

PKTFLOW versione 1.2.0

Questo documento descrive gli aggiornamenti apportati all'applicazione Packet Flow v1.2.0 rispetto la versione 1.0.

Le funzionalita' aggiuntive messe a disposizione in questa versione sono due:

1. Il thread Collector puo' decidere se salvare i flussi in un file flows.xml o stamparli a video come la versione precedente.
2. Viene introdotta la possibilita' di gestire un RRD attraverso gli RRD tools.

In fase di start-up si puo' decidere se caricare un profilo di "default" o personalizzato dell'applicazione.

Gestione di RRD.

Questa versione permette la gestione di alcuni dati riguardanti i flussi attraverso un RRD (`./flows.rrd`).

Nel database viene salvato periodicamente un contatore dei flussi "spirati".

Il comando utilizzato per la creazione dell'RRD e':

```
rrdtool create flows.rrd --start now --step 60  
DS:flows:COUNTER:119:0:U RRA:LAST:0:1:60 RRA:AVERAGE:0.5:5:288
```

Il tipo di dato utilizzato e' COUNTER. Il database si aspetta di acquisire il contatore dei flussi allo scadere di ogni minuto, dopodiche' effettua operazioni matematiche in maniera tale che i dati vengano salvati nel formato *flows/second*.

L'heartbeat e' di 1 minuto -1secondo affinche' non vi siano problemi di sincronizzazione tra database e applicazione se questa inizia la sua esecuzione in momento qualsiasi dopo che il RRD e' stato creato.

I dati acquisiti vengono salvati in due RRA..

Il primo RRA consolida subito i dati ricevuti ed e' formato da un totale di 60 rows.

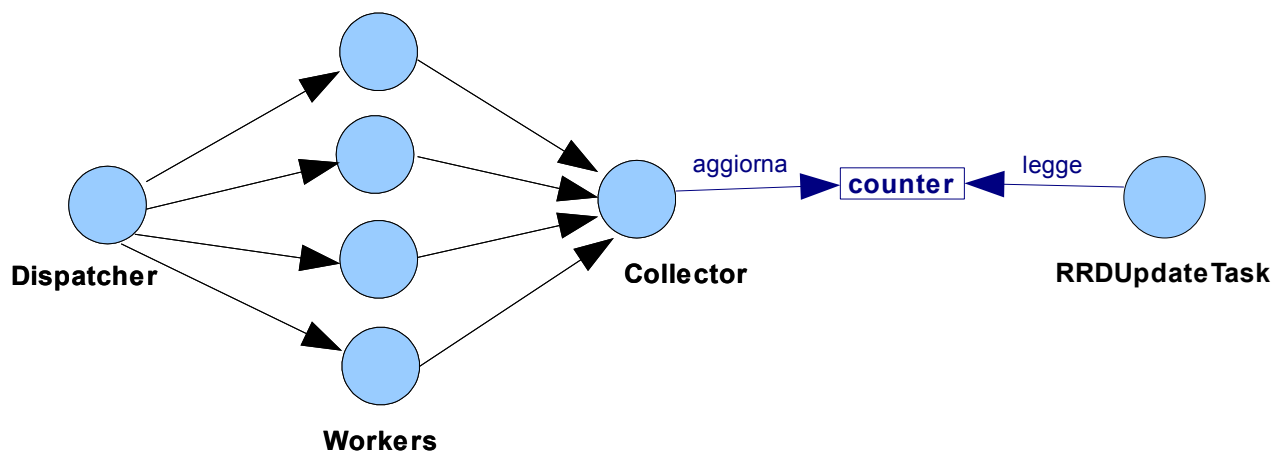
In altre parole salva la media su 1 minuto della velocita' dei flussi e li tiene in archivio per un ora.

Il secondo RRA invece consolida la media su 5 minuti della velocita' dei flussi per un totale di 288 rows.

Ossia fa la media di 5 medie su un minuto della velocita' dei flussi consolidando il dato e tenendolo in archivio per un giorno.

Per velocita' si intende la quantita' di flussi ricevuti al secondo (*flows/second*) di cui pero' si fa una media al minuto o ai 5 minuti.

Per permettere quindi a `flows.rrd` di acquisire i dati riguardanti i flussi "spirati" l'applicazione e' stata aggiornata in questa maniera:



Il thread Collector aggiorna un contatore dei flussi ogni volta che riceve un flusso “expired” e prosegue nella sua normale esecuzione.

Il thread RRDUpdateTask e' associato ad un timer. Allo scadere di ogni minuto legge il contatore dei flussi ed esegue:

```
rrdtool update flows.rrd --template=flows N:<valore counter>
```

E' chiaro che questa soluzione non comporta particolari problemi computazionali o di sincronizzazione.

Packet Flow presenta un piccolo “issue”.

Il programma e' in grado di scegliere l'interfaccia di rete da monitorare. Tuttavia i dati riguardanti i flussi sono salvati nel file `flows.rrd` indistintamente da quale fonte provengano.

Ossia `flows.rrd` raccoglie le statistiche riguardanti i flussi senza avere idea da quale interfaccia di rete provengano.

Il thread Collector e il pacchetto “pktflowgui”.

Per la visualizzazione dei dati salvati nel RRD sono stati utilizzati gli strumenti messi a disposizione da **RRDtool** per la realizzazione di grafici, salvati nel formato png.

In particolare:

- `./pktflowgui/graph/graph_hour.png` visualizza l'andamento della velocita' dei flussi nell'ultima ora (con granularita' del dato al minuto)
- `./pktflowgui/graph/graph_day.png` visualizza l'andamento della velocita' dei flussi nell'ultimo giorno (con granularita' del dato ai 5 minuti)

E' chiaro che a questo punto i dati riguardanti i flussi e i grafici potrebbero essere visualizzati in un unica pagina web.

Da qui nasce “*pktflowgui*”.

La directory `./pktflowgui/` contiene una pagina html (`./pktflowgui/pktflow.html`), una directory in cui sono salvati i flussi (`./pktflowgui/flows/`) e una in cui sono salvati i grafici (`./pktflowgui/graphs/`).

`pktflows.html` visualizza i grafici e stampa a video i flussi salvati in

`./pktflowgui/flows/flows.xml`.

La pagina dispone anche di un piccolo script per il refresh automatico in maniera tale da visualizzare eventuali nuovi dati a “real time”.

Alla pagina `flows.xml` e' stato associato un foglio di stile xslt

(`./pktflowgui/flows/tabletemplate.xsl`) in maniera tale da dividere nettamente il livello dei dati dal livello della presentazione.

Questo mi ha aiutato non poco in fase di implementazione, permettendomi di arrivare ad una soluzione dall'impatto computazionale minimo. Infatti il thread Collector in questo caso invece di stampare i dati a video, stampa i dati su `./pktflowgui/flows/flows.xml`, utilizzando un formato xml molto scarno in modo da non sprecare spazio ed eseguendo l'operazione a tempo costante.

Sara' il browser a stampare a video i dati contenuti nel file xml, processandoli secondo il foglio di stile xslt.

Pktflowgui e' realizzato in maniera tale che la visualizzazione della pagina `pktflow.html` sia completamente off-line in maniera da non generare eventuale traffico “spurio” sull'interfaccia di rete monitorata.

