

DevOps e Reti Bayesiane

Gregorio Piccoli

Padova, 12 Dicembre 2018

ZUCCHETTI SpA

Prima software house italiana, per storia e dimensione:

- 40 anni
- 600M Euro di fatturato
- 4500 persone
- 130000 clienti attivi
- Presenza in 50 paesi

... siamo una “software house”, non un “system integrator” ...

ZUCCHETTI SpA

Tre grandi divisioni responsabili dei prodotti:

1. Gestionali: contabilità, acquisti, vendite, magazzino, ...
2. Risorse umane: paghe, stipendi, presenze, controllo accessi, ...
3. Fiscale: dichiarazione dei redditi, 730, conservazione sostitutiva, ...

e con molti prodotti a contorno: Business Intelligence, robotica, IoT, sicurezza,
...

Facciamo applicazioni di Business!

Fatturazione elettronica

A partire dal 1 Gennaio 2019 tutte le fatture emesse in Italia dovranno essere spedite al MEF (Ministero dell' Economia e delle Finanze) per poter essere considerate valide.

Ogni anno vengono emesse circa 1.300.000.000 fatture, che diventano 2.600.000.000 di documenti da spedire tra chi emette e chi riceve .

La Zucchetti offre ai propri clienti un il servizio di spedizione diretta dai propri gestionali.

Data la nostra quota di mercato ci aspettiamo di trattare circa 500.000.000 di documenti.

Fatturazione elettronica

La Zucchetti è contemporaneamente produttrice del software e gestore del servizio.

La fabbrica è in contatto con le operazioni sul campo, questo è lo scenario indicato dal termine “DevOps”, contrazione delle due parole “Development” ed “Operations”.

Cosa cambia quando un unico attore
si occupa sia del “development” che
delle “operations”?

DevOps

“Development” + “Operations”: un unico attore che svolge i due ruoli.

Lo sviluppo opera per un sistema specifico e che controlla in ogni sua componente, non per una “universalità” non ben definita.

L’operatività quotidiana deve garantire che i sistemi siano attivi e funzionanti, ma anche segnalare i possibili miglioramenti allo sviluppo.

I sistemisti devono conoscere cosa potrebbe essere in grado di realizzare la “fabbrica”, i programmatori devono diventare un po’ sistemisti.

DevOps

Ciclo tradizionale: analisi, progettazione, sviluppo e test.

Si arriva quindi al rilascio: normalmente sono pacchetti distribuiti ai clienti.

Il software, uscito dalla produzione, viene gestito dai sistemisti.

Gli sviluppatori ricevono dei feedback dal servizio di assistenza, i sistemisti devono far funzionare al meglio quello che ricevono.



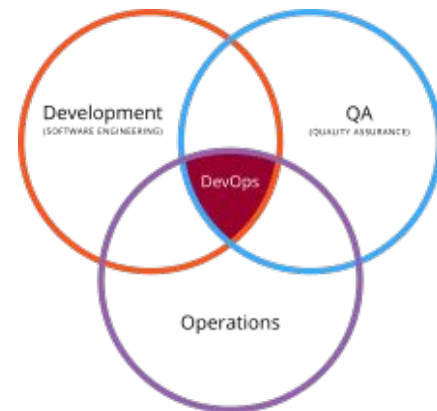
DevOps

Nel caso “DevOps” il programma non esce dalla “fabbrica” e viene gestito da sviluppatori e sistemisti.

I sistemisti possono richiedere modifiche al prodotto, non sono costretti ad usare le sole “leve” delle configurazioni dell’hardware.

Gli sviluppatori ricevono un flusso continuo di informazioni sullo stato della procedura e sul suo funzionamento.

Il “Responsabile della qualità” diventa molto più importante.



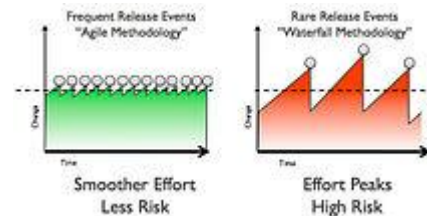
DevOps

La velocità di rilascio, data dal controllo dell'unica installazione, porta a release molto più frequenti.

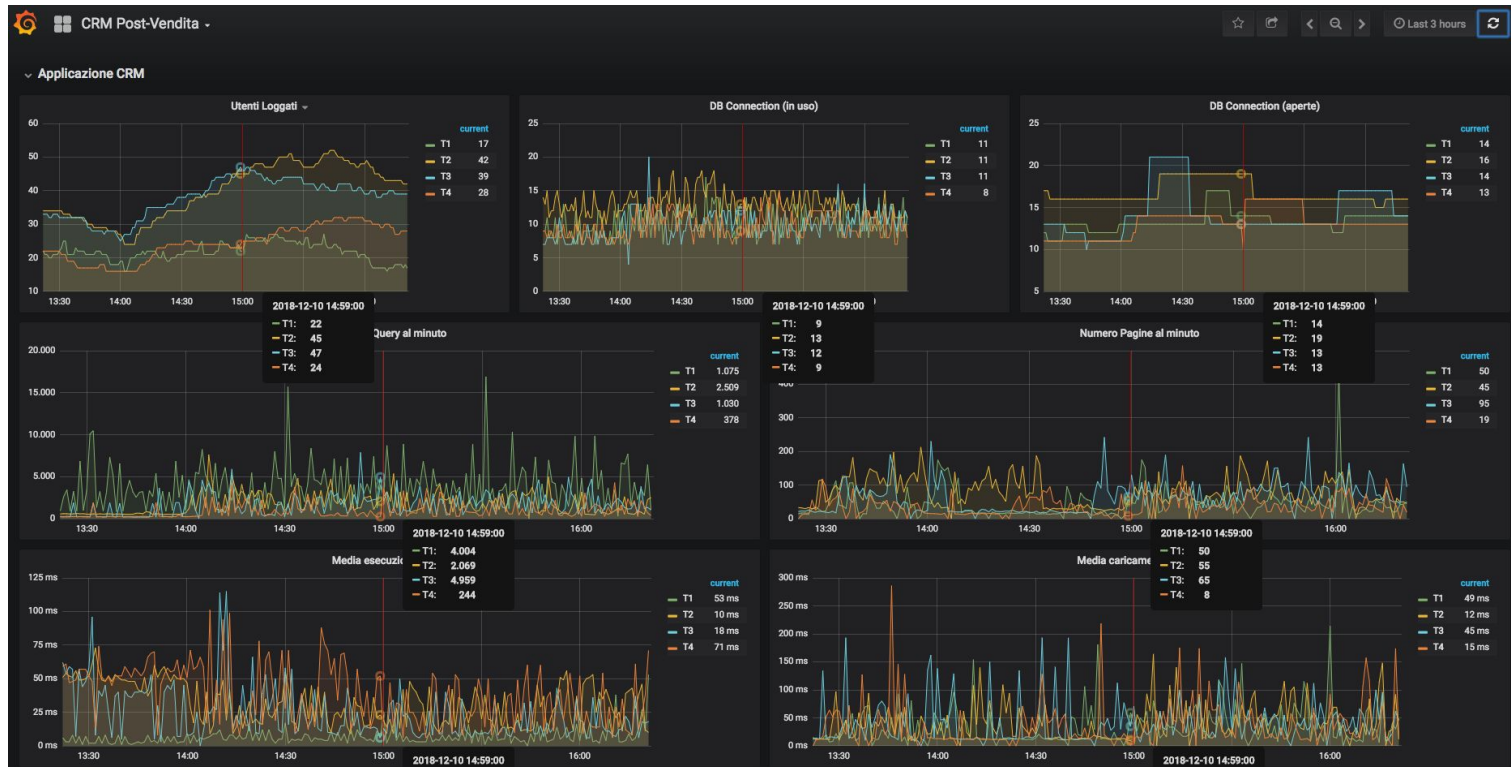
Quindi i team devono organizzarsi con cicli di vita molto più rapidi, il che significa quasi sempre adottare metodologie agili.

La possibilità di rilasciare velocemente diventa essenziale: si possono fare più release al giorno!

Ci si scontra con la continuità del servizio (SLA).



GRAFANA



GRAFANA

Il monitoraggio delle applicazioni diventa particolarmente importante perché ora ha due compiti:

1. Controllare la salute del sistema e verificare che le performance siano entro i livelli di SLA (Service Level Agreement)
2. Identificare quali punti deboli possono essere corretti dal team di sviluppo e fornire elementi per definire la scala delle priorità nelle migliorie e nelle nuove implementazioni

Si deve monitorare ogni cosa: le pagine servite, il numero degli utenti, la loro provenienza, l'uso della CPU, il disco, la rete ...

GRAFANA

Esempio di pagina di “alert”.

Se qualcosa è in condizione critica e richiede l’attenzione del sistemista l’indicatore prende un colore diverso.

Nel caso di situazioni estreme scattano avvisi come e-mail ai responsabili.

Reperibilità 24/7 dei sistemisti.

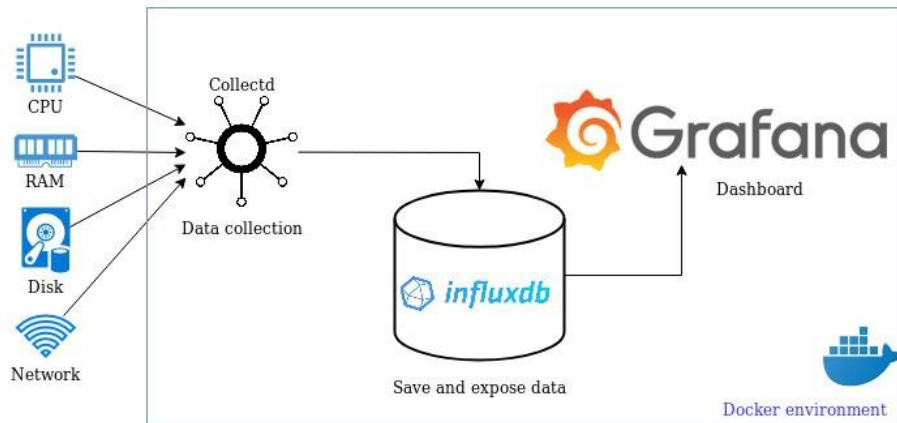


GRAFANA

L'architettura del “sistema” Grafana prevede degli agenti che raccolgono delle misure dalle varie macchine.

I dati vengono spediti a una istanza di InfluxDB, un database specializzato in serie temporali.

Grafana interroga InfluxDB presentando dashboard di grafici e notificando allarmi.

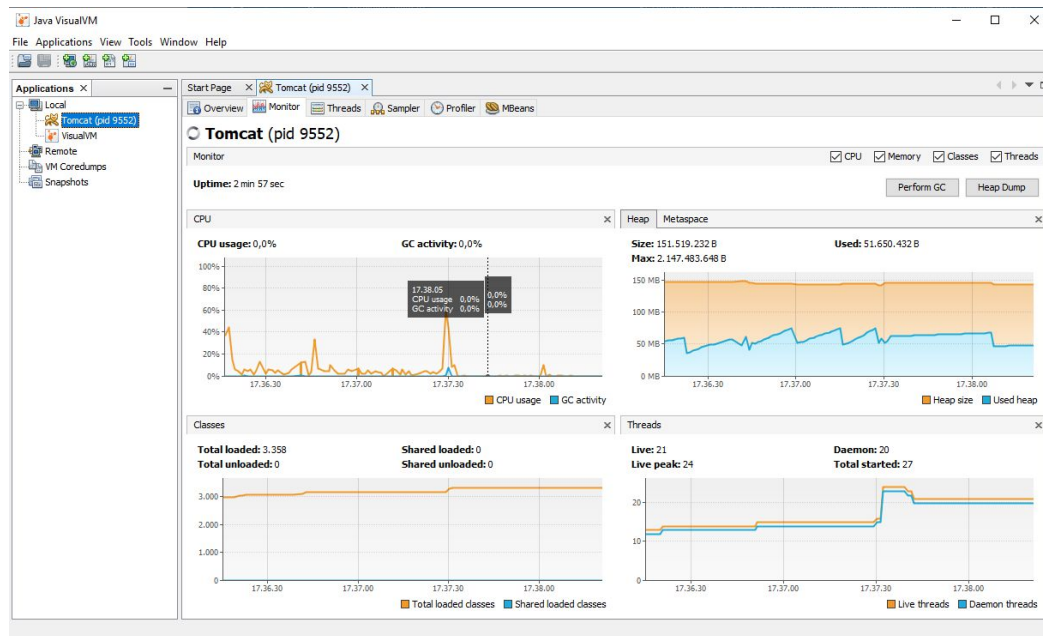


Java JMX

In procedure basate su Java i “sensori” sono spesso degli M-bean.

JMX è lo standard per il monitoraggio della Java Virtual Machine.

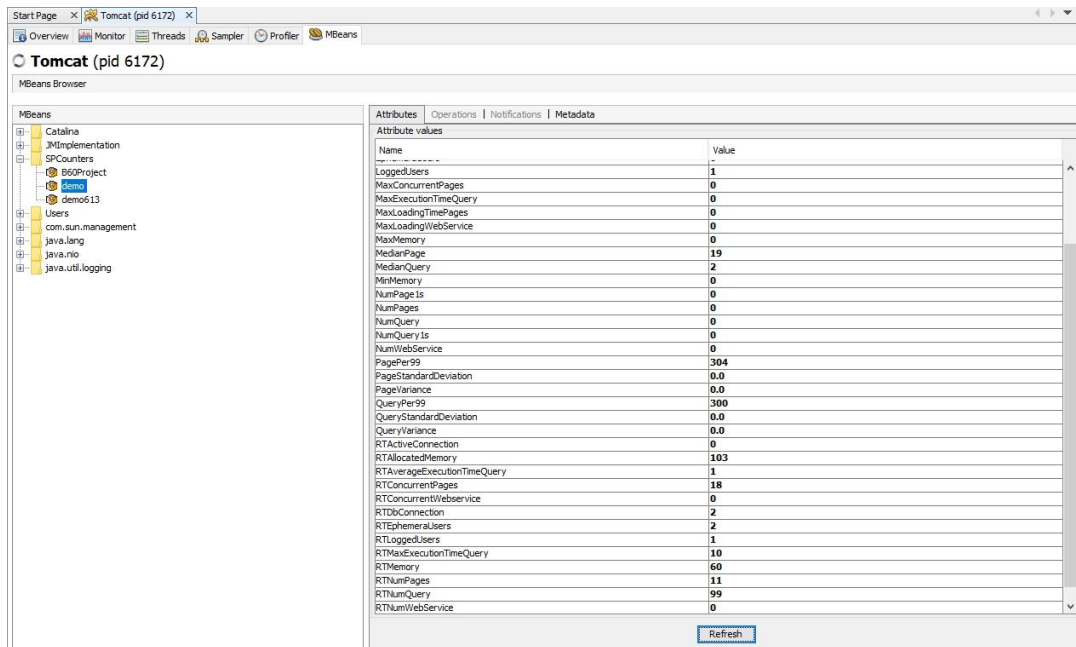
Nella distribuzione ufficiale di Java troviamo VisualJVM e JMC (Java mission control).



Java JMX

Le applicazioni possono riportare dei dati ai sistemisti creando degli M-bean per esporre il loro stato interno.

Abbiamo sviluppato dei bean per il numero di utenti loggati, le connessioni al database, lo stato del pooler, i tempi di esecuzione delle query ...

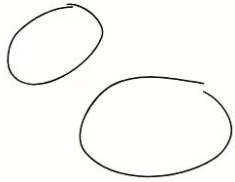


The screenshot shows the JMX console for Tomcat (pid 6172). The 'MBeans' tree on the left shows the hierarchy: Catalina > J2EEImplementation > SPCounters > demo613. The 'Attributes' tab is active, displaying a table of metrics and their values.

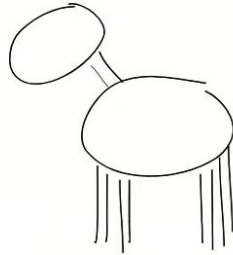
| Name | Value |
|-----------------------------|-------|
| ----- | ----- |
| LoggedUsers | 1 |
| MaxConcurrentPages | 0 |
| MaxExecutionTimeQuery | 0 |
| MaxLoadingTimePages | 0 |
| MaxLoadingWebService | 0 |
| MaxMemory | 0 |
| MaxOpenPage | 19 |
| MedianQuery | 2 |
| MinMemory | 0 |
| NumPagesIs | 0 |
| NumPages | 0 |
| NumQuery | 0 |
| NumQueryIs | 0 |
| NumWebService | 0 |
| PagePer99 | 304 |
| PageStandardDeviation | 0.0 |
| PageVariance | 0.0 |
| QueryPer99 | 300 |
| QueryStandardDeviation | 0.0 |
| QueryVariance | 0.0 |
| RTActiveConnection | 0 |
| RTAllocatedMemory | 103 |
| RTAverageExecutionTimeQuery | 1 |
| RTConcurrentPages | 18 |
| RTConcurrentWebService | 0 |
| RTDbConnection | 2 |
| RTEphemeralUsers | 2 |
| RTLoggedUsers | 1 |
| RTMaxExecutionTimeQuery | 10 |
| RTMemory | 60 |
| RTNumPages | 11 |
| RTNumQuery | 99 |
| RTNumWebService | 0 |

Performance delle applicazioni in architettura Web

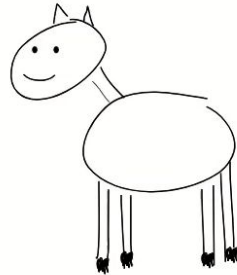
Fixing web performance is as easy as drawing a horse



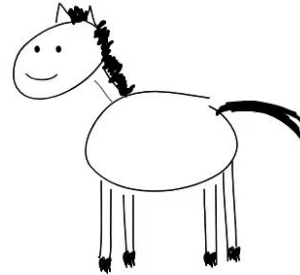
1. DRAW 2 CIRCLES



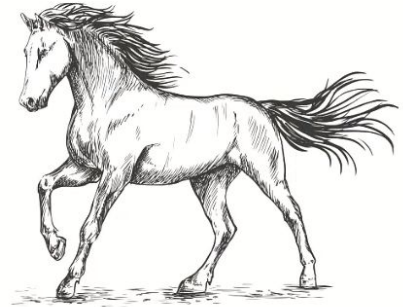
2. DRAW THE LEGS



3. DRAW THE FACE



4. DRAW THE HAIR

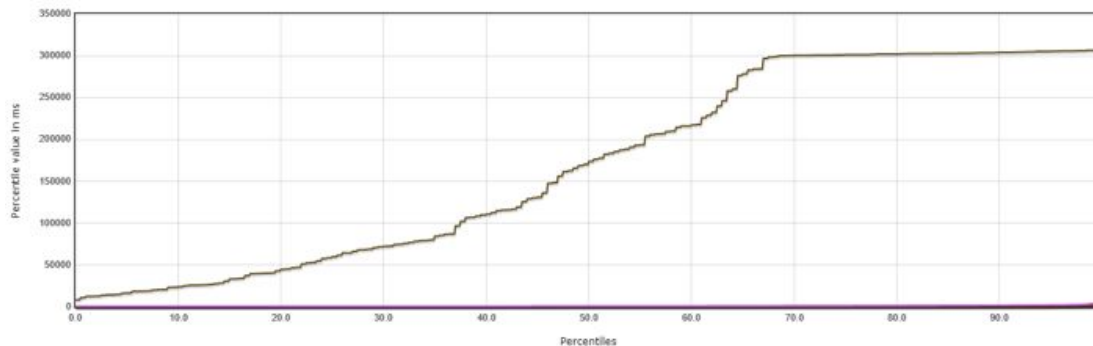
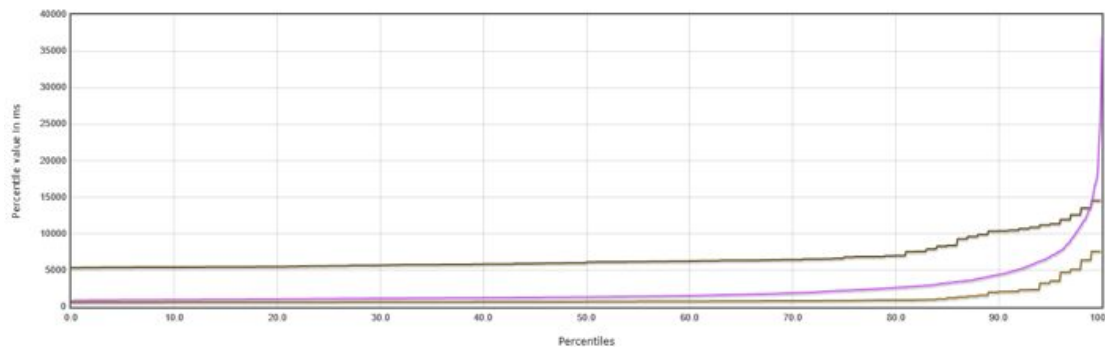


5. ADD SMALL DETAILS

Java JMeter

“Prevenire è meglio di curare”: i sistemi possono essere messi artificialmente sotto stress.

JMeter è uno strumento della Apache Foundation ben integrato nello stack Java per verificare il comportamento delle applicazioni Web.

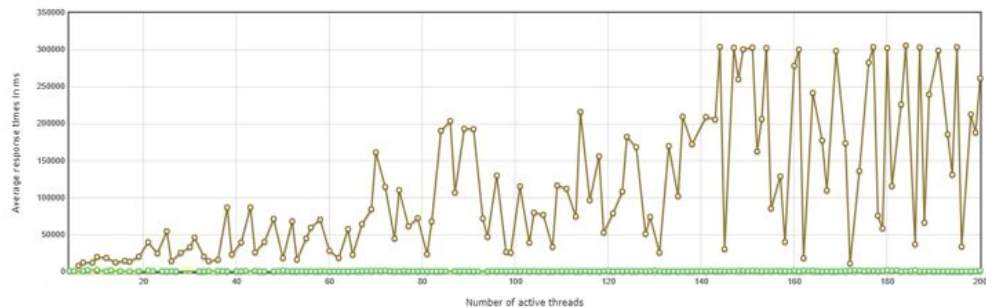
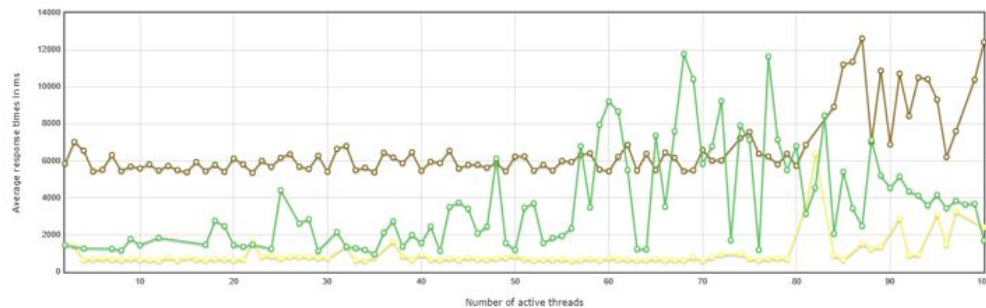


Java JMeter

Nelle immagini a lato si vede la risposta del sistema all'aumentare dei thread, quindi degli utenti concorrenti.

L'immagine a fianco mostra un sistema che fallisce.

Le immagini della slide precedente mostrano la tipica distribuzione dei tempi di esecuzione.



Reti Bayesiane

Thomas Bayes (Londra, 1702 – Royal Tunbridge Wells, 17 aprile 1761) è stato un matematico e ministro presbiteriano britannico.

Deve la sua fama ai suoi studi nel campo della matematica e della filosofia.

E' noto soprattutto nella statistica per il teorema di Bayes, vertente sulla probabilità condizionata, pubblicato postumo nel 1763.



Teorema di Bayes

THE PROBABILITY OF "B"
BEING TRUE GIVEN THAT
"A" IS TRUE

THE PROBABILITY
OF "A" BEING
TRUE

$$P(A|B) = \frac{P(B|A) P(A)}{P(B)}$$

THE PROBABILITY
OF "A" BEING TRUE
GIVEN THAT "B" IS
TRUE

THE PROBABILITY
OF "B" BEING
TRUE



Teorema di Bayes

$$P(H|E) = \frac{P(H) * P(E|H)}{P(E)}$$

Prior Probability

Likelihood of the evidence 'E' if the Hypothesis 'H' is true

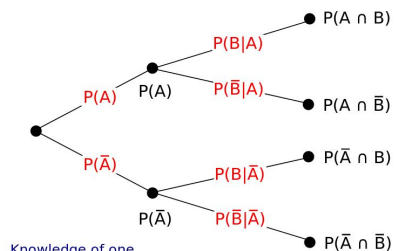
Posterior Probability of 'H' given the evidence

Priori probability that the evidence itself is true



Teorema di Bayes

La regola di Bayes permette di calcolare le probabilità condizionate passando da un diagramma all'altro.

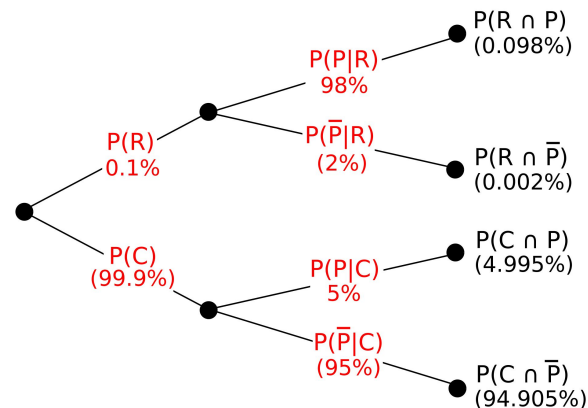
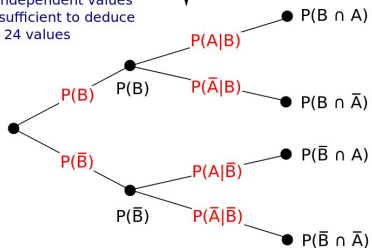


Knowledge of one diagram is sufficient to deduce the other

Use Bayes' Theorem to convert between diagrams

$$P(\alpha|\beta) P(\beta) = P(\alpha \cap \beta) = P(\beta|\alpha) P(\alpha)$$

Knowledge of any 3 independent values is sufficient to deduce all 24 values

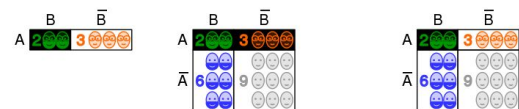


Teorema di Bayes

Il teorema di Bayes si verifica facilmente con una ispezione visuale.

Non è un metodo con il rigore matematico, ma dimostra la forza naturale dell'idea sottostante.

| Number of occurrences | case B | case \bar{B} | sum |
|-----------------------|--------|----------------|-----|
| condition A | 2 | 3 | 5 |
| condition \bar{A} | 6 | 9 | 15 |
| sum | 8 | 12 | 20 |



$$P(B, \text{ given } A) \cdot P(A) = P(B|A) \cdot P(A)$$

$$\frac{2}{2+3} \cdot \frac{2+3}{2+3+6+9} = \frac{2}{2+3+6+9}$$



$$P(A, \text{ given } B) \cdot P(B) = P(A|B) \cdot P(B)$$

$$\frac{2}{2+6} \cdot \frac{2+6}{2+3+6+9} = \frac{2}{2+3+6+9}$$

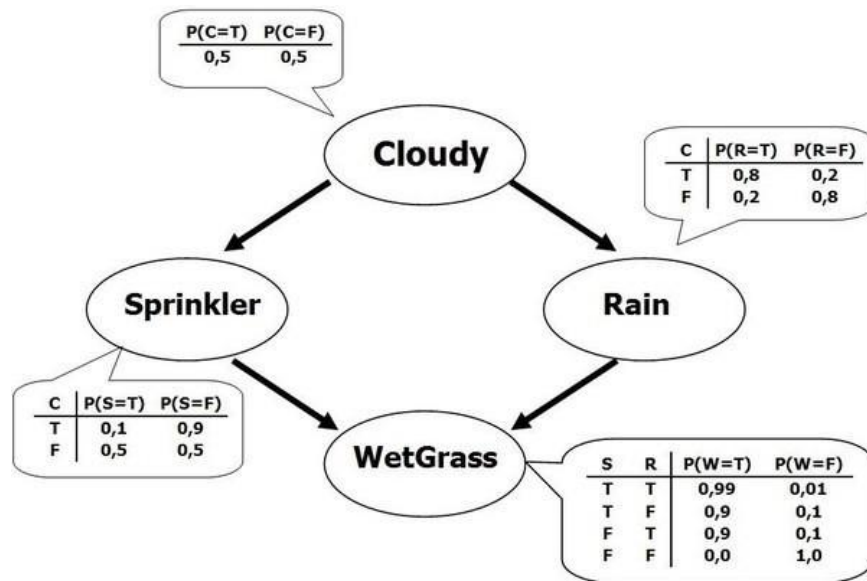
$$P(A|B) \cdot P(B) = P(B|A) \cdot P(A)$$

$$\therefore P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

Reti Bayesiane

Le reti bayesiane sono un modello grafico probabilistico che rappresenta una serie di variabili stocastiche e le loro dipendenze condizionali utilizzando un grafo diretto aciclico (DAG).

Le reti bayesiane permettono di rappresentare la conoscenza degli esperti incorporando le condizioni “a priori”.

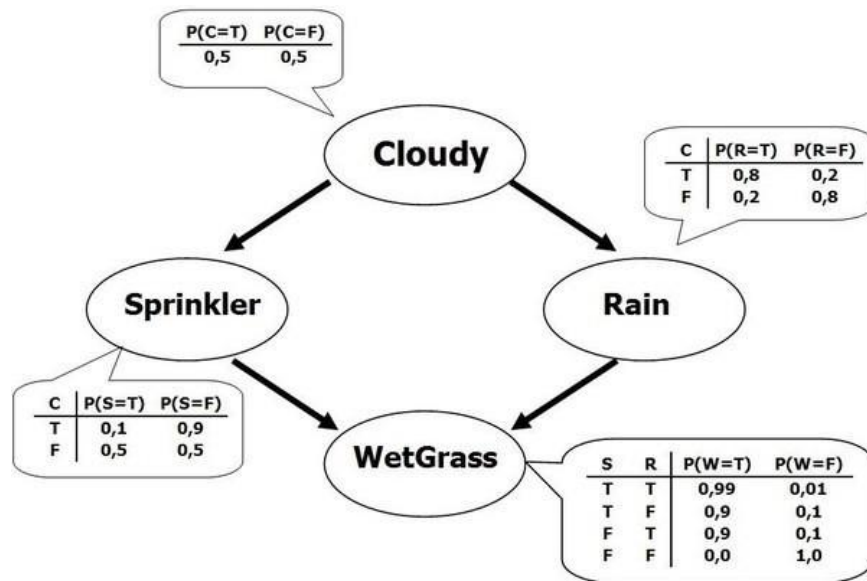


Reti Bayesiane

Apprendimento ed inferenza nelle reti bayesiane:

1. Inferenza di variabili non osservabili
2. Apprendimento dei parametri
3. Apprendimento della struttura

A noi interessa il primo caso: costruire il DAG con l'aiuto degli esperti di un settore, collegare alcune variabili ai sensori, inferire le altre.



Definizione della rete bayesiana

La rete definisce degli stati collegati tra loro.

In questo esempio:

A: viaggio in Asia

T: tubercolosi

S: fumatore

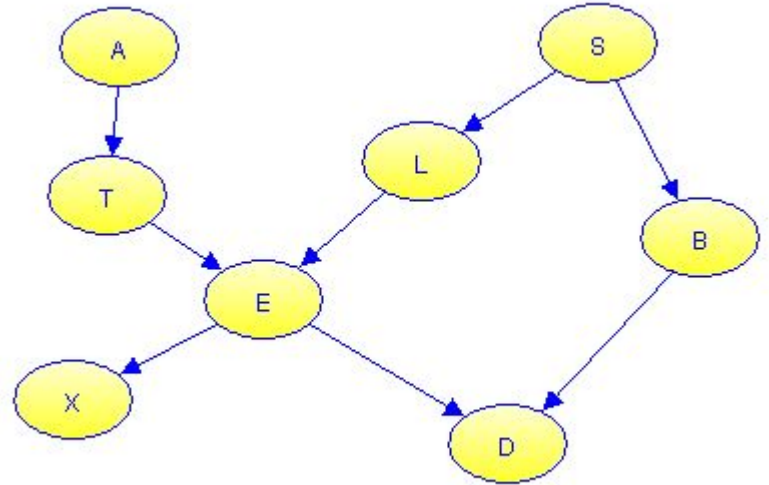
L: tumore al polmone

B: bronchite

E: tumore o tubercolosi

X: radiografia con macchie

D: dispnoico



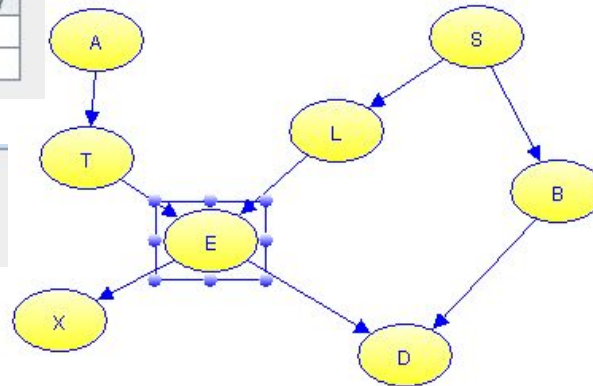
Definizione della rete bayesiana

Tabelle delle
probabilità
condizionate

| Table | | | | |
|-------------------------------|-------|----|-----|----|
| Create uniform distribution ▼ | Apply | | | |
| T | yes | no | | |
| L | yes | no | yes | no |
| yes | 1 | 1 | 1 | 0 |
| no | 0 | 0 | 0 | 1 |

| State | Probability |
|------------|-------------|
| yes | 0,01 |
| no | 0,99 |

| A | yes | no |
|------------|------|------|
| yes | 0,05 | 0,01 |
| no | 0,95 | 0,99 |

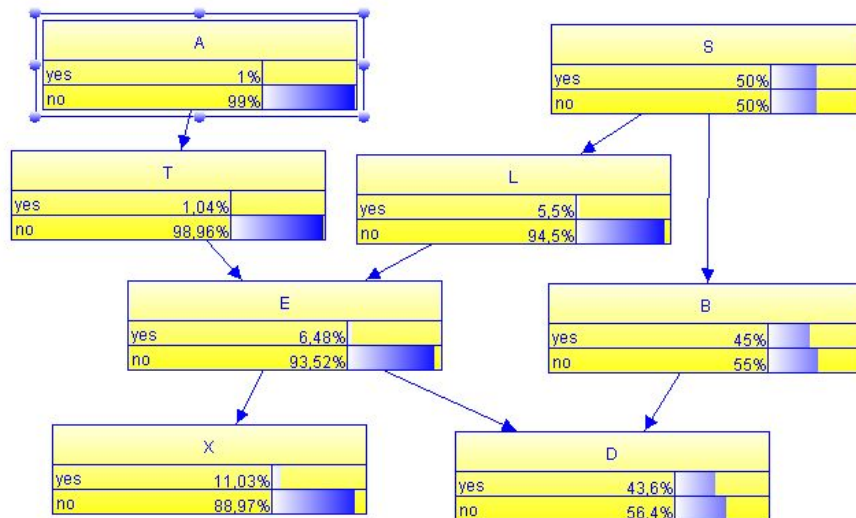


Inferenze nelle reti bayesiane

Propagazione delle probabilità.

Le probabilità degli stati nei nodi vengono calcolate a partire dalle probabilità dei nodi precedenti e dalle tabelle delle probabilità condizionate per il nodo stesso.

Possiamo vedere la probabilità che il paziente abbia la bronchite o sia dispnoico.



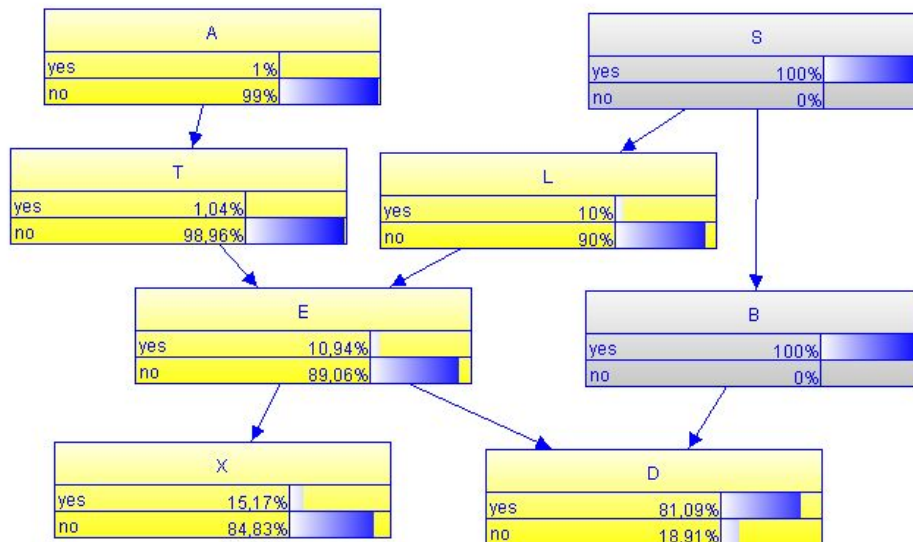
Inferenze nelle reti bayesiane

In questo caso:

S: fumatore

B: con bronchite

è aumentata la probabilità che
sia malato anche di altre
malattie e che sia dispnoico



Inferenze nelle reti bayesiane

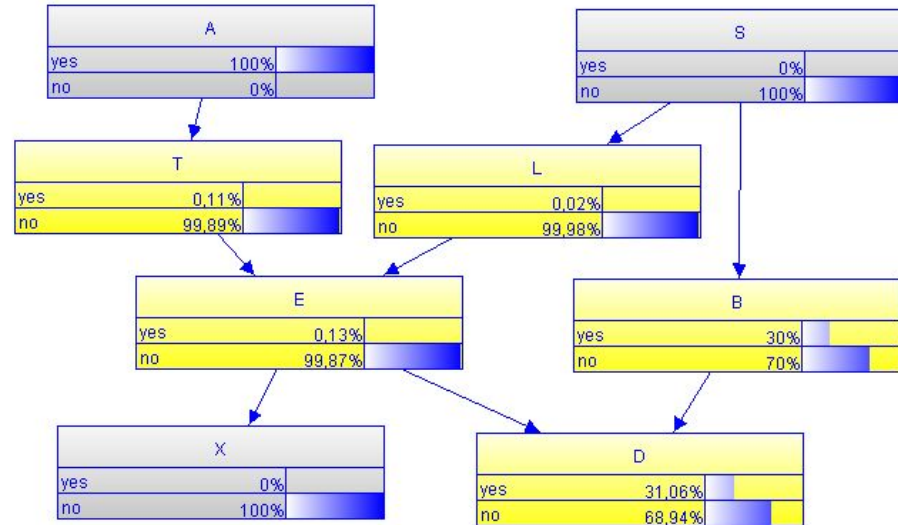
In questo caso:

A: viaggio in Asia

S: non fumatore

X: radiografia pulita

sono aumentate tutte le
probabilità che il soggetto sia
sano.



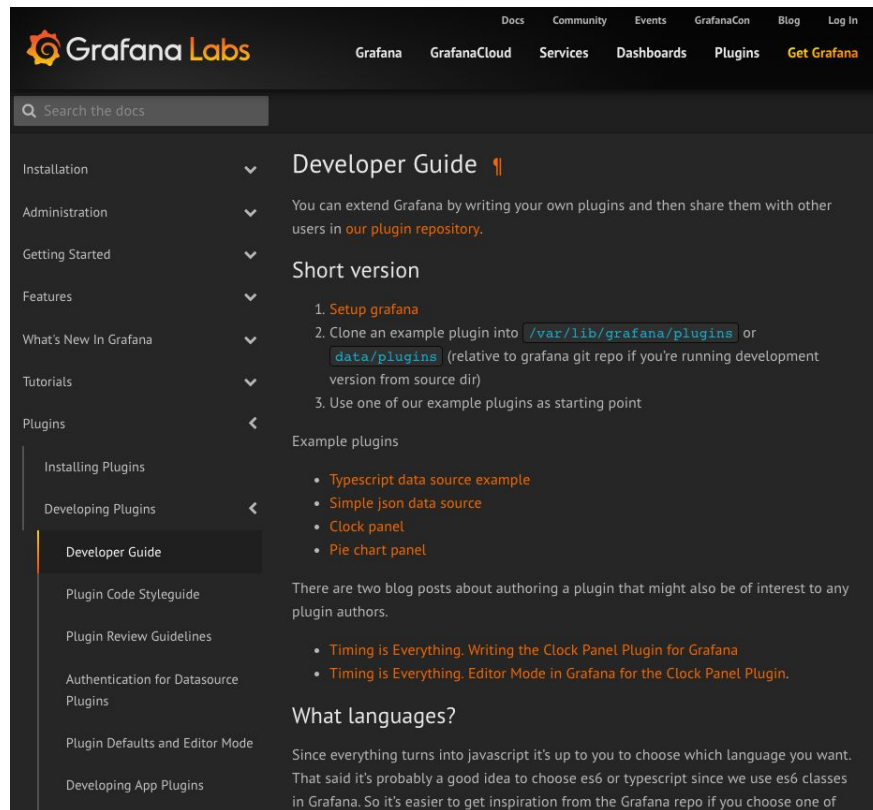
Plug-in di Grafana

Grafana è estendibile sviluppando dei plug-in in linguaggio JavaScript.

Una rete bayesiana applicata ai sensori collegati a Grafana potrebbe fornire allarmi sulla variabili nascoste.

<http://docs.grafana.org/plugins/developing/development/>

<https://github.com/vangj/jsbayes>



The image shows a screenshot of the Grafana Labs website's Developer Guide page. The page is dark-themed and features a navigation menu at the top with links for Docs, Community, Events, GrafanaCon, Blog, and Log In. Below the navigation is a search bar and a sidebar menu with categories like Installation, Administration, Getting Started, Features, What's New In Grafana, Tutorials, and Plugins. The main content area is titled "Developer Guide" and includes a "Short version" section with three numbered steps: 1. Setup grafana, 2. Clone an example plugin into `/var/lib/grafana/plugins` or `data/plugins` (relative to grafana git repo if you're running development version from source dir), and 3. Use one of our example plugins as starting point. Below this, there is a section for "Example plugins" with a list of links: Typescript data source example, Simple json data source, Clock panel, and Pie chart panel. At the bottom, there is a section titled "What languages?" which states that everything turns into javascript and suggests choosing es6 or typescript.

Domande?