`SNMPTweet` - demone per la ricezione di trap e la notifica su Twitter

Nicola Corti - Michael Sanelli
Corso di Laurea in Informatica (L-31) - Università di pisa

12 Settembre 2011

Sommario

Questa relazione ha lo scopo di illustrare i dettagli implementativi, le scelte che sono state prese, e le metodologie di programmazione utilizzate nello sviluppo di `SNMPTweet`.

Indice

1 Introduzione

`SNMPTweet` è un demone che permette di ricevere le trap SNMP, di elaborarle e di pubblicarle sul social network Twitter (http://www.twitter.com). Le trap SNMP, secondo quanto descritto all'interno dell'RFC 1157, rappresentano messaggi asincroni normalmente utilizzati per indicare eventi o errori che si sono manifestati e a cui è necessario prestare attenzione.

L'utente puó configurare il proprio agent SNMP per inviare le trap al calcolatore dove è in esecuzione `SNMPTweet`, il quale elaborerà la coda delle trap ricevute e le invierà come Tweet su Twitter, secondo le modalitá decise dall'utente nel file di configurazione.

2 Dettagli implementativi e scelte prese

Segue una breve panoramica dell'architettura del sistema, le scelte che sono state prese e le motivazioni che ci hanno spinto a prenderle.

I dettagli descritti in questa sezione, servano d'ausilio ai programmatori che vogliano comprendere maggiormente la logica applicativa.

2.1 Architettura generale dell'applicativo

L'applicativo è stato sviluppato in Java, così da poter essere compatibile con tutte le piattaforme che supportino la JVM.

L'utilizzo di Java va però a discapito delle performance generali del software, che risultano inferiori rispetto ad una possibile realizzazione del software con un linguaggio più a basso livello (ad esempio C).

Abbiamo deciso di strutturare il nostro sistema in modo modulare, così da favorire una futura espandibilità dell'applicativo. L'utilizzo di un linguaggio di programmazione a oggetti ci ha aiutato in questo intento, in particolar modo ció é stato possibile grazie all'utilizzo dei `package` di Java.

I package che abbiamo realizzato sono:

- `twitter` che contiene le classi che si occupano dell'interfacciamento con Twitter;
- `snmptrap` che contiene le classi che si occupano di ricevere e leggere le trap SNMP;
- `intracomunication` che contiene le classi per gestire una o più code di trap.

Abbiamo utilizzato inoltre alcune librerie esterne:

Java SNMP [1] che implementa il protocollo SNMP;

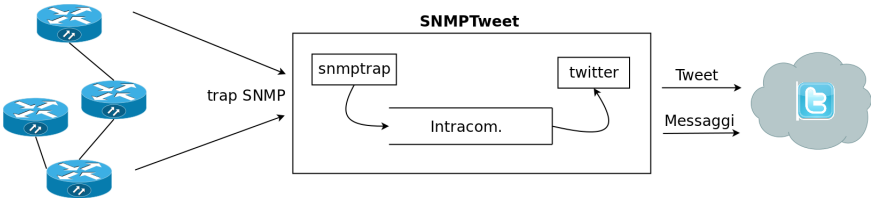
Twitter4J [2] che implementa le API di Twitter;

Mibble [3] che implementa un semplice parser per i MIB Files;

[1] http://gicl.cs.drexel.edu/people/sevy/snmp/
[2] http://twitter4j.org/
[3] http://mibble.org/

Log4j [4] che fornisce gli strumenti per gestire i file di log.

Segue una rappresentazione grafica dell'architettura del sistema:



2.2 Ricezione di trap SNMP

La ricezione delle trap è affidata alle classi contenute nel package `snmptrap`.

In particolare la classe `ListenTrapV1` implementa `SNMPv1TrapListener`, interfaccia definita nella libreria Java SNMP, che ci permette di definire un listener per le trap in entrata. All'interno di questa classe abbiamo definito il comportamento che il software deve avere all'atto della ricezione di una trap: ogni volta che viene ricevuta una trap, viene creato un nuovo oggetto all'interno del quale vengono inseriti i dati appena ricevuti e l'oggetto viene accodato per essere processato.

La classe `TrapReceiver` permette di inizializzare l'interfaccia di ascolto delle trap e di iniziarne la ricezione.

2.2.1 Ricerca di OID tra i MIB file

La classe `MibParser` implementa un piccolo parser per i file di Mib. Abbiamo deciso di realizzare il parser mediante un vettore di oggetti di tipo `Mib`, al quale si accede grazie ad un iteratore, e che permette di convertire un OID nella stringa della sua descrizione.

Grazie ai metodi contenuti nella classe è possibile:

- Caricare in memoria tutti i Mib da una directory settata fra le configurazioni.
- Convertire da un OID a Stringa, ricercando fra i Mib precedentemente caricati.

2.3 Interfacciamento con Twitter

La gestione dell'interfaccia con Twitter viene gestita dal package `twitter`.

La classe `MyTwitter` contiene tutti i metodi per potersi interfacciare in modo corretto con Twitter. Il metodo `HandShaker` consente di effettuare il login su twitter con l'account desiderato; nel caso in cui l'utente non abbia mai utilizzato in precedenza il software, verrà chiesto all'utente di visitare una pagina web per poter effettuare il login, una volta autenticato correttamente, i dati di accesso verranno memorizzati in modo da non dover più ripetere tale procedura.

Con il metodo `sendUpdate` è possibile inviare tweet o messaggi privati su Twitter in base alla configurazione. Il metodo si occupa anche di gestire tweet più lunghi di 140 caratteri.

[4] http://logging.apache.org/log4j/

Il metodo `ExceptionHandler` consente di gestire tutti gli eventuali problemi che potrebbero sorgere nell'autenticazione o nell'invio dei tweet (Twitter impone infatti varie limitazioni sul numero dei tweet che possono essere inviati), analizzando il codice della risposta http ricevuta da Twitter.

2.3.1 Gestione dell'autenticazione

Per il processo di autenticazione viene utilizzato il protocollo `OAuth`, che risulta il più sicuro fra i protocolli disponibili per l'autenticazione su twitter.

Per la gestione delle chiavi abbiamo utilizzato la classe `KeyContainer`, che funge da "contenitore" per le chiavi: sia per quelle legate al software, sia quelle legate all'utente.

Abbiamo utilizzato la serializzazione per rendere le chiavi persistenti, e per un minimo livello di sicurezza. Per rendere più sicura la gestione delle chiavi avremmo potuto cifrare il contenuto del file.

2.4 Logfile e Config file

Ogni operazione eseguita dal software viene registrata all'interno di un file di log, gestito grazie alla libreria `log4j`. In questo modo l'utente può capire meglio quali problemi sono avvenuti o quali tweet sono andati persi.

Abbiamo inoltre deciso di fornire all'utente un file di configurazione (il file "./snmptweet.conf") in modo da poter configurare a piacimento le seguenti funzionalità:

LOG_FILE il nome del file di log,

MIB_DIR la directory dei Mib,

PORT la porta di ascolto delle trap SNMP.

Inoltre settendo il valore del parametro `PRIVATE` a true (di default il valore è false), verranno inibiti tutti gli invii di tweet pubblici, everranno inviati soltanto messaggi privati. Se si imposta `PRIVATE` a true, è necessario configurare inoltre le associazioni fra utenti di Twitter e indirizzi IP dell'agent che ha inviato la trap.

Tali associazioni si possono impostare mediante l'inserimento di coppie del tipo $(USER_i, IP_i)$.

Nel caso in cui non sia settata la modalità `PRIVATE`, ma siano state settate delle associazioni, verrà inviato un tweet pubblico e verrà inviato inoltre un messaggio privato all'utente relativo.

3 Possibili sviluppi futuri

Il software è stato sviluppato in un'ottica da poter permettere eventuali miglioramenti futuri, grazie alla sua architettura modulare, e alla sua suddivisione in pacchetti.

Presentiamo alcuni dei possibili miglioramenti che potrebbero essere implementati nelle versioni successive:

- Possibilità di interrogare i dispositivi, o di modificare le loro configurazione con delle `GET/SET SNMP` mediante dei messaggi privati inviati all'utente collegato a snmp-tweet.
- Possibilità di impostare varie tipologie di filtri per limitare la ricezione delle `TRAP`.
- Possibilità di impostare la modalità privata o pubblica in modo separato per ogni dispositivo.
- Possibilità di impostare la geo-localizzazione dei dispositivi che emettono trap e di corredare i tweet delle informazioni geografiche.
- Possibilità di gestire più di un account twitter contemporaneamente.
- Possibilità di impostare più porte sul quale ricevere le `TRAP`.
- Possibilità di ricevere `TRAP` versione 2 e 3 e le `INFORM`.

4 Installazione del software

Il software è corredato di un file `ant` per l'automazione del processo di installazione.

Per creare il `jar` eseguibile è sufficiente invocare il comando

```
ant makejar
```

nella directory root del progetto. Così facendo, il file `jar` si troverá all'interno della directory `dist`, e da lì potrà essere eseguito mediante il comando:

```
java -jar SNMPTwitter.jar
```

Se invece si è interessati a compilare i file binari, è sufficiente invocare il comando `ant`. I file binari si troveranno dentro la directory `bin`.

5 Documentazione allegata

Tutto il codice sorgente scritto è corredato di commenti che descrivono i dettagli implementativi del software.

Inoltre è stato scritto il javadoc dell'intero progetto in modo da poter aiutare i programmatori nella comprensione dell'organizzazione logica delle classi.

Per ricreare il javadoc è sufficiente utilizzare il target `javadoc` del file ant.

6 Licenza d'uso

SNMPTweet è rilasciato sotto licenza libera, in modo che ogni utente sia libero di poterlo modificare e ridistribuirlo liberamente.

In particolare è stato rilasciato sotto licenza GNU GPL (General Public Licence) v.3:

SNMPTweet is free software: you can redistribute it
and/or modify it under the terms of the GNU General
Public License as published by the Free Software Foundation,
either version 3 of the License,
or (at your option) any later version.

SNMPTweet is distributed in the hope that
it will be useful, but WITHOUT ANY WARRANTY;
without even the implied warranty of MERCHANTABILITY
or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General
Public License along with Foobar.
If not, see <http://www.gnu.org/licenses/>.