Una rassegna di strumenti per il monitoraggio e la manutezione di sistemi distribuiti

Alerting, monitoring, logging: cosa si intende *esattamente* con questi termini ?

M. Finelli
BioDec

BIODEC
Evolving ICT Infrastructures

Indice

Indice

BIODEC
Evolving ICT Infrastructures

Alerting, monitoring, logging

CHIARIRE I CONCETTI legati agli aspetti di *alerting*, *monitoring* e *logging*, che sono spesso sovrapposti o confusi, e fare una panoramica che mostri quali alternative di software libero conviene utilizzare in contesti differenti, coi pro e i contro di diverse soluzioni.

Perché è un tema di interesse ?

Perché, col crescere delle infrastrutture — grazie ai sistemi di virtualizzazione e alla disponibilità di soluzioni *cloud* — certi temi non sono solo di interesse per chi amministra un datacenter, ma diventeranno sempre più importanti anche per le aziende ordinarie.

Indice

Alcune definizioni

DEFINIAMO BREVEMENTE i concetti chiave: ci servirà per capire come si "incastrano" gli strumenti di cui parleremo nel seguito.

Si suppone di avere un *sistema* sotto osservazione, di cui ci interessa:

Alerting la gestione delle notifiche.

Monitoring la gestione delle misure.

Logging la gestione degli eventi.

Alcune definizioni

DEFINIAMO BREVEMENTE i concetti chiave: ci servirà per capire come si "incastrano" gli strumenti di cui parleremo nel seguito.

Si suppone di avere un *sistema* sotto osservazione, di cui ci interessa:

Alerting la gestione delle notifiche.

Monitoring la gestione delle misure.

Logging la gestione degli eventi.

Alcune definizioni

DEFINIAMO BREVEMENTE i concetti chiave: ci servirà per capire come si "incastrano" gli strumenti di cui parleremo nel seguito.

Si suppone di avere un *sistema* sotto osservazione, di cui ci interessa:

Alerting la gestione delle notifiche.

Monitoring la gestione delle misure.

Logging la gestione degli eventi.

Alcune definizioni

DEFINIAMO BREVEMENTE i concetti chiave: ci servirà per capire come si "incastrano" gli strumenti di cui parleremo nel seguito.

Si suppone di avere un *sistema* sotto osservazione, di cui ci interessa:

Alerting la gestione delle notifiche.

Monitoring la gestione delle misure.

Logging la gestione degli eventi.

Indice

Alerting

Un sistema d'allarme è un meccanismo che genera specifici messaggi (di allarme) e li recapita ad un determinato destinatario.
Un sistema d'allarme è composto da:

1. un generatore di allarmi,
2. il messaggio, che descrive l'allarme,
3. il destinatario del messaggio,
4. il sotto-sistema che si occupa della consegna del messaggio.

Un allarme in se è quindi un semplice messaggio.

Esempi

Esempi di allarme:

- la finestra che appare su un portatile quando la batteria va sotto una certa soglia percentuale,
- la notifica che ti manda *Dropbox* quando lo spazio si sta per esaurire,
- la notifica che ti manda una *mailing list* alla quale si è iscritti.

Indice

BIODEC
Evolving ICT Infrastructures

Cosa si intende con monitoraggio

Il monitoraggio è connesso al concetto di *misura*. Una misura è un valore numerico con un nome e l'istante in cui è stata effettuata. Una successione di misure è pertanto una serie temporale di valori numerici associati ad un'etichetta.

Cosa si intende con monitoraggio

Sui server Linux non esistono vere e proprie misure, se si eccettua i file di applicazioni come `sysstat`, `rrdtool` o simili: spesso le misure sono inserite dentro un file di log, e sono mescolate con informazioni semanticamente differenti.
Ad esempio una riga di un log di apache potrebbe riportare il tempo in ms di risposta, assieme agli altri dati del CustomLog.

Esempi

Torniamo ad un esempio ben noto e presente su ogni server su cui sia stato installato Apache:

```
109.234.57.170 - - [07/Jul/2011:09:34:26 +0200] GET /clienti-e-progetti/biocomp/biocomp-ups
HTTP/1.1 302 5367 - Mozilla/5.0 (X11; U; Linux x86_64; en-US; rv:1.9.2.18) Gecko/20110628
Ubuntu/10.10 (maverick) Firefox/3.6.18
```

Esempi

5367 è la *Size of response in bytes, excluding HTTP headers*

```
109.234.57.170 - - [07/Jul/2011:09:34:26 +0200] GET /clienti-e-progetti/biocomp/biocomp-ups
HTTP/1.1 302 5367 - Mozilla/5.0 (X11; U; Linux x86_64; en-US; rv:1.9.2.18) Gecko/20110628
Ubuntu/10.10 (maverick) Firefox/3.6.18
```

BIODEC
Evolving ICT Infrastructures

Esempi

5367 è la *Size of response in bytes, excluding HTTP headers*, **302** è lo *Status*

```
109.234.57.170 - - [07/Jul/2011:09:34:26 +0200] GET /clienti-e-progetti/biocomp/biocomp-ups
HTTP/1.1 302 5367 - Mozilla/5.0 (X11; U; Linux x86_64; en-US; rv:1.9.2.18) Gecko/20110628
Ubuntu/10.10 (maverick) Firefox/3.6.18
```

BIODEC
Evolving ICT Infrastructures

Indice

BIODEC
Evolving ICT Infrastructures

Logging - ovvero la gestione di un event log

Un *event log* è un insieme di voci, ognuna delle quali rappresenta un **evento** accaduto nel sistema sotto osservazione.
La **rappresentazione** o descrizione di un evento è ciò che permette di identificare in modo univoco quell'evento.

Evento

Si noti l'enfasi sul termine *evento*:

> *event - noun "a thing that happens"*

Se le voci di cui è composto il log non sono relative ad eventi, non è un log che ci interessa (ai fini di questo discorso).

La rappresentazione dell'evento

La *rappresentazione* è composta necessariamente da:

cosa un'identificazione, un tipo, un nome . . . qualcosa che identifica l'evento,

spazio dove è accaduto ovvero chi ha registrato l'evento,

tempo una collocazione temporale ovvero una marca temporale del momento della registrazione o dell'accadimento dell'evento (non è detto che la registrazione sia il momento dell'evento).

La rappresentazione dell'evento

La rappresentazione deve essere conosciuta a priori, se non esiste un modo di "decifrare" il messaggio, ovvero di ricostruire il "cosa", il "quando" e il "dove", allora non si tratta di un event log ma di altro.

La mancanza di una decodifica è un **antipattern**: lo schermo blu, il kernel panic.

La rappresentazione dell'evento

La rappresentazione deve essere conosciuta a priori, se non esiste un modo di "decifrare" il messaggio, ovvero di ricostruire il "cosa", il "quando" e il "dove", allora non si tratta di un event log ma di altro.
La mancanza di una decodifica è un **antipattern**: lo schermo blu, il kernel panic.

Esempi

I log classici che trovate su ogni server Linux:

```
Oct 20 07:49:08 kygerlitor anacron[30307]: Normal exit (1 job run)
Oct 20 09:33:04 kygerlitor dhclient: DHCPREQUEST of 192.168.137.101 on eth0 to 192.168.137.10
port 67
[Sun Oct 20 07:48:57 2013] [notice] Apache/2.2.22 (Ubuntu) configured -- resuming normal
operations
```

Questi **non** sono log ben formati, per varie ragioni:

```
(/usr/share/texmf-texlive/tex/generic/oberdiek/ifpdf.sty Package: ifpdf 2009/04/10 v2.0
Provides the ifpdf switch (HO)
[t=0.00] Locale en_US.UTF-8 matched to language en.
```

Indice

Come si incastrano i vari sistemi fra loro

Una cosa importante, che a volte genera confusione, è che i software di cui parleremo spesso svolgono diverse funzioni **tutte insieme**.

I.e. raccolgono log e ci fanno sopra un analisi volta a generare un alert, oppure riportano una misura relativa a *quando* si è rilevato un certo alert, eccetera.

Come si incastrano i vari sistemi fra loro

Una cosa importante, che a volte genera confusione, è che i software di cui parleremo spesso svolgono diverse funzioni **tutte insieme**.
I.e. raccolgono log e ci fanno sopra un analisi volta a generare un alert, oppure riportano una misura relativa a *quando* si è rilevato un certo alert, eccetera.

Modularità

Come principio generale sarebbe bene avere un sistema di **alert indipendente**, che faccia (bene) solo quello, e **che usi come componenti** i dati dei sistemi di **logging e di monitoring**.

I sistemi commerciali falliscono quasi tutti su questo punto: il *feature creep* — derivante dal fatto che sono comparati sul numero di funzionalità e non sulla qualità — è responsabile di creare software *moloch* che fanno tutto, ma **male**.

Modularità

Come principio generale sarebbe bene avere un sistema di **alert indipendente**, che faccia (bene) solo quello, e **che usi come componenti** i dati dei sistemi di **logging e di monitoring**.
I sistemi commerciali falliscono quasi tutti su questo punto: il *feature creep* — derivante dal fatto che sono comparati sul numero di funzionalità e non sulla qualità — è responsabile di creare software *moloch* che fanno tutto, ma **male**.

Indice

BIODEC
Evolving ICT Infrastructures

Software libero per . . .

Alerting	Nagios, Check_MK, OMD, Zenoss, Zabbix, . . .
Monitoring	PNP4Nagios, Cacti, Munin, Graphite, . . .
Logging	Syslog (e varianti), Logstash, Greylog2, . . .

Una nota sull'uso dei font

- il testo normale indica i programmi utilizzati in produzione, in BioDec,
- il *testo slanted* indica programmi valutati ma non usati (per ragioni diverse),
- il ~~testo cancellato~~ indica programmi che mi sento di sconsigliare (ma, come sempre *your mileage may vary*).

Indice

BIODEC
Evolving ICT Infrastructures

Everybody loves to hate Nagios

CI SARÀ UNA RAGIONE del fatto che Nagios e i suoi derivati siano al contempo fra gli strumenti **più utilizzati** (sia relativamente agli altri strumenti liberi, sia in assoluto, rispetto alle alternative commerciali),

. . . e fra i programmi **più bestemmiati** ?

Everybody loves to hate Nagios

CI SARÀ UNA RAGIONE del fatto che Nagios e i suoi derivati siano al contempo fra gli strumenti **più utilizzati** (sia relativamente agli altri strumenti liberi, sia in assoluto, rispetto alle alternative commerciali),
. . . e fra i programmi **più bestemmiati** ?

But still Nagios kicks ass !

Nagios ha dei *pro* molto “pesanti”:

- **stabile e performante** — scalabilità verso migliaia di *hosts*;
- in uso da **millenni**: nato nel 1996, rilasciato come progetto open *Netsaint* nel 1999, ribattezzato NAGIOS (*Nagios Ain't Gonna Insist On Sainthood*) per ragioni di trademark;
- **pletora** di plugin;
- **ecosistema** ancora attivo, e prolifico, nonostante l'età.

BIODEC
Evolving ICT Infrastructures

But still Nagios kicks ass !

Nagios ha dei *pro* molto “pesanti”:

- **stabile e performante** — scalabilità verso migliaia di *hosts*;
- in uso da **millenni**: nato nel 1996, rilasciato come progetto open *Netsaint* nel 1999, ribattezzato NAGIOS (*Nagios Ain't Gonna Insist On Sainthood*) per ragioni di trademark;
- **pletora** di plugin;
- **ecosistema** ancora attivo, e prolifico, nonostante l'età.

But still Nagios kicks ass !

Nagios ha dei *pro* molto “pesanti”:

- **stabile e performante** — scalabilità verso migliaia di *hosts*;
- in uso da **millenni**: nato nel 1996, rilasciato come progetto open *Netsaint* nel 1999, ribattezzato NAGIOS (*Nagios Ain't Gonna Insist On Sainthood*) per ragioni di trademark;
- **pletora** di plugin;
- **ecosistema** ancora attivo, e prolifico, nonostante l'età.

But still Nagios kicks ass !

Nagios ha dei *pro* molto “pesanti”:

- **stabile e performante** — scalabilità verso migliaia di *hosts*;
- in uso da **millenni**: nato nel 1996, rilasciato come progetto open *Netsaint* nel 1999, ribattezzato NAGIOS (*Nagios Ain’t Gonna Insist On Sainthood*) per ragioni di trademark;
- **pletora** di plugin;
- **ecosistema** ancora attivo, e prolifico, nonostante l’età.

Ed è proprio l'ecosistema dei progetti che stanno attorno a Nagios, che permette di sopperire ai lati negativi:

- usabilità terribile;
- ma **terribile veramente**.

Ah, e se usate un database per la gestione dei dati, Nagios supporta solo MySQL.

Ed è proprio l'ecosistema dei progetti che stanno attorno a Nagios, che permette di sopperire ai lati negativi:

- usabilità terribile;
- ma **terribile veramente**.

Ah, e se usate un database per la gestione dei dati, Nagios supporta solo MySQL.

Ed è proprio l'ecosistema dei progetti che stanno attorno a Nagios, che permette di sopperire ai lati negativi:

- usabilità terribile;
- ma **terribile veramente**.

Ah, e se usate un database per la gestione dei dati, Nagios supporta solo MySQL.

& Co.

Icinga è nato come un *fork* di Nagios, per lo meno la parte *core*. Supporta PostgreSQL, ha un'interfaccia web migliore, è compatibile con i check e le configurazioni Nagios (per cui passare ad Icinga è relativamente indolore).

Check_MK è una "collection of extensions for the IT-Monitoring-Kernel of Nagios and together with this, and ideally also with PNP4Nagios and NagVis constitutes a complete IT-Monitoring-System." Contiene alcune migliorie effettive come Livestatus e Check_MK Multisite. Anche l'interfaccia web è migliore di quella di Nagios.

& Co.

Shinken è un'alternativa a Nagios, che enfatizza gli aspetti di HA e di essere distribuita (quindi con più componenti indipendenti). Non si integra bene con i *check* di Check_MK (test di qualche mese fa).

OMD è una distribuzione (*Open Monitoring Distribution*, ovvero "(it) bundles Nagios together with many important addons and can easily be installed on every major Linux distribution."
Pertanto OMD è un modo comodo per avere, senza problemi di dipendenza, già pacchettizzati tutti i sistemi di cui sopra, e altro (Thruk, Docuwiki eccetera).

Altri sistemi

Il principio di modularità precedentemente espresso è la ragione per cui, dopo averne fatto esperienza in passato, abbiamo deciso di non utilizzare più certi strumenti come ~~Zabbix~~ e ~~Zenoss~~, che vogliono fare troppe cose insieme.

Un esempio

DEMO TIME I

Indice

Measure ! Measure ! Measure everywhere !

UNA MISURA è un valore numerico con un nome e l'istante in cui è stata effettuata. Una successione di misure è pertanto una serie temporale di valore numerici associati ad un'etichetta.

Ergo, una serie temporale la si fruisce (legge, analizza, studia eccetera) **visualizzandone il grafico relativo**.

Measure ! Measure ! Measure everywhere !

UNA MISURA è un valore numerico con un nome e l'istante in cui è stata effettuata. Una successione di misure è pertanto una serie temporale di valore numerici associati ad un'etichetta.
Ergo, una serie temporale la si fruisce (legge, analizza, studia eccetera) **visualizzandone il grafico relativo**.

Measure ! Measure ! Measure everywhere !

Morale: se i vostri strumenti *di monitoraggio* non vi permettono di manipolare **efficacemente** dei grafici . . .

BUTTATELI NEL RUSCO !

Measure ! Measure ! Measure everywhere !

Morale: se i vostri strumenti *di monitoraggio* non vi permettono di manipolare **efficacemente** dei grafici . . .

BUTTATELI NEL RUSCO !

Measure ! Measure ! Measure everywhere !

Come si realizza un sistema di misura:

Route	collectd, statsd, *metricsd*,
Store	graphite (whisper),
Aggregate	graphite (carbon),
Visualize	graphite-web,
Analyze	*sensu*,
Alert	Nagios / Icinga, o altro analogo.

Per intenderci, un sistema “classico” ha tutti i componenti svolti da Nagios, con Cacti / ~~Pnp4Nagios~~ o ~~Munin~~ come sistema di visualizzazione.

BIODEC
Evolving ICT Infrastructures

Un esempio

Demo Time II

Indice

Andare oltre `tail -f /var/log/syslog`

UN LOG è diverso da un sistema di misura, perché, sebbene abbia la medesima connotazione di serie temporale, quanto tracciato sono *eventi* e non dati numerici.

Andare oltre `tail -f /var/log/syslog`

Come si realizza un sistema di logging:

Route	syslog-ng, rsyslog, *logstash*,
Store	elasticsearch (mongodb),
Aggregate	graylog2,
Visualize	graylog2, *kibana*,
Analyze	graylog2, *kibana*,
Alert	Nagios / Icinga, o altro analogo.

Per intenderci, un sistema “classico” ha tutti i componenti svolti da syslog, con programmi come logwatch o simili per farne l’analisi. Oppure soluzioni proprietarie, tipicamente molto costose.

Un esempio

DEMO TIME III

Thanks & see you soon . . .

Grazie dell'attenzione !

IDI2014 Incontro DevOps Italia 2014 — i prossimi annunci saranno su `http://blog.biodec.com` (poi avremo una pagina dedicata su `devops.it`).

Quando e dove Probabilmente a Bologna, o comunque in zona, nella seconda metà di febbraio.

Registratevi tra qualche settimana.

BIODEC
Evolving ICT Infrastructures

Thanks & see you soon . . .

Grazie dell'attenzione !

IDI2014 Incontro DevOps Italia 2014 — i prossimi annunci saranno su `http://blog.biodec.com` (poi avremo una pagina dedicata su `devops.it`).

Quando e dove Probabilmente a Bologna, o comunque in zona, nella seconda metà di febbraio.

Registratevi tra qualche settimana.

BIODEC
Evolving ICT Infrastructures

Thanks & see you soon . . .

Grazie dell'attenzione !

IDI2014 Incontro DevOps Italia 2014 — i prossimi annunci saranno su `http://blog.biodec.com` (poi avremo una pagina dedicata su `devops.it`).

Quando e dove Probabilmente a Bologna, o comunque in zona, nella seconda metà di febbraio.

Registratevi tra qualche settimana.

BIODEC
Evolving ICT Infrastructures

Thanks & see you soon . . .

Grazie dell'attenzione !

IDI2014 Incontro DevOps Italia 2014 — i prossimi annunci saranno su `http://blog.biodec.com` (poi avremo una pagina dedicata su `devops.it`).

Quando e dove Probabilmente a Bologna, o comunque in zona, nella seconda metà di febbraio.

Registratevi tra qualche settimana.

*licenza della presentazione:
`http://creativecommons.org/licenses/by-sa/3.0/`

BIODEC
Evolving ICT Infrastructures