OMP 2020 Mentorship Program Project Plan

Mentor: Alex Kim

Mentee: Salisu Ali

Project: **Zowe parsing engine for SMF or RMF PP Reports**

Project Description:

Zowe is a great systems operations tool. One of the systems programmers or performance analyzer's job is to decode SMF/RMF reports to check system's health. Having a generic parser for SMF datasets and/or RMF (or CMF) reports for Zowe would open various opportunities to create/re-use many open-source monitoring tools out there.

Problem Definition:

There is no intuitive way to analyze SMF/RMF records by default in z/OS. Some users might choose to analyze these documents in a spreadsheet, and identify anomalies by hand, but this as tedious as it is inefficient. We wish to create a parsing engine that will automate the analysis of SMF/RMF reports using modern technologies that will make systems programmers or performance analyzer's job easier and efficient.

Date: 19-05-2020

Project Plan

TASK 1: FETCH SMF/RMF DATA IN REALTIME

➢ Using Distributed Data Service (DDS) HTTP API

DDS HTTP API provide access to RMF data from RMF Monitors and XML Postprocessor Reports. To get RMF data in real-time, RMF Monitor III resources/data can be accessed using the DDS HTTP API. An application program can request selected performance metrics or complete reports from the DDS.

FETCHING REALTIME AND POSTPROCESSOR STATIC DATA

This Depends on the Request Type.

i. Realtime

The rmfm3.xml filename is used to specify Monitor III report request. This provide Realtime RMF reports.

ii. RMF Postprocessor Static data

The rmfpp.xml filename is used to specify a postprocessor request. Duration and report type are also included in the Request.

DDS HTTP API

This API takes different parameters for requesting specific as well as general RMF data from a single-system or sysplex RMF reports. The API returns an XML file.

SOME DDS HTTP API PARAMETERS

i. ***ddshost*** : is the IP address or the symbolic name of the DDS server.

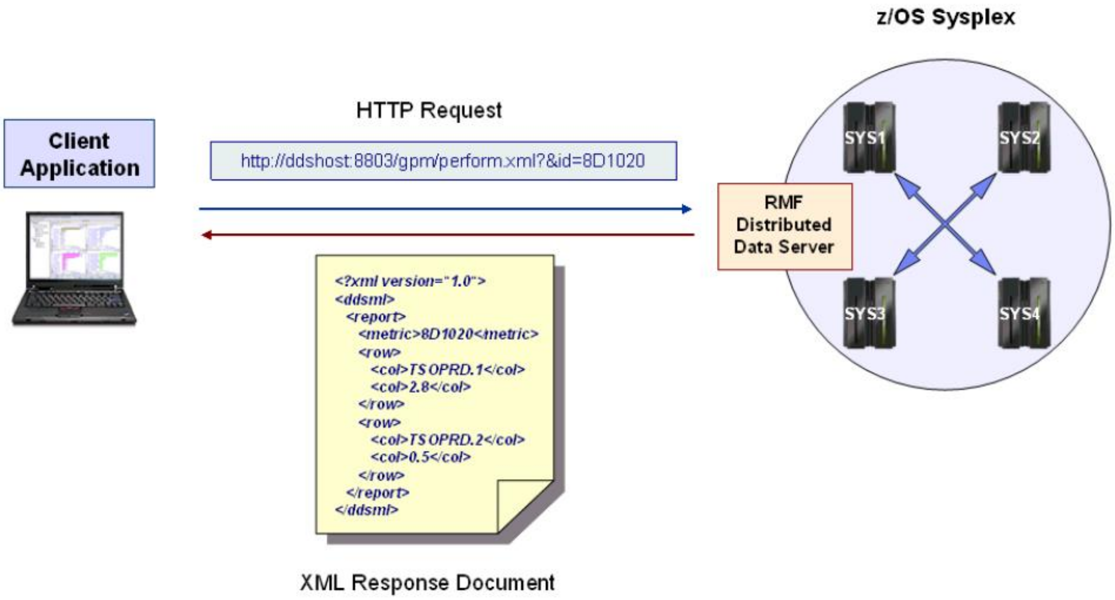
Example:

ddshost or 192.***.**.*

ii. ***ddsport*** : is the port number of the DDS server (GPMSERVE or GPM4CIM).

Example:

8803



```
<?xml version="1.0">
<ddsml>
  <report>
    <metric>8D1020<metric>
    <row>
      <col>TSOPRD.1</col>
      <col>2.8</col>
    </row>
    <row>
      <col>TSOPRD.2</col>
      <col>0.5</col>
    </row>
  </report>
</ddsml>
```

Figure 8: Example: How to use the DDS HTTP API in a z/OS environment

iii. ***The smfdata parameter***: This parameter contains a list of names of SMF data sets or logstreams which are used as input for the generation of Postprocessor reports. The names must be fully qualified and valid z/OS data set names.

Examples:

smfdata =RMF.SMFDATA.SYSA,RMF.SMFDATA.SYSB,RMF.SMFDATA.SYSC

smfdata =IFASMF.PERF.SYSDPLEX

TASK 2: CREATE NODEJS APP

➢ APP

Node js app with MongoDB backend expose to the browser using express. A mongoose ODM for easy access to MongoDB.

➢ Components

i. API

- ***DDS HTTP API***

This API will be constructed to provide access from the node js app to RMF data through DDS.

- ***Database API***

  This API will provide access to MongoDB from the node js app. This will help in writing and retrieving data from the database without tightly coupling the database interactions with the application logic.

ii. **Files**

- ***Config file***

  This file will contain data needed parameters by the DDS HTTP API to retrieve different resources/reports from the RMF data. Example include ***ddshost, ddsport, files, date, duration*** etc.

- ***Database file***

  This file will contain database logic like opening and closing connection to the database.

iii. **XML PARSER**

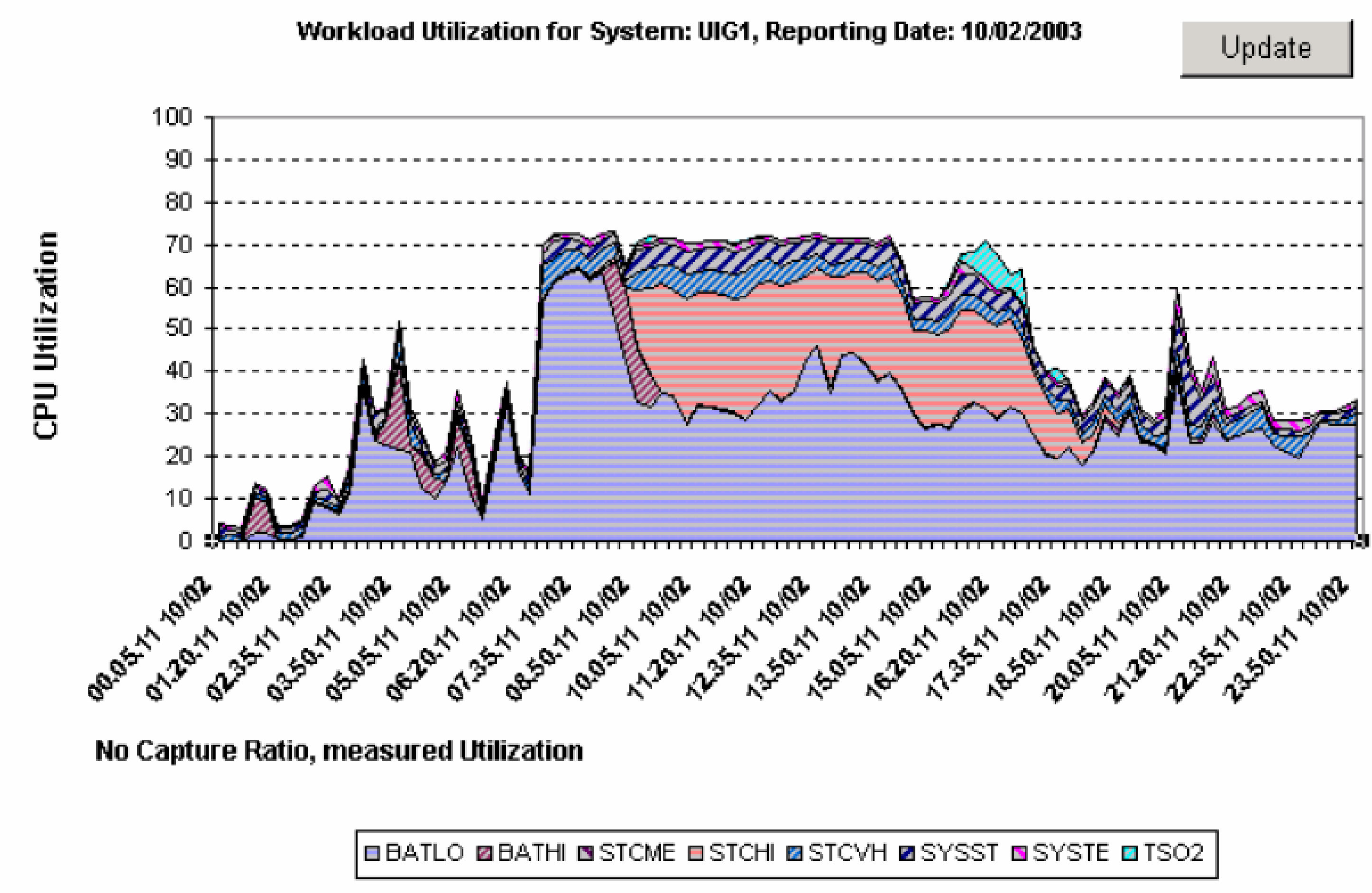
This parser will be used to extract information from the XML file returned by the DDS HTTP API request.

iv. **UI**

The APP will contain user interface for collecting config and other relevant data from the user. It will also allow the user to select options and visualize processed report. The UI will be built using Angular.

TASK 3: CONNECT WITH GRAFANA ON Z/OS OR ZCX OR CLOUD

Grafana is an Open Source analytics and monitoring Solution for Databases. It will be used to generate representative charts for all performance-related areas with the App database serving as data source for plotting the graphs. This feature will be available on the Zowe Application framework.



TASK 4: DEPLOY NODE APP ON Z/OS

Node JS App will be deployed to Z/OS as a plugin on top of the Zowe Application framework.

TASK 5: CREATE CONNECTION TO ZOWE AS ZOWE APP

The App will be connected to ZOWE as a Zowe App which is accessible through the Zowe Desktop.

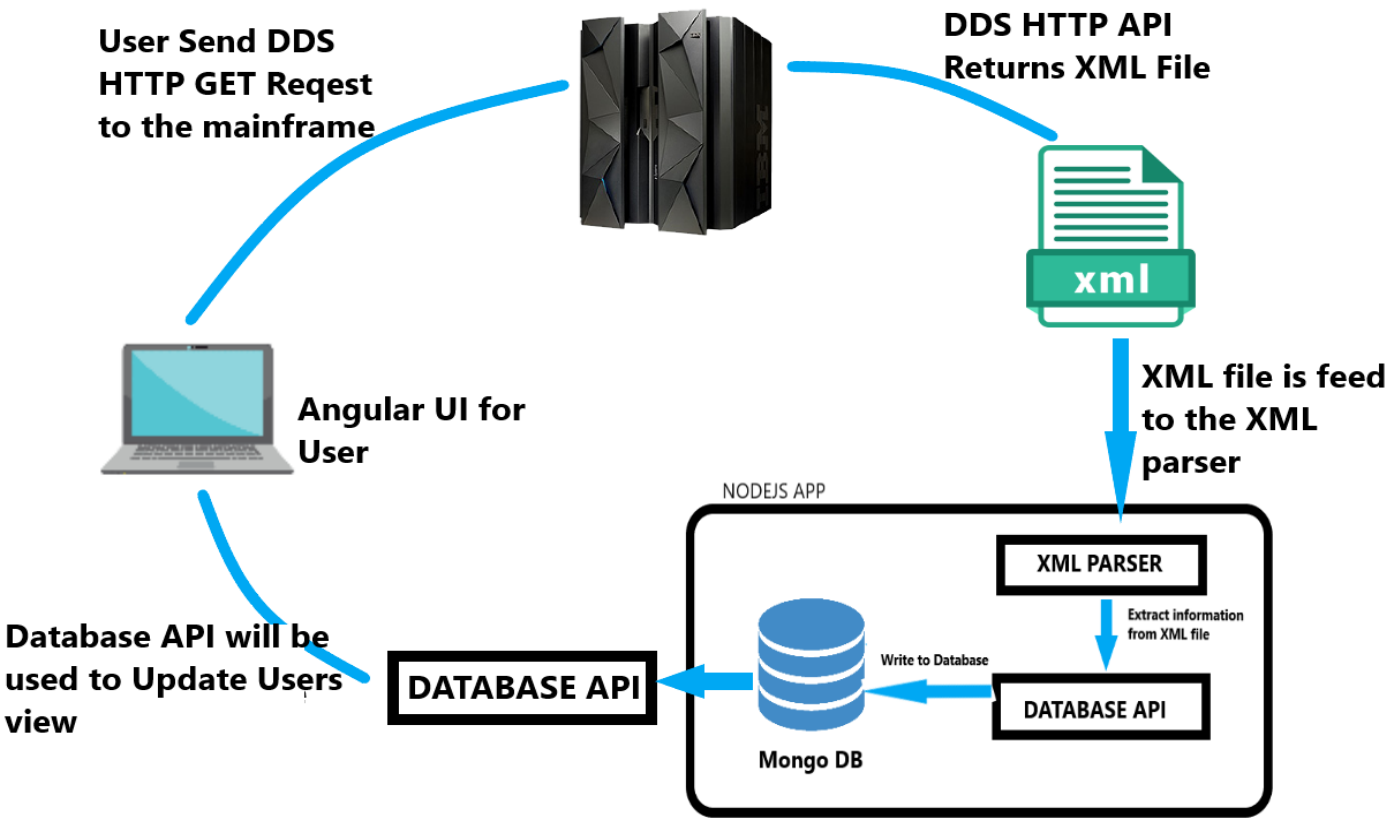
TASK 6: LIST APP IN API CATALOGUE

An API version of the App will be added to the Zowe API catalogue which will allow users to access RMF performance metrics in form of JSON response.

TASK 7: SMF PROCESSING

To Access and parse RMF performance metrics from SMF dataset residing within the Mainframe

APPLICATION ARCHITECTURE



User Send DDS HTTP GET Reqest to the mainframe
DDS HTTP API Returns XML File
xml
XML file is feed to the XML parser
Angular UI for User
NODEJS APP
XML PARSER
Extract information from XML file
DATABASE API
Write to Database
Mongo DB
DATABASE API
Database API will be used to Update Users view

DELIVERABLES

- ➢ Zowe Application Plugin

Features:

i. Interactive Angular UI
ii. Chart display through Grafana
iii. Accessible Through Zowe Desktop

- Zowe API

Features:

i. Returns RMF performance Metrics as JSON Response
ii. Access through the API Catalogue

TIMELINE

i. Week 1 (18-05-2020 to 24-05-2020)
   - Set up Nodejs App
   - Set up Request to DDS Server
   - Parse first batch of RMF Activity Reports

ii. Week 2 (25-05-2020 to 31-05-2020)
   - Parse Second Batch of Activity Reports

iii. Week 3 (1-06-2020 to 7-06-2020)
   - Create API for Writing data to NoSQL DB
   - Create API for Retrieving data From NoSQL DB

iv. Week 4 (8-06-2020 to 14-06-2020)
   - - Build Angular Frontend
   - - Add Grafana Charts

- - Create API version Of APP to be uploaded to API catalog

v. Week 5 (15-06-2020 to 21-06-2020)

- Deploy Node App to z/os
- Add Zowe Connection as Zowe App
- Add App to API Catalogue

vi. Week 6 (22-06-2020 to 28-06-2020)

- Parse RMF Records from SMF Data

vii. Week 7 (29-06-2020 to 5-07-2020)

- Parse RMF Records from SMF Data

viii. Week 8 (6-07-2020 to 12-07-2020)

- Parse RMF Records from SMF Data

ix. Week 9 (13-07-2020 to 19-07-2020)

- Parse RMF Activity Report from Text file

x. Week 10 (20-07-2020 to 26-07-2020)

- Parse RMF Activity Report from Text file

xi. Week 11 (27-07-2020 to 2-08-2020)

- Parse RMF Activity Report from Text file

xii. Week 12 (3-08-2020 to 9-08-2020)

- Documentation

xiii. Week 13 (10-08-2020 to 16-08-2020)

- Documentation

xiv. Week 14 (17-08-2020 to 23-08-2020)

- Documentation

xv. Week 15 (24-08-2020 to 30-08-2020)

- Documentation

xvi. Week 16 (17-08-2020 to 23-08-2020)

- Documentation

xvii. Remaining Period (24-8-202 to 30-9-2020)

- Uncompleted task(if any)