Graphics Protocol: un semplice sistema per la creazione di grafici

Indice

1 Introduzione

Il software in oggetto è un semplice programma per la creazione automatica di grafici, tale programma si appoggia al software nprobe reperibile a questo sito: www.ntop.org/category/nprobe/.
Il programma legge dei file che vengono dati in input, li elabora e alla fine crea una pagina web contenente il grafico.

1.1 Graphics Protocol v.0.1

Il software è compatibile con la maggioranza dei sistemi operativi che supportano la JVM; esso si avvia tramite il comando:
$ java -jar gprotocol
Il software può ricevere in input diverse opzioni, esse sono:

1) --help | -help
2) --db=<path database>
3) --log=<path log file>
4) --file=<file da analizzare>
5) --protocols=<file contenente i nomi dei protocolli>
6) --html=<directory o nome del file di output dell'html>
7) --start=<minuto/ora/giorno/mese/anno che indica da dove far partire il grafico>
8) --end=<minuto/ora/giorno/mese/anno che indica dove far terminare il grafico>

Tali opzioni saranno descritte meglio nel seguito.
E' compito del software controllare se tali file esistono e sono coerenti con le specifiche, se tali opzioni non sono coerenti con le specifiche verrà segnalato un opportuno errore e il software si chiuderà, non appena avviato il software crea sempre un file di log chiamato logfile, esso viene utilizzato per registrare le attività del software e registrare eventuali anomalie.

Dopodiché il software inizia con l'analisi dei file e inserisce i dati in un database di tipo RoundRobin, l'utilizzo di questo tipo di database è molto utile, visto che le dimensioni dello stesso una volta creato non crescono nel tempo e quindi il sistema è perfettamente mantenibile.
Dopo aver aggiornato il database, il software procede con la creazione del grafico, mostrando solamente i protocolli con utilizzo superiore al 3%.
Il software dopo l'analisi di ogni file, lo chiude sempre, per non incidere sull'uso della memoria.

2 Principali scelte del progetto

2.1 Classi utilizzate

Le classi create ad-hoc per il software sono due.
La prima:

```
public class ArgumentParser {

    private Hashtable<String, String> options = new Hashtable<String, String>();

    //Costruttore
    public ArgumentParser(String[] args);

    //Controlla se la stringa passata come parametro è nella lista di quelle
    consentite
    private boolean ControllaEsistenza(String subSequence);
    //Ritorna true se la stringa passata come parametro fa parte delle opzioni
    passate
```

```
    public boolean hasOption(String opt);
    //Ritorna il valore della stringa passata come parametro
    public String getOption(String opt);
```

Questa classe gestisce il parsing delle opzioni e controlla eventuali errori ed inesattezze nelle opzioni. Inoltre questa fa uso di una hash table per la memorizzazione dei parametri e delle relative opzioni.

Un'altra classe importante che ho utilizzato è la seguente:

```
public class ListaConcatenata {
    private Nodo start;
    private int size;
    private int occorrenzetotali;
    private int soglia = 0;
    private long data;
    private String filename ="";

    public ListaConcatenata();
    //Setta la data a cui si riferisce la lista
    public void setData(int year , int month , int date , int hourOfDay,
                        int minute);
    //Ritorna la data a cui si riferisce la lista
    public long getData();
    //Aggiunge i nomi dei protocolli
    public int addNameProtocol(String fileprotocol);
    //Aggiunge il protocollo
    public void add(int protocollo);
    //Aggiunge il protocollo
    public void add(String protocollo);
    //Rimuove il protocollo passato come parametro
    public void remove(int protocollo);
    //Ritorna true se la lista contiene almeno un elemento, false altrimenti
    public boolean isEmpty();
    //Ritorna la lunghezza della lista
    public int size();
    //Ritorna il numero di occorrenze totali
    public int occorrenzetotali();
    //Ritorna null se l'indice è fuori dalla lista altrimenti ritorna il nodo
    public Nodo getNode(int n);
    //Toglie dalla lista, i protocolli che hanno una percentuale d'uso minore ad
    una soglia stabilita
    public void calcolapercentuali();
    //Ritorna true se in nodo "n" è contenuto nella lista, false altrimenti
    public boolean contains(int n);
}
```

Dove `private Nodo` è definito come segue:

```
class Nodo {
    int prot;
    String name_protocol = null;
    int occ = 0;
    int percentuale = 3;
    Nodo next = null;
}
```

Essa è una lista concatenata ordinata che viene utilizzata per salvare le occorrenze di protocolli in ciascun file.

Descrizione classe "`ListaConcatenata`":

Il campo **"start"** indica qual è in nodo di testa, il campo **"size"** indica la dimensione della lista, il campo **"occorrenzetotali"** indica il numero di occorrenze totali di tutti i nodi (questa variabile è molto utile per il calcolo delle percentuali di utilizzo), il campo **"soglia"** indica il lowerbound di percentuale da considerare per i protocolli, il campo **"data"** si riferisce alla data in cui è stato creato il file che si sta considerando, mentre **"filename"** è il nome del file.

Descrizione classe "Nodo":
Il campo **"prot"** è l'intero che identifica il protocollo(ogni numero ha una corrispondenza biunivoca con il nome del protocollo), il campo **"name_protocol"** è la stringa che indica il nome del protocollo, il campo **"occ"** indica le occorrenze di quel protocollo, mentre **"next"** è il puntatore al nodo successivo.

3 Strutturazione del codice

3.1 Opzioni utilizzabili

- `[ --help | -help ]`
  Comando che visualizza la guida.
- `[ --db=<path database>]`
  L'opzione "db" ha una duplice funzione:
  se viene passato come path un database esistente, allora esso viene aperto e i file vengono aggiunti a tale database, altrimenti, viene creato un nuovo database con il nome scelto.
  Se invece come path viene passata una directory, il programma cerca un database con il nome "Database.rrd", se esiste, tale file viene aperto, altrimenti, il programma crea un database con il nome "Database.rrd" nella directory specificata.
  Valore di default: "/tmp/Database.rrd"
- `[ --log=<path log file>]`
  L'opzione "log" ha una duplice funzione:
  se viene passato come path un file esistente, allora esso viene aperto e il log del programma viene aggiunto a tale file, altrimenti, viene creato un nuovo file di log con il nome scelto.
  Se invece come path viene passata una directory, il programma cerca un file di log con il nome "logfile.log", se esiste, tale file viene aperto, altrimenti, il programma crea un file di log con il nome "logfile.log" nella directory specificata.
  Valore di default: "/tmp/logfile.log"
- `[ --file=<file da analizzare>]`
  L'opzione "file" riceve come input il path di una directory. Il programma cerca ricorsivamente all'interno delle directory i file con estensione "xx.flows", dove per xx si intende dei numeri compresi fra 0 e 59. L'applicazione effettua un'analisi dei file, riportando le inesattezze in caso di errori. Alla fine, realizza il grafico e l'applicazione termina.
  Valore di default: "/tmp/"
- `[ --protocols=<path file protocolli>]`
  L'opzione "protocols" serve per fare riconoscere al programma i nomi dei protocolli, tale opzione di default contiene tutti i protocolli, riconoscibili dal programma nprobe, se così non fosse si consiglia di scaricare gli aggiornamenti.
- `[ --html=<nome del file dell'html>]`
  L'opzione "html" serve per dare al programma il path di destinazione del file di output in formato html che fa visualizzare il grafico.
  Se viene passato il nome del file html, allora il file avrà quel nome; altrimenti se viene passata una directory, il sistema assegnerà automaticamente il nome: "VisualGraph.html".
  Valore di default: "/tmp/VisualGraph.html".
- `[ --start=<minuto/ora/giorno/mese/anno>]`
  Opzione che indica da quale data far partire il grafico
  esempio:10/20/02/6/2012

Valore di default: data di creazione database.

- `[ --end=<minuto/ora/giorno/mese/anno>]`
  Opzione che indica in quale data far terminare il grafico
  esempio:19/20/02/7/2012
  Valore di default: data ultimo aggiornamento database.