006 alle 13:37:01
`tempo=[21/11/2006:13:37:01-01/Dec/2006:06:37:30]` Accessi avvenuti dalle 13:37:01 del 21/11/2006 alle 6:37:30 del 1/12/2006
`tempo>[21/Nov/2006:13:37:01]` Accessi avvenuti dopo le 13:37:01 del 21/11/2006
`tempo<[1/12/2006:6:37:30]` Accessi avvenuti prima dalle 6:37:30 del 1/12/2006
`tempo=[21/11/2006]` Accessi avvenuti durante il giorno 21/11/2006
`tempo=[1/Dic/2006:6]` Accessi avvenuti durante le ore 6 del giomo 1/12/2006
`tempo=[1/Dic/2006:6:38]` Accessi avvenuti dalle 6:38:00 alle 6:38:59 del giorno 1/12/2006

Condizioni su richiesta, refer e useragent

Con le keyword `richiesta`, `refer` e `useragent` è possibile imporre condizioni su questi dati si accetta solo la relazione = e la stringa indicata dovrà essere racchiusa tra apici. Verranno cercate tutte le stringhe che iniziano con il testo indicato nella condizione.
Esempi:
`richiesta='POST'` Seleziona tutti gli accessi che hanno fatto una richiesta HTTP di tipo POST
`refer='http://127.0.0.1/phpmyadmin/'` Seleziona tutti gli accessi con refer che parte da http://127.0.0.1/phpmyadmin/
`useragent='Mozilla'` Seleziona tutti gli accessi in è stato riconosciuto Mozilla come user agent o in cui la stringa dello user agent inizia con Mozilla.

Condizioni su status code e dimensione file

Con le keyword `status` e `bytes` è possibile impostare condizioni sullo status code HTTP di ogni accesso oppure sulla dimensione del file corrispondente trasferito. Queste condizioni lavorano esattamente come quella sull'IP con la differenza che al posto degli IP racchiusi tra tonde basta inserire valori numerici corrispondenti agli status code o dimensioni in bytes desiderati.
Esempi:
`status=200` Seleziona tutti gli accessi che hanno 200 come status code HTTP
`bytes=500-2000` Seleziona tutti gli accessi che hanno permesso il trasferimento di un file di dimensione compresa tra i 500 e i 2000 Bytes.

Compilare il progetto

Per compilare il progetto occorre:

1. Generare lo scanner del log con il comando: `jflex LogScanner.jflex`
2. Generare lo scanner delle condizioni con il comando: `jflex QueryScanner.jflex`
3. Generare il parser del log: `cup -parser LogParser -symbols LogSym LogParser.cup`
4. Generare il parser delle condizioni: `cup -parser QueryParser -symbols QuerySym QueryParser.cup`
5. Compilare il tutto: `javac *.java`

Esempio di esecuzione

Eseguendo il programma con il comando:

```
java LogAnalizer access.log query
```

Con il file query contenente:

```
ip=(127.0.0.1-192.168.1.13)
richiesta='POST'
tempo=[30/11/2006]
status=200
bytes<1000
useragent='Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)'
```

Utilizzando lo stesso file di log che abbiamo usato nelle varie prove si ottiene un solo accesso:

Indirizzo IP	**Data e Ora**	**Richiesta HTTP**	**Status**	**Bytes**	**Refer**	**Useragent**
192.168.1.13	30/Nov/2006 19:8:1	POST /phpBB2/install/install.php HTTP/1.1	200	650	http://192.168.1.11/phpBB2/install/install.php	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)