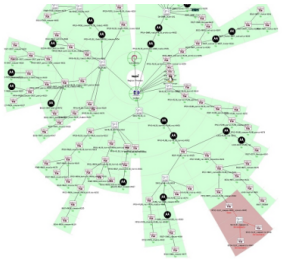
Networking

Analisi e Implementazione di Sistemi per il Monitoraggio della Rete Wireless Relativa al Progetto ADD (Anti Digital Divide) e delle Infrastrutture di Campus AdR RM1.†

Augusto Pifferi,[a] Giuseppe Nantista,[a] Luca Ianniello,[a] Andrea Lora,[a] and Marco Simonetti.[a]



In questo documento verranno valutati 3 sistemi di monitoring Open-Source (Nagios, Cacti e Zabbix) e si mostreranno sia le peculiarità che i differenti pro e contro. Verrà esposto il funzionamento di ogni singolo sistema con le relative operazioni di installazione e configurazione, al fine di fornire un quadro completo delle potenzialità dei mezzi. Infine verrà illustrato l'utilizzo dei tre sistemi di monitoraggio nel caso reale della rete di campus dell'Area della Ricerca Roma 1 e della rete wireless realizzata nel territorio della sabina romana e reatina.

Keywords: Monitoring System, Zabbix, Cacti, Nagios.

1 Introduzione

Gestire un'architettura ICT complessa è un processo che si articola in più fasi, tutte indispensabili al mantenimento in linea dei servizi offerti, dalla progettazione, configurazione e allestimento della rete, atta a offrire il servizio agli utilizzatori, fino alla gestione dei server che ospitano il servizio stesso.

Successivamente alla fase di deployment della struttura e del servizio, inizia una attività fondamentale per mantenere il servizio correttamente funzionante: il monitoraggio dei sistemi, che avrà durata pari a quella del servizio stesso.

Un approccio non automatizzato al monitoring non è pensabile per tre ragioni fondamentali:

- Il monitoring è un'operazione time-consuming;
- Al fine di minimizzare i tempi di ripristino è necessario intervenire tempestivamente;
- In determinati casi il problema è preceduto da una situazione di instabilità di un sistema, non sempre tangibile da un osservatore umano.

La risposta informatica all'esigenza di sollevare l'uomo dal monitoring è stata la creazione di appositi software che permettano di configurare una serie di controlli finalizzati alla raccolta di dati e facciano partire specifiche azioni a seguito del manifestarsi di condizioni prestabilite.

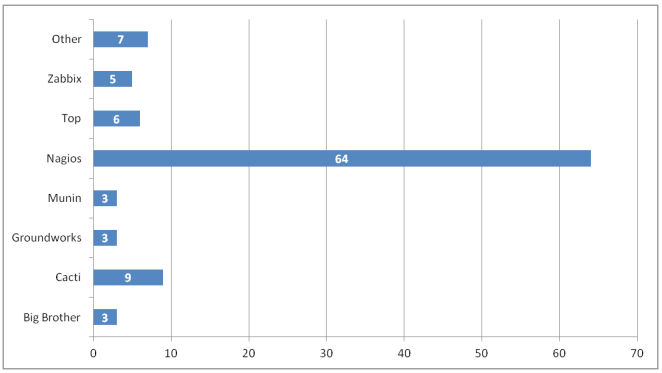


Fig. 1 Diffusioni prodotti OpenSource per il monitoraggio di reti e sistemi ICT.

Nell'ottica quindi di un controllo costante delle infrastrutture ICT all'interno e all'esterno dell'Area della Ricerca Roma 1 del CNR di Montelibretti si è effettuato uno studio dei prodotti Open Source esistenti in rete, quindi sono state realizzate e configurate diverse piattaforme.

2 NAGIOS

2.1 Introduzione

Poiché risultava essere il più diffuso e il più documentato, la prima scelta è ricaduta su Nagios.

Il software è distribuito gratuitamente con licenza GNU – GPL; maggiori informazioni sono disponibili sul sito http://www.nagios.org/

Nel caso specifico il software è stato installato su distribuzione Linux Debian 6.0.3 squeeze su hardware virtuale

[a] *CNR - Istituto di Cristallografia, Strada Provinciale 35/d, Montelibretti, Italia*

† Rapporto tecnico IC-RM/1058 protocollato in data 03/07/2012

VMWare ESXi 5.0

- 1 virtual cpu Xeon®2.0 ghz
- 1 GB ram
- 16 GB hard disk
- 1 GBit ethernet LAN

Nagios possiede un'interfaccia web attraverso la quale è possibile monitorare dispositivi, utilizzando un diverso livello di dettaglio; inoltre è possibile configurare un sistema di invio notifiche riguardante i malfunzionamenti, basato sull'invio di e-mail.

Nagios è in grado di monitorare anche host e servizi, inoltre può essere completato attraverso l'applicazione di scripts e plug-in.

Tale operazione di completamento consente un'ottima flessibilità, tuttavia richiede eccessivo dispendio di tempo perché l'unico modo di configurare il software è da CLI (command line interface).

Una caratteristica interessante è la possibilità di definire, anche graficamente, dipendenze tra apparati monitorati, tramite la costruzione di grafi ad albero, che permettono di evidenziare immediatamente le dipendenza padre-figlio, così da individuare il nodo dell'albero sul quale si è verificato un problema.

2.2 Configurazione

Avendo usato come distribuzione Debian, l'installazione è stata immediata, il gestore di pacchetti ha risolto automaticamente tutte le dipendenze.

```
root@nagios:/# apt-get install nagios3
```

A questo punto la configurazione vera e propria ha riguardato i file relativi agli host da monitorare. Per semplicità ci siamo basati sul template "generic-host"

```
root@nagios:/# more /etc/nagios3/conf.d/generic-host_nagios2.cfg

8< ----------------------

define host{
name            generic-host    ; The name of this host template
notifications_enabled       1 ; Host notifications are enabled
event_handler_enabled       1 ; Host event handler is enabled
flap_detection_enabled      1 ; Flap detection is enabled
failure_prediction_enabled  1 ; Failure prediction is enabled
process_perf_data           1 ; Process performance data
retain_status_information   1 ; Retain status information across
     program restarts
retain_nonstatus_information 1 ; Retain non-status information
     across program
check_command                   check-host-alive
max_check_attempts              10
notification_interval           0
notification_period             24x7
notification_options            d,u,r
contact_groups                  admins
register                        0
}

---------------------- >8
```

Nella configurazione dei singoli host sono state aggiunte solo le informazioni specifiche, come l'indirizzo IP, il nome di sistema e il parent, ossia l'apparato che, in una struttura ad albero, risulta essere il padre dell'host in questione. La totalità delle dipendenze padre-figlio vengono automaticamente gestite da Nagios, che costruisce la site-map (Figura 2).

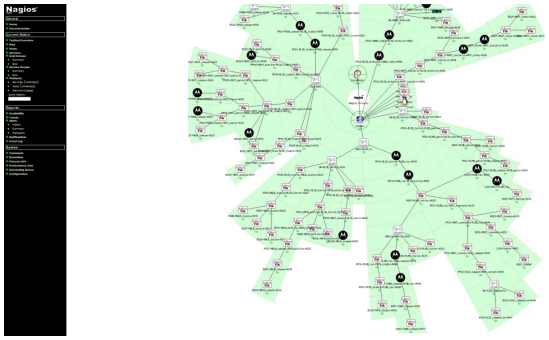


Fig. 2 Mappa di Nagios.

```
root@nagios:/# more /etc/nagios3/objects/host_totali.cfg

8< ----------------------

define host {

use                     generic-host
host_name               BS01-MRCN_bottin-H173
alias                   alias
address                 10.10.235.173
parents                 PP21-MRCN_bottino-MLIB_torre1-H172
contact_groups          admins

}

---------------------- >8
```

Anche gli host sono stati separati in categorie nell'ottica di raggrupparli per tipologie di controlli differenti.

Infine, per distinguere a colpo d'occhio il manufacturer dell'apparato, è stata associata un'icona ed esplicitato il tutto nel file hostextinfo che estende le informazioni appartenenti a un determinato gruppo.

```
root@nagios:/# more /etc/nagios3/objects/hostgroup_totali.cfg

8< ----------------------

define hostgroup {
hostgroup_name   Mikrotik
alias            alias
members         BS01-MRCN_bottin-H173, BS02-SRST_parco-
                                              H014, BS03-PLMB_
}

---------------------- >8
```

Tali configurazioni sono state effettuate seguendo scelte personali e non sono un dictat di Nagios stesso, che invece permette diverse varianti di configurazione a seconda dei gusti o delle specifiche finalità dell'amministratore.

2.3 Visualizzazione

Nelle successive due immagini è visualizzata la classica MAP di Nagios, costruita automaticamente dall'engine prendendo spunto esclusivamente dalle relazioni di parentela specificate in fase di definizione degli host. Si noti come, in caso di down di un ramo (Figura 3) l'attenzione va subito al nodo in cui si è interrotta la comunicazione, facilitando la comprensione del guasto e mettendo in moto rapidamente le procedure di ripristino.

2.4 Valutazioni

L'utilizzo di questa piattaforma ha evidenziato, almeno dal nostro punto di vista, alcuni limiti:

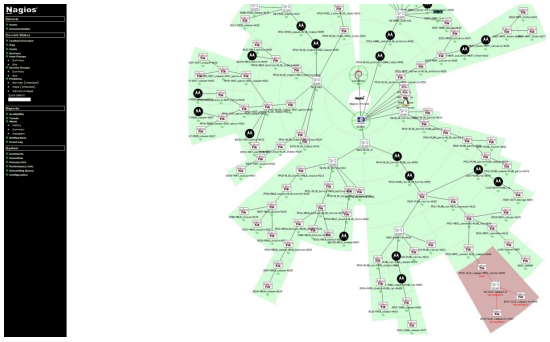


Fig. 3 Mappa di Nagios con evidenziato malfunzionamento host.

- L'installazione, nella sua versione di base, non include uno strumento per la raccolta, memorizzazione e visualizzazione grafica di semplici dati, quali l'utilizzo di CPU, il traffico sulle interfacce di rete, ecc.
- Come già detto, tutta la configurazione, compresa la definizione di allarmi, avvisi e relative soglie di attivazione, si effettua esclusivamente via CLI, rendendo l'operazione poco intuitiva.

3 CACTI

3.1 Introduzione

Il secondo software valutato è stato Cacti, anch'esso distribuito con licenza gratuita GNU – GPL e scaricabile all'indirizzo http://www.cacti.net/

Data la quantità ridotta di risorse richieste e l'assenza di conflitti, è stato installato sulla stessa macchina virtuale utilizzata per Nagios.

Immediatamente si evidenziano due vantaggi:

- La semplice e rapida gestione degli host e dei servizi da monitorare, completamente configurabili via web interface
- La presenza, già nella configurazione di base, dei tool relativi a raccolta, immagazzinamento e visualizzazione grafica dei dati

Merita una nota il sistema utilizzato da Cacti per la memorizzazione dei dati raccolti, il database RRD (Round Robin Database). Questo tiene sotto controllo l'espansione del database, infatti i dati vengono immagazzinati con una densità variabile nel tempo. I dati sono raccolti con una cadenza frequente, ma vengono compressi, tramite media matematica, man mano che diventano più "vecchi", fino ad essere completamente sovrascritti una volta superato il tempo massimo di memorizzazione, che di default è fissato a un anno.

Risulta evidente quindi che, per ogni check effettuato dal software, è fissa la dimensione massima del database su cui i dati sono memorizzati. Questo sgrava l'amministratore di sistema da tutti i problemi relativi alla crescita incontrollata del database.

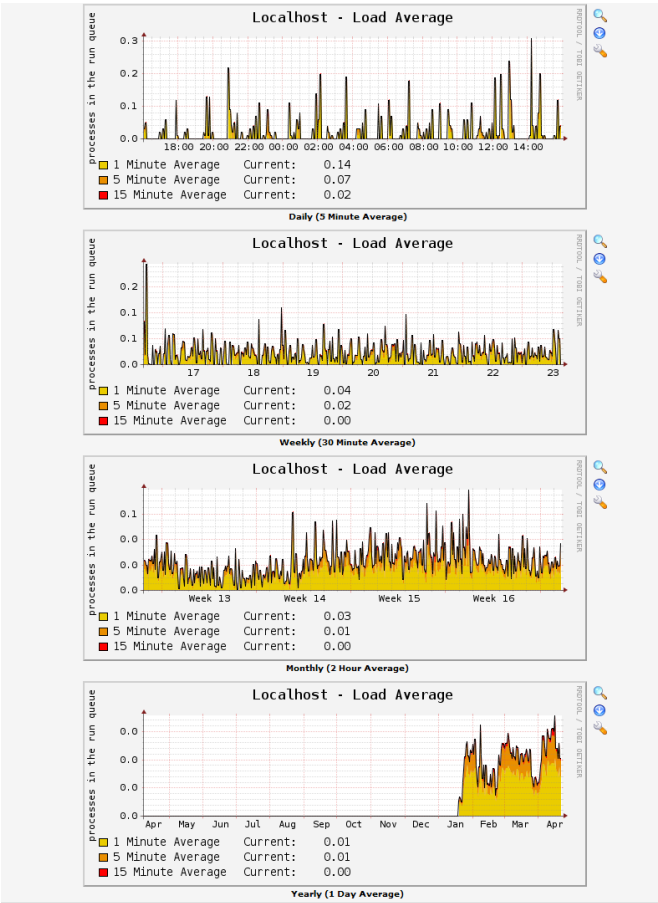


Fig. 4 Grafici e gestione database

3.2 Configurazione

Anche in questo caso la distribuzione Debian ha semplificato enormemente la fase di installazione, installando i componenti aggiuntivi in maniera automatica.

```
root@cacti:/# apt-get install cacti
```

Un aspetto che facilita molto l'inserimento nel database di un gran numero di host è la possibilità di definire dei template, ossia un elenco di controlli da associare a un gruppo di apparati. In questa maniera non appena un host viene creato e associato a un template, immediatamente sono disponibili tutti i relativi check. I template predefiniti sono relativamente pochi, ma è disponibile su internet una grande varietà di personalizzazioni. Nel nostro caso avendo nella rete alcuni host Mikrotik abbiamo aggiunto un template per questi apparati. Tutti i controlli effettuati da Cacti sono query SNMP, pertanto gli apparati devono essere compliant almeno con la versione v1 di tale protocollo.

Creare e visualizzare un grafico è semplice e immediato, questo è il motivo per cui questo strumento è stato preferito ad altri per avere una visione globale di quello che accade nella rete.

Nonostante non abbia, nel pacchetto base, strumenti per la definizione di una soglia di malfunzionamento, lo strumento dà la possibilità, previa osservazione diretta da parte di un osservatore umano, di accorgersi se un

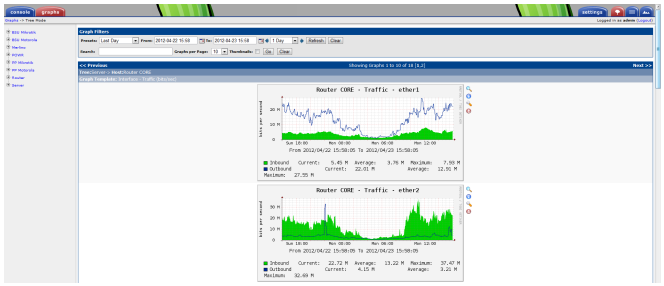


Fig. 5 Traffico in/out sulle interfacce di un router.

apparato funziona male, ad esempio se l'utilizzo medio della CPU è elevato, o se il traffico sulle interfacce arriva a saturazione, oppure se la latenza (icmp ping) sale eccessivamente.

3.3 Valutazioni

Cacti fornisce in definitiva uno strumento facilmente configurabile e particolarmente indicato per il monitoraggio di apparati di rete, non può prescindere dalla presenza di un operatore, ma grazie al database RRD può immagazzinare grandi moli di dati relativi alle osservazioni compiute e fornire un ottimo strumento per verificare le variazioni temporali nell'utilizzo di apparati di rete, prevenendo situazioni di collasso di apparati, ad esempio, per eccessivo utilizzo.

4 ZABBIX

4.1 Introduzione

Il terzo software valutato è stato Zabbix (http://www.zabbix.com). Rilasciato con licenza GNU GPL 2, Zabbix si propone come una soluzione integrata di Nagios e Cacti. Vengono infatti offerte all'interno dello stesso software sia la gestione degli allarmi che la possibilità di visualizzare gli storici dei dati raccolti dal sistema.

4.2 Configurazione Server

Dovendo decidere come approntare il server Zabbix abbiamo constatato che gli sviluppatori offrono una virtual appliance pronta da utilizzare, basata su openSUSE. Abbiamo optato per questa possibilità.

Le risorse assegnate tramite VMWare sono state di 1GB di memoria RAM fisica e 1 Virtual CPU del server VMWare, che nel nostro caso monta degli Intel(R) Xeon(R) CPU E5504@2.00GHz. Il link di rete è ad 1GBps. Per quanto concerne la scelta del sistema disco, essa dipenderà dalla localizzazione del database MySQL.

Nel caso di storage su server esterno, il traffico I/O di Zabbix su disco si limita a 3/5 kbit al secondo durante le fasi di monitoring, e lo spazio necessario è quello dell'installazione della distribuzione. Nel caso invece di server MySQL sulla stessa macchina Zabbix, lo storage locale deve essere aumentato, e le prestazioni del sistema dovranno essere monitorate per evitare che il traffico I/O disco saturi la macchina (I/O waiting).

Poiché le operazioni di I/O disco di Zabbix si limitano alla lettura/scrittura di dati sul server MySQL, a seconda del numero di host e del numero e della frequenza dei check, le query verso il database possono diventare numerose a tal punto da degradare le performances del sistema. Può dunque diventare conveniente delocalizzare il database MySQL fuori dalla macchina Zabbix. Questo è proprio il nostro caso, infatti Zabbix monitora 160 apparati, eseguendo circa 10 scritture su DB al secondo, per un totale di circa 650 check totali.

Una volta installato tutta la configurazione e consultazione avviene tramite interfaccia web, scritta in php, che si appoggia anch'essa su database MySQL. Rimandiamo alla pagina web di Zabbix http://www.zabbix.com/ per una lista esaustiva delle sue capacità, qui ci limiteremo ad elencarne le principali, comunque sufficienti ai nostri scopi.

4.3 Schema di funzionamento

Le modalità con cui funziona Zabbix sono mostrate nel seguente grafico. (fig. 6)

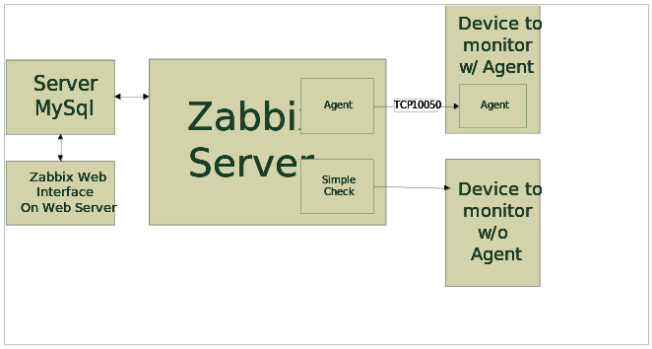


Fig. 6 Schema di funzionamento di Zabbix.

Il processo Zabbix server è il cuore del sistema, si occupa di eseguire tutti check e lanciare le eventuali azioni condizionali. Tutti i dati che il server raccoglie durante il suo funzionamento sono memorizzati su un database SQL. MySQL come già ribadito è il DBMS in questo momento supportato da Zabbix.

La Zabbix web interface è lo strumento con cui gli utenti si interfacciano al sistema. Essa non è legata al processo Zabbix server, e può essere ospitata fuori dalla macchina stessa. L'interfaccia infatti instaura una connessione con il server SQL direttamente, e non opera con il processo server se non tramite questo scambio dati.

L'interfaccia web permette sia la consultazione dei dati, sia l'inserimento/modifica/cancellazione di host e azioni di monitoraggio. Offre anche la possibilità di cambiare i meccanismi di controllo, e permette infine di inserire i classici Acknowledge sugli alert o lo scheduling di manutenzioni. Sempre dall'interfaccia andremo anche a

configurare le eventuali azioni che Zabbix dovrà eseguire in caso di una condizione determinata.

4.4 Zabbix Check

I controlli possono essere di diversi tipi, il sistema è stato infatti concepito per essere il più flessibile possibile. Se quindi vengono offerti di default controlli come quello della risposta al ping, della raggiungibilità di una ben precisa porta TCP o altri controlli basati su SNMP, nulla impedisce di costruirsi check propri, concepiti magari tramite un comando esterno.

I check esterni, chiamati in Zabbix simple checks, non riguardano una singola macchina, ma esprimono piuttosto un'azione che lo Zabbix server esegue. Tali check vengono eseguiti direttamente dalla macchina che ospita il processo Zabbix server, tramite i noti protocolli UDP/TCP. Le macchine da controllare dovranno solo rispondere alle richieste standard. Questo tipo di controlli non prevede particolari configurazioni sulle macchine da monitorare.

4.5 Zabbix Agent

Lo Zabbix agent, di contro, è un'ulteriore opzione che viene data per monitorare i sistemi. Si installa come servizio sulla macchina da monitorare e ha dunque accesso a tutte le informazioni di sistema della macchina stessa. Esso è disponibile per la maggior parte dei sistemi operativi correntemente in uso. L'agent mette a disposizione due tipi di check: attivi e passivi. Per check passivo viene inteso il meccanismo con cui Zabbix Server interroga lo Zabbix Agent (instaurando una connessione TCP tra i due). I check attivi invece vengono configurati direttamente nell'Agent, che si occupa di instaurare una connessione verso il server.

L'agent di Zabbix viene rilasciato come sorgente e come pacchetto precompilato. Nel caso di sistemi Linux esso è disponibili nei repository delle varie distribuzioni. L'installazione su Debian si limita a

```
root@debian:/# apt-get install zabbix-agent
```

Il file di configurazione /etc/zabbix/zabbix_agentd.conf andrà modificato per inserire a quale server Zabbix rispondere. Di base la riga da modificare è solo quella che riguarda l'hostname

```
Server=my.zabbix.server.fqdn
```

Eventualmente è possibile attivare l'esecuzione di comandi remoti aggiungendo questa riga al file

```
EnableRemoteCommands=1
```

Nel caso di sistemi Windows, dopo aver scaricato dal sito di Zabbix il pacchetto precompilato andremo ad estrarre in una directory a nostra scelta il file agent, scegliendolo in base al tipo di architettura che andremo a monitorare (32 o 64 bit). Nella stessa directory dovrà essere creato ex novo un file di configurazione. La

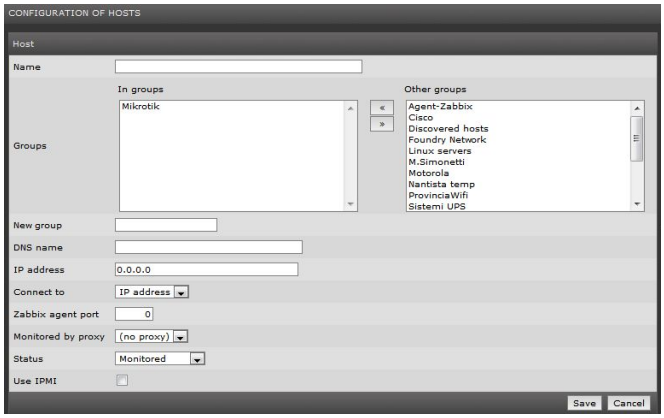


Fig. 7 Configurazione di un host.

configurazione base consiste in poche righe in cui vengono definiti il server a cui l'agent dovrà rispondere, il nome mnemonico del server e i parametri di fork. Per esempio memorizzeremo nel file agentd.conf il seguente contenuto

```
Server=my.zabbix.server.fqdn
Hostname=Windows Test
StartAgents=5
```

L'installazione viene eseguita a riga di commando, tramite un accesso a console Administrator (in caso contrario verrà visualizzato l'errore di accesso negato)

```
Zabbix_Agentd.exe -c agentd.conf -i
```

Il servizio verrà installato con parametri di esecuzione automatica, per eseguire il servizio la prima volta, se si vuole evitare il riavvio della macchina è sufficiente eseguire

```
Zabbix_Agentd.exe -c agentd.conf -s
```

5 ACL

Zabbix usa internamente un meccanismo di ACL (Access Control List) basato su utenti. Gli utenti possono essere raccolti eventualmente in gruppi, ai quali possono essere assegnati permessi per operare sul sistema. La dichiarazione di ruoli non è essenziale ai fini del funzionamento del sistema, ma diventa molto utile sia nella fase di reportistica (qualunque azione del sistema eseguita viene infatti loggata assieme all'utente), sia nella fase di escalation.

6 Hosts e Groups

L'inserimento degli host da controllare sarà sicuramente una delle prime attività con cui si avrà a che fare se si vuole utilizzare Zabbix. Un host è identificato da Zabbix con due parametri obbligatori: un'etichetta e un IP o un FQDN. Gli host possono essere raggruppati, al fine di applicare policy di controllo su interi gruppi piuttosto che su singoli hosts.

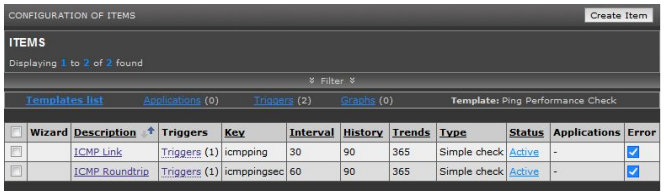


Fig. 8 Lista degli items attivi.

7 Items / Triggers / Actions

La distinzione tra items, triggers e actions è alla base per la comprensione del funzionamento di Zabbix nel suo processo di monitoring.

Gli items sono i dati grezzi raccolti dal sistema: tutte le caratteristiche che vogliamo controllare sono degli items. Questi possono essere frutto di misurazioni, come per esempio il tempo di roundtrip da un host all'altro, o la quantità di carico della macchina, oppure calcolati a partire da altri dati grezzi; per esempio si può calcolare la percentuale di spazio libero su disco a partire da dati quali la capacità totale e la capacità occupata.

I triggers descrivono una serie di condizioni, soddisfatte le quali Zabbix provvede ad entrare in uno stato di allerta. Nei trigger vengono inseriti dunque quei parametri che identificano eventuali problemi; ad esempio uno dei modi convenzionalmente utilizzati per dichiarare un host down è controllare il suo roundtrip dal server Zabbix. Se il valore massimo delle misurazioni negli ultimi 3 minuti è uguale a zero si suppone che l'host non sia raggiungibile. Al verificarsi di questa condizione nell'interfaccia di controllo Zabbix verrà visualizzato un nuovo problema. In questa fase non vengono inviate comunicazioni all'esterno, questo perché i triggers (come il loro nome fa intendere) non si occupano di eseguire una certa azione, ma fungono semplicemente da meccanismi di attivazione. I triggers hanno un campo descrittivo particolare, chiamato Severity, che descrive l'importanza del trigger stesso, consentendo azioni diverse a seconda di questo parametro.

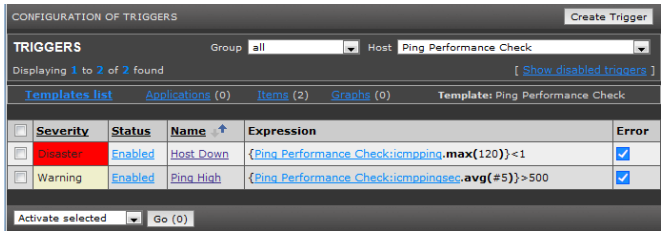


Fig. 9 Lista triggers attivi sul sistema.

Di base una action è semplicemente un'azione generica, che viene eseguita ogni volta che un trigger viene attivato. Una serie di parametri di configurazione permette di limitare l'esecuzione della action ai soli casi di interesse. Per esempio si possono configurare action che vengono eseguite soltanto nel caso in cui la gravità del

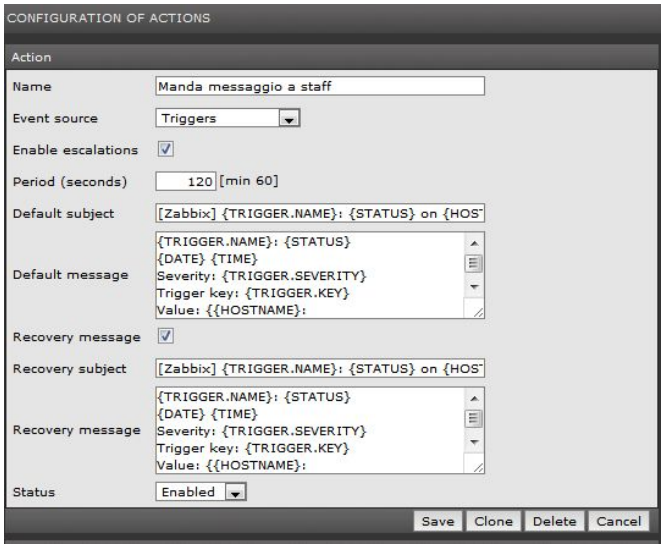


Fig. 10 Creazione di una action.

trigger sia di tipo "Disaster", o si può evitare che venga mandata in esecuzione nel caso il sistema sia posto in "Maintanence" mode, o ancora che venga ignorata nel caso in cui qualcuno abbia già eseguito l'operazione di Acknowledge del trigger.

Le actions sono in grado di eseguire due compiti: mandare un messaggio (attraverso i media configurati in Zabbix) o eseguire un comando (Zabbix server può eseguire un comando a nostra scelta). Il primo compito è quello che ci aspettiamo da un sistema di monitoring: a seguito di un trigger attivato un messaggio email viene inviato ad un gruppo di utenti Zabbix. Nel corpo del messaggio è possibile inserire tutti i dettagli riguardanti l'evento ricorrendo ad una serie di variabili che il sistema mette a disposizione. Il secondo caso fa sì che, a seguito dell'attivazione del trigger, Zabbix risponda con un comando lanciato sul server stesso, o, nel caso sia stato installato l'agent, direttamente sul server monitorato. Sul server "XYZ" il processo "QWE" non risponde da troppo tempo? Zabbix può tentare di riavviare il servizio, garantendo che il downtime sia minimo.

8 Escalations

Le escalations sono delle specializzazioni delle actions, estendono il loro comportamento introducendo il concetto di evoluzione temporale. Col passare del tempo l'azione esegue operazioni differenti. Supponiamo che un server sia down. Dopo 2 minuti viene inviata una mail allo staff IT che segnala il malfunzionamento. Questa mail verrà ripetuta ogni 60 minuti, o finché verrà dato l'Acknowledge dell'evento.

L'invio delle mail non è l'unica azione che Zabbix può eseguire, sebbene sia la più utilizzata. In casi particolari può essere più utile eseguire un comando. Zabbix mette a disposizione questa funzionalità tramite un nuovo tipo di operazione: "Remote Command".

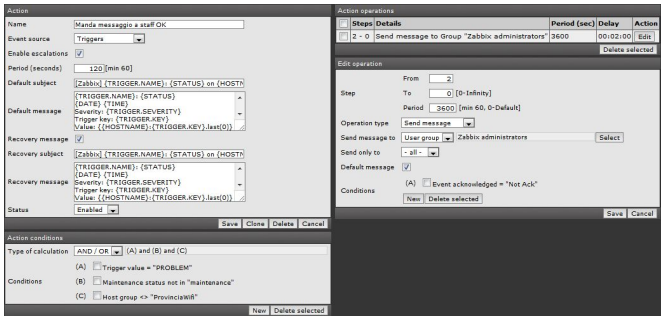


Fig. 11 Una action che prevede escalation.

Un esempio di questo tipo è visualizzabile nella prossima figura. Dopo 30 minuti dal rilevamento di un problema di severità “Disaster” (precedentemente definito nella configurazione degli host) sull’host “Zabbix Server” viene eseguito lo script /bin/send_sms.sh con parametri definiti a riga di comando come visto nello screenshot. Lo script si occupa di inviare ad un gateway mail-to-sms i contenuti dell’alert.

```
#!/bin/bash
echo $@ > /tmp/zabbix.txt
mail m2s@gatewaysms -s numerotelefono < /tmp/zabbix.txt
```

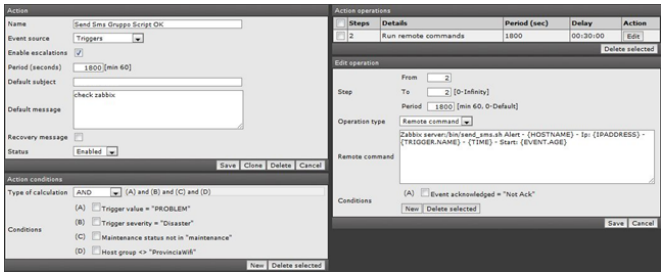


Fig. 12 Esecuzione di un comando remoto tramite Zabbix.

9 External Check

La presenza dell’agent su una macchina permette di monitorarne praticamente qualunque aspetto, attraverso le descrizioni che vengono inserite nel file zabbix_agentd.conf. Supponiamo di aver necessità di monitorare lo stato della replication master-slave tra due server MySQL. Abbiamo provveduto ad individuare i parametri che sarebbero stati necessari controllare. Una possibilità è quella di confrontare lo stato dei binary log files e verificare che il server slave non stia rimanendo indietro nell’applicazione degli stessi.

Sullo slave nello zabbix_agentd.conf aggiungeremo le righe

```
UserParameter=mysql.slaveposition,mysql -e “show slave status” -E
-uroot| grep Exec | cut -d ":" -f 2 | cut -d " " -f 2
```

Sul master nello zabbix_agentd.conf aggiungeremo le righe

```
UserParameter=mysql.masterposition,mysql   -e   “show   master   status”
-E -uroot | grep mysql-bin | cut -f 2
```

Gli items da monitorare in Zabbix a questo punto saranno mysql.slaveposition e mysql.masterposition, che sono le rispettive posizioni dei file bin log. Questi parametri, ad ogni controllo richiesto da Zabbix, assumeranno il valore dell’output dello script che viene descritto nei file di configurazione. Nel nostro caso un semplice intero.

Per poter capire se la replication sta avvenendo in maniera corretta andiamo a creare un terzo item di tipo Calculated che chiameremo BinlogDiff. I Calculated sono tipi astratti che derivano da operazioni eseguite su uno o più item reali, non necessariamente legati ad una singola macchina. In questo caso quello che ci servirà è la differenza tra la posizione del master e la posizione dello slave.

Un esempio di formula Calculated può essere il seguente

```
last(SQL Master:mysql.masterposition) - last(SQL
Slave:mysql.slaveposition)
```

Un valore pari a 0 indicherà che la replica sta funzionando a dovere. Un valore troppo alto indicherà dei problemi nella replication.

Il trigger dunque verrà lanciato a seguito del superamento di una soglia prestabilita proprio di BinlogDiff

10 Case study

10.1 Servizi Area

Nella nostra esperienza lavorativa i software descritti hanno un’importanza cruciale per il corretto svolgimento delle attività gestionali della rete. Sono utilizzati per tenere sotto controllo 71 switch e 21 access point, che sono presenti nei vari edifici e istituti attraverso dei semplici ICMP check, e quindi controllare basilarmente il loro corretto funzionamento. Abbiamo organizzato la struttura a gruppi, per capire dove fisicamente intervenire in caso di guasto. A seguito di malfunzionamento rilevato dal sistema esso invia una mail dopo 3 minuti dall’evento, nel caso esso non fosse stato nel frattempo risolto, e si occupa di inviarne un’altra ogni 15 minuti, fino alla risoluzione del problema o all’acknowledge dello stesso.

Discorso particolare invece è quello legato al monitoraggio delle macchine server presenti all’interno del CED; esse infatti hanno la possibilità di eseguire al proprio interno lo Zabbix Agent che, come precedentemente descritto, permette di tenere sotto monitoraggio non soltanto le risposte al ping (ICMP) ma anche lo stato generale del sistema mediante un gran numero di parametri, come la quantità di memoria o spazio disco disponibile, il carico della CPU, l’attività delle schede di rete fino al tipo e numero di processi in esecuzione. Ad esempio sul server web dell’Area della Ricerca viene tenuto sotto controllo lo stato del processo Apache, così da essere avvisati tempestivamente nel caso non dovesse rispondere

correttamente a seguito di qualche malfunzionamento o attacco informatico esterno.

10.2 Rete wireless geografica a 5Ghz

La rete 5ghz comprende più di 140 apparati da monitorare. Essi sono dispositivi eterogenei, e anche quelli che svolgono le stesse funzioni possono appartenere a costruttori differenti. Abbiamo dunque deciso dividere in gruppi i vari apparati sia secondo il costruttore che secondo il funzionamento. In tal modo ci siamo garantiti la possibilità di applicare template agli apparati secondo le nostre necessità. Abbiamo ovviamente mantenuto una template comune che si occupava di verificare i link, tramite un controllo icmp in cui si considerava solo il packetloss, e un altro che invece analizzava il round-trip, informazione utile per capire se i link radio erano soggetti a qualche tipo di deterioramento. Per tutti gli apparati che lo permettevano abbiamo monitorato via SNMP i parametri di utilizzo degli stessi, verificando che i valori di utilizzo cpu, memoria e spazio disco non oltrepassassero una soglia di allarme che non avrebbe garantito il corretto funzionamento. Altri dati vengono stockati senza che vi siano allarmi specifici per essi, ma se ne mantiene uno storico nel caso fosse necessario analizzarli a posteriori.

10.3 Tipologie di allarme

Per quanto concerne le action che vengono eseguite da Zabbix a seguito di evento, esse sono diverse a seconda della gravità del problema e della sfera di competenza. E' possibile infatti configurare gli invii di alert solo a specifici gruppi utente Zabbix. Nel caso più grave possibile, ovvero la perdita di link verso un determinato apparato, il sistema provvede ad inoltrare ai destinatari preimpostati una email dopo 2 minuti. Ne invierà un'altra ogni ora nel caso in cui il problema non dovesse essere risolto o riconosciuto tramite acknowledge. Oltre a questa azione, dopo 15 minuti dall'inizio del malfunzionamento viene eseguito uno script da Zabbix che tramite un gateway mailto-sms invia ai destinatari preimpostati un sms contenente le informazioni necessarie per identificare l'apparato che ha smesso di essere raggiungibile.

Nel caso di problemi meno seri, come un roundtrip time troppo elevato tra Zabbix e un apparato, il problema viene memorizzato e viene inviata una singola mail di segnalazione ai destinatari preimpostati. Il sistema provvede ad avvertire i destinatari che avevano ricevuto l'alert non appena riconosce l'avvenuta risoluzione del problema.

10.4 SLA

Un altro utilizzo che abbiamo fatto della piattaforma Zabbix è stato quello di conservare le statistiche relative al livello di servizio offerto dalla rete. Per ogni apparato posto sotto monitoraggio, infatti, è disponibile una rappresentazione statistica ad istogramma che evidenzia la disponibilità, espressa in percentuale sul tempo totale, di un singolo apparato. Tali numeri sono indispensabili per comprendere a fondo quanto sta funzionando una rete. Le maggiori aziende che offrono servizi ICT usano questi dati con molteplici finalità, promozionali ma anche legali, per avere una controprova della qualità dei servizi offerti alla clientela. Analogamente anche il servizio reti si è voluto dotare di uno strumento di valutazione statistica della bontà dei servizi offerti. Un down di un apparato di trasmissione radio comporta delle procedure più o meno onerose per il suo ripristino ed è impossibile annullare qualunque possibilità di fault sulla rete, tuttavia è possibile, tenendo traccia dei tempi di down e di ripristino, capire dove è più conveniente investire in rinnovo del parco hardware.

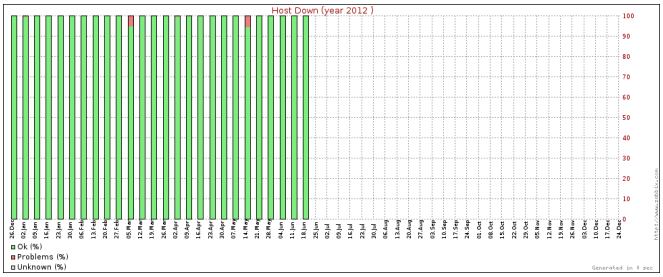


Fig. 13 Availability report.

10.5 Valutazioni

In definitiva Zabbix ha rappresentato, nel caso specifico dell'utilizzo che ne abbiamo fatto, uno strumento prezioso per il monitoraggio attivo e passivo di apparati di rete così come di servizi IT erogati all'interno dell'Area della Ricerca Roma 1 del CNR, offrendo ai sistemisti la possibilità di intervenire tempestivamente in caso di malfunzionamento di apparati o servizi grazie a un efficace sistema di alerting via SMS verso il reparto tecnico. La motivazione principale che ha portato Zabbix ad essere scelto è da imputare alla facilità di configurazione. L'appliance offre un sistema turn key già pronto, senza però impedire customizzazioni anche profonde.

Il sistema sarà di facile utilizzo anche al sistemista che si affaccia per la prima volta ad un sistema di monitoring. Un altro vantaggio è sicuramente la capacità di Zabbix di stockare i dati raccolti su MySQL e di renderli immediatamente consultabili. Considerando che essi sono memorizzati su una base di dati è possibile interfacciare al database un qualsiasi altro applicativo per un'analisi approfondita. A differenza di Cacti Zabbix non utilizza un database RRD. Questo comporta da un lato la necessità di archiviare i dati per evitare che il database cresca senza limiti, un'operazione chiamata house-keeping che può svolgere autonomamente, e dall'altro la persistenza di informazioni non aggregate, che mantengono quindi l'originalità dei dati. Sebbene Zabbix sia inferiore a Na-

gios per quel che concerne il sistema mappa, le sue capacità di alert basati su trigger consentono una gestione granulare e la configurazione di scenari complessi.

11 Considerazioni finali

Scegliere quale tipo di piattaforma monitoring utilizzare come strumento finale è stato oggetto di confronto all'interno del gruppo SRA dell'AdR Roma 1. Sebbene le funzionalità messe a disposizione da Zabbix fossero indubbie, alcune feature tipiche degli altri sistemi risultavano troppo peculiari per poter essere abbandonate.

Considerando che le risorse richieste da Nagios e Cacti sono molto inferiori a quelle richieste da Zabbix (che fa pesante uso di MySql), e che tutti e tre gli ambienti convivono in un ambiente virtualizzato la scelta finale è ricaduta sul mantenere attivi tutti e tre i sistemi. Di Nagios non ci si è voluti privare per via della potente mappa integrata. Considerando che la rete monitorata è di tipo gerarchico la mappa di Nagios svolge un lavoro eccellente nel mostrare dove è il problema e quanto è profondo. Un tale tipo di approccio consente anche a personale non tecnico di poter consultare la mappa e capire subito l'entità del problema.

Cacti da parte sua, avvalendosi di un database RRD garantisce la persistenza di informazioni per quanto concerne il traffico generato dai diversi apparati di rete, senza limite di tempo, al contrario di Zabbix che necessita di operazioni di house-keeping al fine di cancellare i dati più vecchi di una certa soglia. Seppure si perde in Cacti risoluzione nell'andare indietro nel tempo, i dati stockati saranno comunque consistenti, e permetteranno di capire come alcuni parametri si siano evoluti in un arco temporale anche lungo. A tutti gli effetti Cacti offre uno strumento eccellente per verificare lo stato di salute di una rete e l'analisi delle sue performance in un periodo temporale.

Il core del sistema di monitoring è comunque costituito da Zabbix, a lui è delegato il compito della reportistica. Zabbix, da noi provato nella versione 1.8.9, è risultato un prodotto maturo dal punto di vista delle features che offre, anche se pecca ancora per quanto riguarda l'interfaccia utente. La presenza di agent disponibili per una gran quantità di sistemi operativi e la possibilità di creare condizioni anche complesse per quanto riguarda l'invio degli alert fanno di Zabbix uno strumento indispensabile a disposizione degli amministratori di sistema.

Riferimenti

1 Nagios http://www.nagios.org/.

2 Cacti http://www.cacti.net/.

3 Zabbix http://www.zabbix.com/.