Relazione Progetto Gestione di Reti

Davide Rucci

10 luglio 2017

1 Introduzione e Struttura dei Files

Lo scopo di questo progetto è di raccogliere metriche riguardanti l'utilizzo della rete e lo stato della macchina rispettivamente tramite `ntopng` e `snap-telemetry` e di mostrarle con `Grafana`. Questa relazione tratta gli aspetti generali del progetto, per una guida dettagliata all'installazione e alla configurazione di tutti i tools si consulti il file README contenuto nella stessa cartella di questa relazione. La stessa cartella contiene inoltre i seguenti files:

- README
- Makefile
- dashboards/
  - machine_dash.json — esportazione della dashboard di Grafana con le metriche riguardanti stato della macchina
  - net_dash.json — esportazione della dashboard di Grafana con le metriche riguardanti la rete
- task.json — file manifest del task da avviare con Snap, contiene l'elenco delle metriche da raccogliere.
- plugins/∗ — eseguibili dei vari plugin usati da Snap per raccogliere e salvare le metriche.

2 Snaptel: plugin, metriche e task

Il framework Snap-Telemetry consente di raccogliere, elaborare e memorizzare diversi tipi di dati grazie ad un sistema di plugin suddivisi in tre categorie: *collector*, *processors* e *publishers*. I primi si occupano di collezionare effettivamente i dati, i secondi li elaborano in vari modi, se necessario, e gli ultimi li pubblicano (cioè salvano) in memorie che possono essere di diverso tipo, dagli RRD a dei semplici file JSON. Il *publisher* che più si è adattato alle nostre esigenze è `KairosDB`, un RRD open source scritto in Java che fornisce, tra le altre cose, sia un'interfaccia web che un plugin di integrazione con Snap rendendo quindi immediata la sua configurazione. Una volta installati i plugin che si vuole usare è necessario creare un *task*, ovvero un "lavoro" da far fare a Snap secondo dei parametri prestabiliti. Il file di task, nello specifico il *manifest* del task, può essere scritto in YAML o in JSON (nel nostro caso si è preferito quest'ultimo linguaggio) e si compone principalmente di tre parti: la specifica di scheduling, le metriche da raccogliere e dove salvarle. In questo caso il tipo di scheduling è uno scheduling semplice, che raccoglie tutte le metriche elencate nel seguito una volta ogni secondo e quindi le invia al database Kairos, configurato per essere accessibile sulla porta 8080 della stessa macchina.
Le metriche riguardanti lo stato della macchina vengono dunque raccolte tramite specifici plugin di Snap, mentre quelle relative alla rete vengono raccolte sia da un plugin di Snap che dalle API messe a disposizione da ntopng. Ecco quindi l'elenco dei plugins di tipo *collector* utilizzati e quali metriche vengono raccolte da essi:

- PSUtil
  - load/load1: il carico medio della CPU nell'ultimo minuto, un numero decimale compreso tra 0 e il numero di core a disposizione sulla macchina
  - load/load15: come sopra ma riferito agli ultimi 15 minuti
  - net/all/packets_recv: il numero totale di pacchetti ricevuti dalla rete (su tutte le interfacce)
  - net/all/packets_sent: il numero totale di pacchetti inviati sulla rete (da tutte le interfacce)
  - net/all/dropin: Il numero di pacchetti di rete scartati in ricezione (su tutte le interfacce)
  - net/all/dropout: il numero di pacchetti di rete scartati in invio (su tutte le interfacce)
- Netstat
  - tcp_syn_recv: pacchetti TCP ricevuti col flag SYN settato, utile per accorgersi di possibili attacchi DOS di tipo SYN Flood
  - tcp_established: il numero di connessioni TCP attualmente attive e nello stato *established*
- Disk
  - sda/ops_read: il numero di operazioni del disco `sda` al secondo (in lettura)
  - sda/ops_write: come sopra, ma in scrittura
- Meminfo

- mem_total: la quantità di memoria totale del sistema, in bytes
- mem_used: la quantità di memoria attualmente in uso sul sistema, in bytes, ottenuta come $mem\_total - mem\_free - buffers - cache - slab$ dove *slab* indica la memoria riservata al kernel per mantenervi le sue strutture dati
- mem_available: una stima della memoria disponibile per avviare un nuovo processo senza ricorrere allo *swap*

La maggior parte di questi dati viene raccolta dal `procfs`, lo pseudo file system di Linux che mette a disposizione varie informazioni direttamente dal kernel.

3 Grafana e ntopng

Grafana è una piattaforma per la visualizzazione, l'analisi e il controllo di una vasta gamma di metriche in tempo reale che acquisisce dati da una o più *data sources* e li visualizza a schermo in varie forme, come grafici, tabelle, o un unico valore ed è molto potente e flessibile grazie al suo sistema di plugin che consente di espandere le sue funzionalità di base. Grazie al plugin `Kairosdb-data-source` riesce a collegarsi all'interfaccia REST di un'istanza di Kairosdb per ottenere i dati da aggiungere alle varie visualizzazioni. In un modo simile, utilizzando un plugin che abilita le JSON *data sources* è possibile collegarsi ad un'istanza di ntopng per ottenere metriche anche da questo software di monitoraggio di rete. Nello specifico, da ntopng si ottiene una misurazione del traffico in bit/sec a livello di protocolli applicativi resa possibile dall'integrazione di nDPI, un software aggiuntivo in grado di distinguere tra più di 100 protocolli ricorrendo a tecniche di deep packet inspection. È possibile creare anche più grafici di questo tipo, uno per ogni interfaccia di rete utilizzata, ma in questo caso se ne è creato solo uno che acquisisce dati da una sola interfaccia, la più utilizzata. Ancora tramite ntopng è stato aggiunto un grafico che permette di monitorare la quantità di bit al secondo passanti su una certa interfaccia, anche in questo caso la più utilizzata, consentendo così di evidenziare dei picchi di traffico e, se necessario, agire di conseguenza.
Grafana offre la possibiltà di personalizzare gli intervalli di visualizzazione delle varie dashboard, così come l'intervallo di auto-refresh delle stesse; nelle dashboard fornite questi parametri sono impostati su "ultime 12 ore" e "5 secondi" rispettivamente, ma con l'uso degli appositi controlli in alto a destra nella pagina della dashboard è possibile modificarli a proprio piacimento, ottenendo ad esempio dei grafici come quelli in figura.

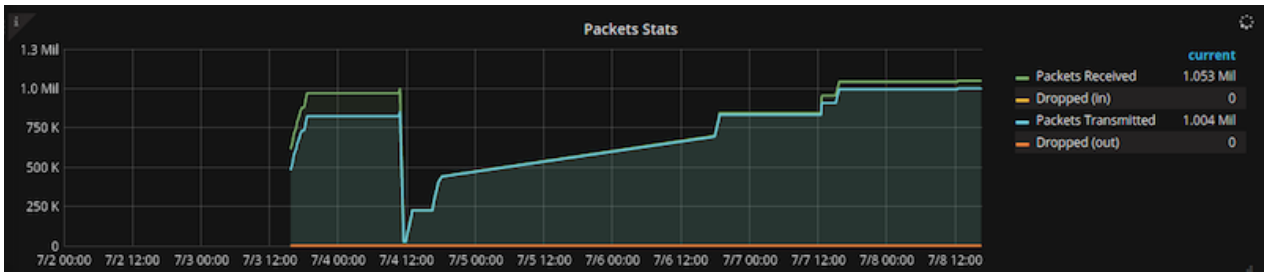

Figura 1: Esempio di grafico dei pacchetti inviati e ricevuti sull'interfaccia principale nell'ultima settimana

Il secondo tipo di pannello più usato è stato quello "single stat" che consente di visualizzare l'andamento numerico di una singola metrica eventualmente coadiuvato da un piccolo grafico nella parte inferiore di esso e da un indicatore di gauge che consente di stabilire delle soglie e di colorare la metrica con tre tonalità di verse in base a due soglie di attenzione. Nella figura seguente, si può notare come il valore sia colorato di verde in quanto sotto alla soglia di attenzione di valore 1, oltre all'andamento della metrica grazie al semplice grafico posto in basso.

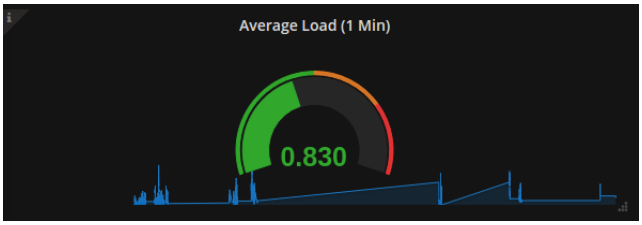

Figura 2: Esempio di pannello "single stat" con soglie di attenzione 1 e 1.5 e grafico nella parte inferiore