
Internals of Application Server

IoT Platform - Scheduler

18th March 2021

TEAM-2 (Group-5)

SUBMITTED BY :

ABHISEK MOHAPATRA (2020201020)
PARTH ARORA (2020201054)
PARAS JOSHI (2020201028)

Table Of contents :

1. Introduction

2. Scope

3. Intended use

4. Assumptions and Dependencies

5. Functional Overview

6. List of Services/Technologies used

7. Interactions with other Modules

8. Block Diagram

Introduction

Scheduling is essentially a decision-making process that enables resource sharing among a number of tasks by determining their execution order on the set of available resources. The process of managing task allocation, to where and to who, is the responsibility of a scheduler. In other words, a *scheduler* is responsible for managing incoming process requests and taking scheduling decisions to determine which process needs to be executed based on process priority, job type etc. The scheduler of a distributed system performs akin to the process scheduler on any operating system.

Scope

This is an application deployment platform which contains various independent services deployed using Microservices architecture such as handling dynamic scheduling requests, maintenance of job queue and database management services.

Intended Use

The deployed application can have different scheduling policies either they can be instant or may be having within a certain time interval. Also there can be some prior priority decided for the execution of the application which needs to be taken care.

So this work is being done with the help of a scheduler having efficient scheduling policies and the best algorithm. Also there can be dynamic scheduling depending upon the needs of the application and various preemptive features of the server.

Assumptions and Dependencies

Initially the client must assign the priority either it may be high or low, it should not be void and must be chosen from the given options. This will decide the scheduling priority of any process and the application will run on that basis only. The scheduler will perform its due responsibilities only after the configuration file is successfully validated by the Application Manager module. The user can add configuration details via UI or upload file option. The supported file types for upload option are json and xml files.

Functional Overview

The application works as :

- 1.) Scheduler module accepts the information from the application manager which validates the information of the user and now the scheduler will handle its scheduling information given by the user.
- 2.) After reading the scheduling information, the scheduler will now apply the assigned algorithm and send the application to the queue or can immediately schedule depending upon the priority and information provided.
- 3.) Now the scheduler will update the metadata information having user id,application id, execution time, various time(start time, scheduling time, end time) etc. and send it to the deployment module via communication module for further deployment and processing to take place.

List of Services/Technologies used

- 1.) *Job Queue* Maintenance where processes are added on First In,First Out Basis.
- 2.) Handling Interaction with different people with the help of *FLASK*, such as message passing (eg. Metadata) and intermediate interactions of receiving and sending information.
- 3.) Interaction with *MongoDB* for retrieving scheduling algorithms in real time.
- 4.) UI / File upload service is provided to the user to determine the nature of the job/process which is then processed by the scheduler via Configuration file. Technologies used to develop UI include *HTML,CSS,Bootstrap,Javascript*.

Interactions with other Modules

- 1.) The scheduler component interacts with the Application Manager module to retrieve the configuration file containing important information regarding the processes (eg: process priority,process type etc) and then determines the scheduling algorithm to be used to schedule the processes.
 - 2.) The scheduler will update the metadata information having user id,application id, execution time, various time(start time, scheduling time, end time) etc. and send it to the deployment module via communication module for further deployment and processing to take place.
-

Block Diagram

