



Carbon12 – Predire in Grafana

Verbale Esterno

Informazioni sul documento

Versione	1.0.0
Data di creazione	2020/02/13
Redazione	Francesco Gobbo
Verifica	Nicolò Fassina
Uso	Interno
Destinatari	Carbon12

1 Informazioni sull'incontro

Data	2020/02/13
Luogo	Sede Zucchetti – Via Cittadella 7, Padova
Ora inizio	14.30
Ora fine	15.00
Partecipanti	Giacomo Callegari Manuel De Franceschi Nicolò Fassina Francesco Gobbo Andrea Longo Alessandro Lovo Veronica Pederiva
Partecipanti esterni	Gregorio Piccoli per Zucchetti SPA

2 Ordine del giorno

1. Quali modelli indicare nell'Analisi dei Requisiti
2. Quali funzioni indicare nell'Analisi dei Requisiti
3. Quali sorgenti monitorare su Grafana
4. Ruolo assunto dall'Utente
5. Spiegazione in merito all'analisi svolta su SVM e RL
6. Discussione prototipo per il Proof of Concept
7. Varie ed eventuali

3 Discussione Argomenti

3.1 Quali modelli indicare nell'Analisi dei Requisiti

Si è concordato con il proponente di specificare che i modelli che svilupperemo saranno Support Vector Machine e Regressione Lineare, come modelli principali, mentre inseriremo le Reti Neurali tra i modelli opzionali.

Attualmente il gruppo ha concordato lo sviluppo sia del modello SVM, che del modello RL, mentre le Reti Neurali si pensa di non svilupparle per il progetto.

3.2 Quali funzioni indicare nell'Analisi dei Requisiti

Si è concordato con il proponente che sono più che sufficienti le trasformazioni logaritmica ed esponenziale come funzioni inseribili nell'attività di addestramento.

3.3 Quali sorgenti monitorare su Grafana

In merito alle sorgenti da monitorare, il proponente ci ha esposto una classificazione generale delle sorgenti che loro stessi monitorano. La classificazione si suddivide in 3 aree: Hardware, App Server e Database.

Per l'area Hardware ci ha consigliato di monitorare:

- L'uso della CPU (tempo attesa disco, uso dedicato all'utente);
- RAM in uso;
- I/O sul disco;
- Traffico di Rete;

- Spazio usato del disco.

Per l'area App Server si sono ipotizzate i seguenti elementi da monitorare:

- Numero di pagine al secondo o minuto;
- Tempo medio di risposta della pagina;
- Numero di sessioni attive tramite i cookie (elemento da valutare a seconda della tipologia di pagina);
- Dimensione media della pagina.

In fine per l'area Database:

- Numero di connessioni;
- Numero di query;
- Tempo della query;
- Tempo medio.

Il proponente ci ha poi suggerito di interagire con le dashboard standard offerte da Grafana, così per poi sarà più semplice affrontare una scelta più adeguata ai nostri obiettivi di sviluppo. Quindi le sorgenti saranno stabilite appena il team avrà fatto sufficiente esperienza tramite il PoC e la preparazione della TB, vedendo così più chiaramente e in dettaglio il traffico delle informazioni tra le varie tecnologie.

3.4 Ruolo assunto dall'Utente

Per conferma abbiamo chiesto dei chiarimenti al proponente in merito al ruolo che si deve assumere l'Amministratore per l'avvio dell'addestramento.

Il proponente ci ha spiegato che l'Amministratore con i dati da monitorare, accederà a Grafana e li metterà in monitoraggio e se nota un'attività interessante, inserirà una etichetta ed estrarrà i dati con il formato desiderato. In questo modo i dati sono stati già classificati da Grafana, perciò l'Amministratore può ora inserire i dati nell'applicazione di addestramento per avviare questa attività. Nel caso di addestramento continuo, invece, i dati verranno prelevati direttamente in Grafana.

3.5 Spiegazione in merito all'analisi svolta da SVM e RL

Il proponente ci ha spiegato più nello specifico quali sono i cambiamenti che ci interessano maggiormente per l'addestramento dei modelli di apprendimento.

I dati che vengono analizzati permetteranno ai modelli di tracciare una "linea" dell'andamento dei dati in generale, essa sarà la threshold a cui dobbiamo porre attenzione. In quanto ciò che il modello analizzerà sarà lo scostamento dei dati nei confronti della soglia, ma i dati che ci importano saranno solo quelli che superano tale soglia, poiché il modello apprenderà ricordandosi l'andamento di sviluppo delle situazioni "critiche" definendo così dei pattern di riferimento. I pattern generati serviranno per il confronto, in quanto i modelli attivi osserveranno l'andamento dei dati e il modello potrà prevedere una situazione critica se ci dovessero essere i presupposti per il suo raggiungimento.

3.6 Discussione prototipo per il Proof of Concept

Il team si è concordato col proponente su cosa presentare per il PoC:

- Per quanto riguarda il monitoraggio dell'applicazione si è deciso di sviluppare una versione primitiva che metta già in collegamento tecnologie come InfluxDB e Telegraf per avere un sistema che stia già girando. Per quanto riguarda il plug-in si è deciso di inserire un titolo

“Carbon-12” è un grafico che valuti i dati nell'immediato. Inoltre, per integrare le tecnologie legate al front-end verrà realizzato un mock del pannello di configurazione.

- Mentre per l'applicazione di addestramento, si è ritenuto sufficiente ottenere il caricamento dei dati, la prova di addestramento di uno dei modelli disponibili e il download del predittore.

3.7 Varie ed eventuali

Durante le discussioni in merito alle sorgenti da monitorare e al prototipo da presentare come PoC, il proponente ci ha citato una lista di tecnologie di cui potremmo far uso per lo sviluppo del progetto.

- **InfluxDB**: è un database di serie temporali open-source (TSDB), è un database ottimizzato per ottenere un'alta disponibilità per l'archiviazione e il recupero di dati delle serie temporali. Usato in campi quali monitoraggio delle operazioni, metriche delle applicazioni, dati dei sensori dell'IoT (Internet of Things) e analisi in tempo reale.
- **Prometheus**: applicazione software gratuita usata per il monitoraggio e la segnalazione di eventi. Registra metriche in tempo reale in un database di serie temporali (ciò consente un'alta dimensionalità) creato utilizzando un modello pull http, con query flessibili e avvisi in tempo reale. L'applicazione è concessa con la licenza di Apache 2.
- **MySQL**: è un sistema di gestione di database relazionale open-source (RDBMS).
- **Telegraf**: è un agente di plugin-driver server per la raccolta e l'invio di metriche da database, sistemi e sensori IoT.
- **Collectd**: è un daemon Unix che raccoglie, trasferisce e archivia i dati sulle prestazioni dei computer e di apparecchiature di rete. I dati acquisiti hanno lo scopo di aiutare gli amministratori di sistema a mantenere una visione d'insieme delle risorse disponibili per rilevare i colli di bottiglia esistenti o imminenti.
- **Apache**: Apache HTTP Server, o più comunemente Apache, è il nome di un server web libero.
- **Tomcat**: Tomcat è un Application server open source, ma è anche un web server che implementa le specifiche JavaServer Pages e servlet, fornendo quindi una piattaforma software per l'esecuzione di applicazioni Web sviluppate in linguaggio Java.

Una parte di queste tecnologie ci sono state suggerite anche dal consulente, abbiamo preso nota anche di questi suggerimenti.

Procedendo con la pianificazione della TB decideremo quali a nostro avviso siano le tecnologie che ci aggradano maggiormente e le comunicheremo al proponente.