Report 27/10/2020

Alessandro Franca

Attività svolte queste settimane:

- Analisi e test del codice repo "p7_Aggregator_v2", in modo tale da comprenderne appieno il funzionamento.
- Una volta capito come discriminare Pod singoli, da configurazioni miste, mi sono soffermato sul concetto di flessibilità.

In attesa di delucidazioni, ho deciso di cominciare a realizzare la WebApp tramite il framework *Flask* (di seguito, ciò che ho implementato fino ad ora):

- Model:
 - User: utenti registrati al servizio (User db molto leggero, realizzato con estensione SQLAlchemy di Flask)
 - Configuration: per salvare una configurazione di Pod inserita dall'utente tramite form (Figura 3-4)
- Realizzazione di APIs in modo da offrire l'ottimizzatore come servizio web.
 Endpoint realizzati fino ad ora:
 - index()
 - login()
 - logout()
 - register()
 - reset_password()
 - getUserProfile()
 - setConfiguration()
- Realizzazione di una prima demo dell'interfaccia web:
 - form di Registrazione/Login/ResetPassword (integrazione con servizio di mailing, grazie all'estensione Email di Flask) (Figura 1-2)
 - o form per il setting dei parametri di Configurazione tramite user input (*Figura 3-4*)
 - Utilizzo dell'estensione di WTForms per rendere semplice ed intuitiva la creazione dei form direttamente tramite codice python
 - Utilizzo di Bootrstrap per il rendering

Figure 1-2:

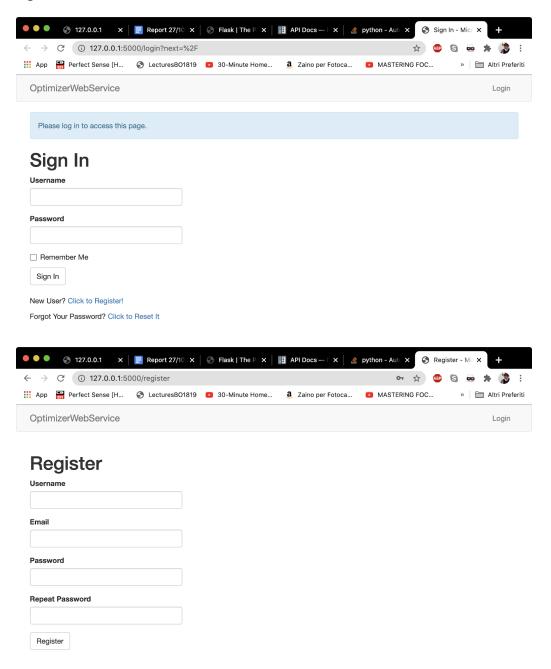
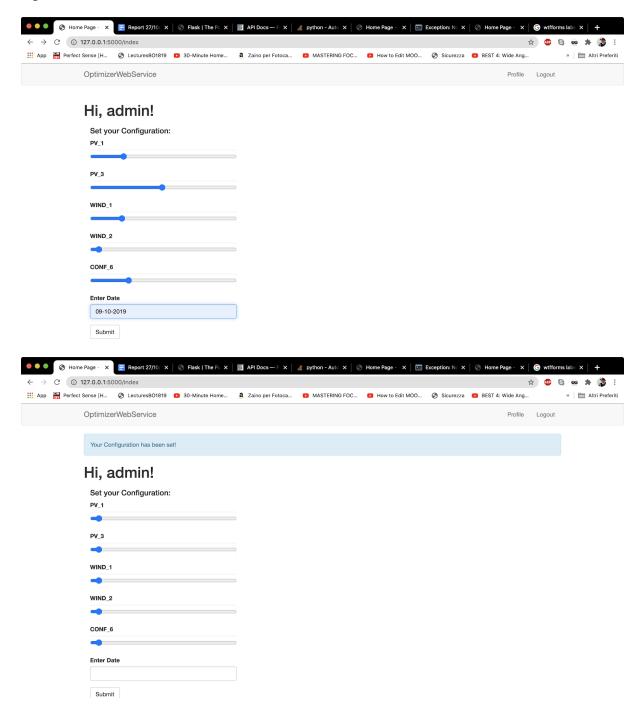


Figure 3-4:



L'intenzione per ora, sarebbe quella di concentrarsi sulla webapp, realizzando altri endpoint. Principalmente:

- Lettura delle baseline da Libra, a partire dalla configurazione settata
- Creazione dei Pods da dare in input all'aggregatore
- Esecuzione delle ottimizzazioni locali dei Pods misti letti da Libra
- Visualizzazione risultati

Ho predisposto una sezione utente, in cui possono essere eventualmente salvate alcune configurazioni effettuate, in modo tale da essere direttamente riutilizzate al bisogno, senza andare a reinserire ogni volta l'input dal form per le configurazioni.

Di seguito la demo:

