Analysis and Up-gradation of Batch Fusion Development in RAF

Submitted By

Saloni Shah

18MCEN13

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2020

Analysis and Up-gradation of Batch Fusion Development in RAF

Major Project

Submitted in fulfillment of the requirements

for the degree of

Master of Technology in Computer Science and Engineering (Networking Technologies)

Submitted By

Saloni Shah

18MCEN13

Guided By

Prof. Tejal Upadhyay

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2020

Certificate

This is to certify that the major project entitled **"Analysis and Up-gradation of Batch Fusion Development in RAF"** submitted by **Saloni Shah, 18MCEN13**, towards the partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Engineering (Networking Technologies) of Nirma University, Ahmedabad, is the record of work carried out by her under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project part-I, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Prof. Tejal Upadhyay
Guide & Assistant Professor,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. Gaurang Raval
Associate Professor,
Coordinator M.Tech - CSE (NT)
Institute of Technology,
Nirma University, Ahmedabad.

Dr. Madhuri Bhavsar
Professor and Head,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. R. N. Patel
I/C Director,
Institute of Technology,
Nirma University, Ahmedabad.

Statement of Originality

I, **Saloni Shah**, **18MCEN13**, give undertaking that the Major Project entitled "**Analysis and Up-gradation of Batch Fusion Development in RAF**" submitted by me, towards the fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering (Networking Technologies)** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

_____

Signature of Student

Date:

Place:

Endorsed by

Prof. Tejal Upadhyay

(Signature of Guide)

Acknowledgements

It gives me immense pleasure in expressing thanks and profound gratitude to **Vishwanathraddy Patil**, Technical Lead, Oracle RGBU POM-Dev, Bangalore for his valuable guidance and continual encouragement throughout this work. The appreciation and continual support he has imparted has been a great motivation to me in reaching a higher goal. His guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

It gives me immense pleasure in expressing thanks and profound gratitude to **Prof. Tejal Upadhyay**, Guide, Assistant professor, Institute of technology, Nirma University, Ahmedabad for her valuable guidance and continual encouragement throughout this work. The appreciation and continual support she has imparted has been a great motivation to me in reaching a higher goal. Her guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

It gives me an immense pleasure to thank **Dr. Madhuri Bhavsar**, Hon'ble Head of computer science and engineering Department, Institute of Technology, Nirma University, Ahmedabad for her kind support and providing basic infrastructure and healthy research environment.

A special thank you is expressed wholeheartedly to **Dr. R. N. Patel**, Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for the unmentionable motivation he has extended throughout course of this work.

At last I must express my sincere heartfelt gratitude to all the employees of Oracle RGBU, POM-Dev who helped me directly or indirectly during this course of work.

Saloni Shah

18MCEN13

Abstract

There has been a massive growth in the retail software market worldwide. This is mainly due to the expansion of the retail software in the emerging economy. Also the enterprise retailers are re-architecting their systems to adapt themselves to the changing technological requirement. With the fusion applications migration to SaaS, the batch execution requires a scheduler to automate the batch running. It also requires auto-monitoring capabilities to ease out the batch execution operations via AI techniques. Monitoring module utilizes the scheduling capabilities of Batch scheduler to drive the batch execution cycles within and also between retail applications.

The key engines to automate the batch running are POM and JOS, where one can run batch jobs and other can monitor and accordingly generate notifications for updating the status for the batches to the customers. In the process of automating batch running, three modules namely Job admin, Process flow and Batch scheduler are used. Each of the three modules are provided with artifacts as seed input which is generated by AG. AG is a semi-automated tool which takes XSLM sheets as an input and generates artifact to be fed to each of the three modules respectively.

Since all the applications are deployed independently in distributed systems, there arise a need to use ReST calls for transferring data to establish inter-application communication. The ReSTful implementation of Retail Platform Services is done by the creation of a separate service layer followed by the data access object layer that has a direct interaction with the database. As ReST calls are expensive, some optimizations are required. From the performance perspective, the Retail Platform Services are deployed using the weblogic server.

Abbreviations

ADF	Application Development Framework
AG	Artifact Generator
AI	Artificial Intelligence
AM	Application Module
ARAF	ADF-based Retail Application Framework
CSS	Cascading Style Sheets
CNN	Convolutional Neural Network
EO	Entity Object
HAN-ATT	Hierarchical Network Attention Model
HAN-AVE	Hierarchical Network Averaging Model
IDE	Integrated Development Environment
JAX-RS	Java API for RESTful Web Services
JET	Javascript Extension Toolkit
JRAF	JET-based Retail Application Framework
JSP	JavaServer Pages
JOS	Job Orchestration System
LSTM	Long Short-Term memory
MVC	Model View Controller
POM	Process Orchestration Monitoring
PSRAF	Platform Services-based Retail Application Framework
RAF	Retail Application Framework
ReST	Representational state transfer
RNN	Recurrent Neural Network
SaaS	Software as a Service
SSH	Secure Shell
SVM	Support Vector Machine
VO	View Object
WAR	Web application ARchive
XSLM	Excel Macro - Enabled Workbook (Microsoft Corporation)

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 What is Oracle Fusion Applications?

Oracle Fusion Application is a collection of 100% open standards-based business applications which are made for the latest technology and incorporating the best practices gathered from Oracle's customers. It provides a new standard for the way businesses innovate, work and adopt technology. It ca be briefly described as:

- Built on an open standards-based platform
- Based on best practices business processes
- Deployed through a selection of options
- Built with security as a priority

1.2 Introduction to Retail Fusion Platform

Retail Fusion Platform provides a collection of reusable artifacts that encompasses the Java libraries, page templates, and other assets that are importable which indirectly help the teams in constructing retail fusion applications that satisfy standard benchmarks. The Retail Fusion Platform, likewise called the RAF, enhances the capabilities of Oracle ADF to merge libraries that will provide a positive edge to the teams to build up a Fusion web application, task streams, and other ADF segments with coherent style and behavior.

The features of RAF comprise of the following:

- Login and logout functionality
- Page templates that provide a consistent shell, for the applications to display their contents.
- Navigation model for commencing functional task flows onto the UI Shell
- Manage the language preferences, i.e the language of the application can be changed using the settings
- UI Components for consistently providing tables, trees, panels, and tree tables.
- RAF contains business extension classes that contains the functionality to support multiple languages in data
- Utility classes for accessing the databases, working with ADF bindings and JSF managed beans

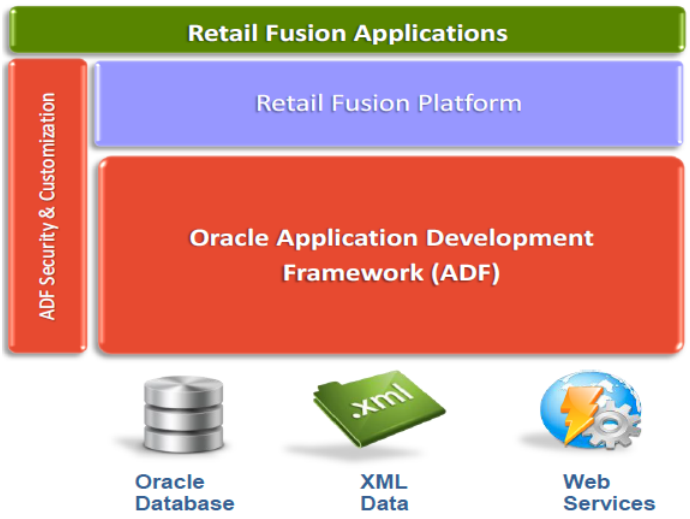

Figure 1.1: Retail Fusion Platform Architecture

Retail Fusion Platform serves as a foundation for all the next generation development of fusion applications within the retail business units of the enterprise. Retail Fusion Platform is delivered as a set of runtime libraries that will be incorporated in the deployment of the Retail Fusion applications. The Retail Fusion Platform, also called as Retail Application Framework (RAF), enhances the capabilities of Oracle ADF to merge libraries that will provide a positive edge to the teams to build up a Fusion web application, task streams, and other ADF segments with coherent style and behavior. Fusion applications are very generic web applications holding the RAF for providing retail specific functionality. RAF in case of Fusion Application can be of three types:

1. ARAF

   It is ADF-based framework with Java implementation. The entire system is built using EOs, VOs and AM. EO is the source code representation of a database table where as VO is read-only database query or updatable object that represent an EO. AM represents paricular application tasks[5].

2. JRAF

   JET-based framework, which provides a foundation to accelerate the development of client-side applications using Oracle JET. It includes JavaScript libraries which will enable the teams to develop workflows and UI components with some established standards.

3. PSRAF

   Platform Services-based famework is implemented using JAX-RS API. It provides services for EOs and VOs in case of ARAF and on-click UI implementation in case of JRAF.

1.3 Objectives

The main objective of the project is to enhance the existing system for POM and JOS to reduce the overall response time in order to migrate POM to JET.

- To develop common components by extending ADF to impose coding standards and patterns across the retail enterprise firms and to assist accelerate application development.
- Build AG to optimize the effort in creation of input fed to the existing system.
- As a part of POM-migration from ARAF to JRAF, implement ReST endpoints for various screens of the application. Also, ensure that the development of fusion applications based on JRAF is independent of ADF Business Components.
- Assist in replacing the JOS-Batch Scheduler with the POM-Batch Scheduler.
- Enhance Batch Scheduler by providing the functionality to Auto-monitor the batch-runs in current schedule.

1.4 Area of Computer Science

- Software Development

  Developing a software which will automate batch processing as well as monitor the status of each batch. The development of the software involves oracle tech stack. For front-end, ADF framework has been used for now which will be later migrated to JET framework. Back-end support is maintained using Oracle database which comprises the details regarding the batch runs like list of process, job within the process with their dependency, type of recurring cycles etc. For querying the oracle database to trigger the jobs, PL/SQL has been used. Inter-communication between two existing systems is handled via JAX-RS.

- ADF

  It is an end-to-end application framework that is built on top of Java-EE standards. ADF is an open-source technologies that implements enterprise applications in a simplified manner [5]. It is highly selective choice for enterprise developers to create applications that allows user to search, display, create, modify, and validate data using web, mobile, and desktop interfaces. At present, ADF is used on front-end for POM.

- JET

  JET is a set of native retail firm and open source JavaScript libraries that is used to develop client-side web and mobile applications. This is completely based on web technologies such as HTML5 and CSS. JET is aimed to create dynamic web pages, create an amalgamated mobile application similar to that of Android or iOS, and also create an end-to-end web application using the latest web technology stack. The existing ADF application follows MVC model. As a part of migration activities, view will be completely replaced by the JET user interface.

- SOA

  It is an architectural pattern that provide services to other components using a communications protocol over a network. It's principles are independent of any product, vendor or technology. This makes it easier for software components over various networks to work with each other.

- ReST

  It is an API architecture style. It specify the format of data representation which is convenient for user or client. REST APIs are based on uniform resource identifier and the HTTP protocol [1]. Data exchanges can be made using either in JSON or XML format [4]. In the fusion system, as POM and JOS are two distinct components they communicate using ReST calls. Moreover, there can be diversity in requirement specification for different customers and products across Oracle. So services must be provided according to the requirement. This segregation of the services is handled using ReST end-points. Each ReST end-points is mapped with an individual service. So as per the customer's requirement, one or more ReST end-points can be assigned. From implementation point of view, Jersery framework of JAX-RS is used to facilitate this requirements.

- AI

  It is the simulation of human intelligence processes by machines, especially computer systems. The key steps followed by this process involves:

  - Learning

    The collection of information and rules for the information.

  - Reasoning

    Utilization of the specified rules to derive approximate or definite conclusions.

  - Self-Correction

    Learning based on experience. [7]

  The smart auto-monitoring system for scheduling the batches in fusion application will follow all the steps.

1.5 Tools used

- Apache Maven

  Maven is a powerful project management tool based on project object model that is used to build the projects, handle its dependency and provide proper documentation. Most basic command to build a cluster is *'mvn -e install'* which will print the entire stack trace.

- Apache Tomcat

  It is a Servlet and JSP Server serving java based technologies. You can deploy the WAR file of your java project by placing it in the deploy directory of the admin console. It is basically an HTTP Server that is serving HTTP.

- Fusion Middleware

  It is a collection of standards-based software products from Java-EE and other developer related tools, for integration of services, identity management, business intelligence, and collaboration.

- Jenkins

  Jenkins being an open source automation server, enables the developer to build, test and deploy reliably.

- Microsoft Excel

  It is a spreadsheet software used to generate the input for the artifact generator with macros validations.

- MobaXTerm

  MobaXterm is an enhanced terminal for Windows. It facilitates the windows system with all the necessary Unix commands to Windows desktop, This is done by importing a single portable .exe file in the windows desktop.

- Oracle JDeveloper

  JDeveloper is a open IDE provided by Oracle Corporation. It provides features for development in object oriented language like Java along with XML, SQL and PL/SQL for backend support and HTML, JavaScript, BPEL and PHP for web development. It contains the full lifecycle development starting from design via coding, debugging, optimization and profiling to deploying.

- Postman

  It is a Google Chrome plug-in for API development testing. The main features of Postman includes managing API collections, environments, tests and sharing. It is used for testing the results of ReST end-points on deploying the WAR file to Apache Tomcat or Weblogic Admin Console.

- Oracle SQL Developer

  It is an IDE for working with SQL in Oracle databases. It uses the Java Development Kit.

- TortoiseSVN

  TortoiseSVN is a Subversion client used to handle directories along with files. All the files to be shared are placed inside a repository which is in center to all the users from the access perspective. The work of a repository which is used for the storage of the file is to monitor and log easch and evry modification or updates ever been made to the files. This helps the user to maintain older versions of the existing files and analyse, monitor and examine the previous records of how and when and what changes were made to your data along with details of who changed it.

- Weblogic Server

  It is an application server which provides a platform for developing and deploying multi-tier distributed enterprise applications.

- WinSCP

  WinSCP is a freeware windows client for the secure copy protocol that provides a way to transfer files across the network using the SSH encrypted protocol. It is basically a replacement for FTP programs lying on windows users in scenarios where SSH must be connected to the server.

Chapter 2

POM and JOS Integration

2.1 Introduction to JOS

JOS is a generic tool to schedule tasks, manage and orchestrate dependencies between tasks and run non-interactive long running Jobs.

As show in Figure 2.1, the three main components of JOS are mentioned below :

1. JOS Scheduler

   Generic GUI tool to define and manage time based scheduling work

2. JOS Process Flow

   Defines workflow by connection and orchestrating multiple tasks (Processes). Provides an engine to execute the workflows.

3. JOS Job Admin

   Based on standard JavaBatch (JSR 352) technology. GUI to manage and monitor Jobs. [6]

2.2 Overview of POM

With RGBU applications migrating to the SaaS cloud, batch job execution requires an in house scheduler to automate running and monitoring nightly batch executions. POM works with JOS components. POM utilizes the scheduling capabilities of JOS components, to drive batch execution cycles within and also between retail applications. It is built on the top of ARAF. It also supports multiple batch schedules. Originally POM

was only built for RMS application.

2.2.1 Features of POM

- POM ADF Application

  ADF GUI provided to monitor and administer Batch Schedules. Provides a whole lot of other features and is central to POM.

- POM Process Services

  REST Interface for POM. All the JOS communication is with Process Services

- Data Privacy Services

  ARAF component for GDPR adherence.

- Alerts Manager

  ARAF component for Alerting via different channels (Email, mobile etc)

- ORAAC ARAF component used for administering policies.

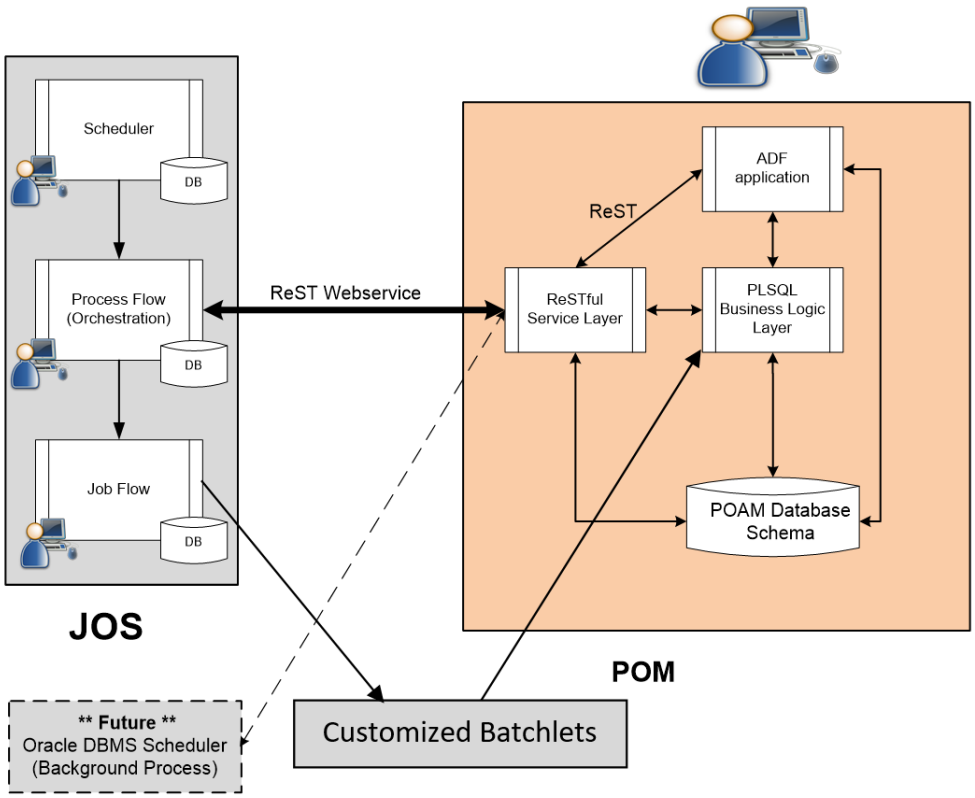

Figure 2.1: Overview of POM and JOS

2.3 POM and JOS Architecture Change

2.3.1 Old approach

- JOS and POM interacts via ReST webservice.
- Separate database for JOS POM.
- During install time, the batch schedule information maintained in spreadsheet is replicated into JOS and POM. Status is maintained in JOS and POM.
- Batch execution happens in JOS but POM controls the batch run.
- POM provides AMS with batch monitoring and management capability.

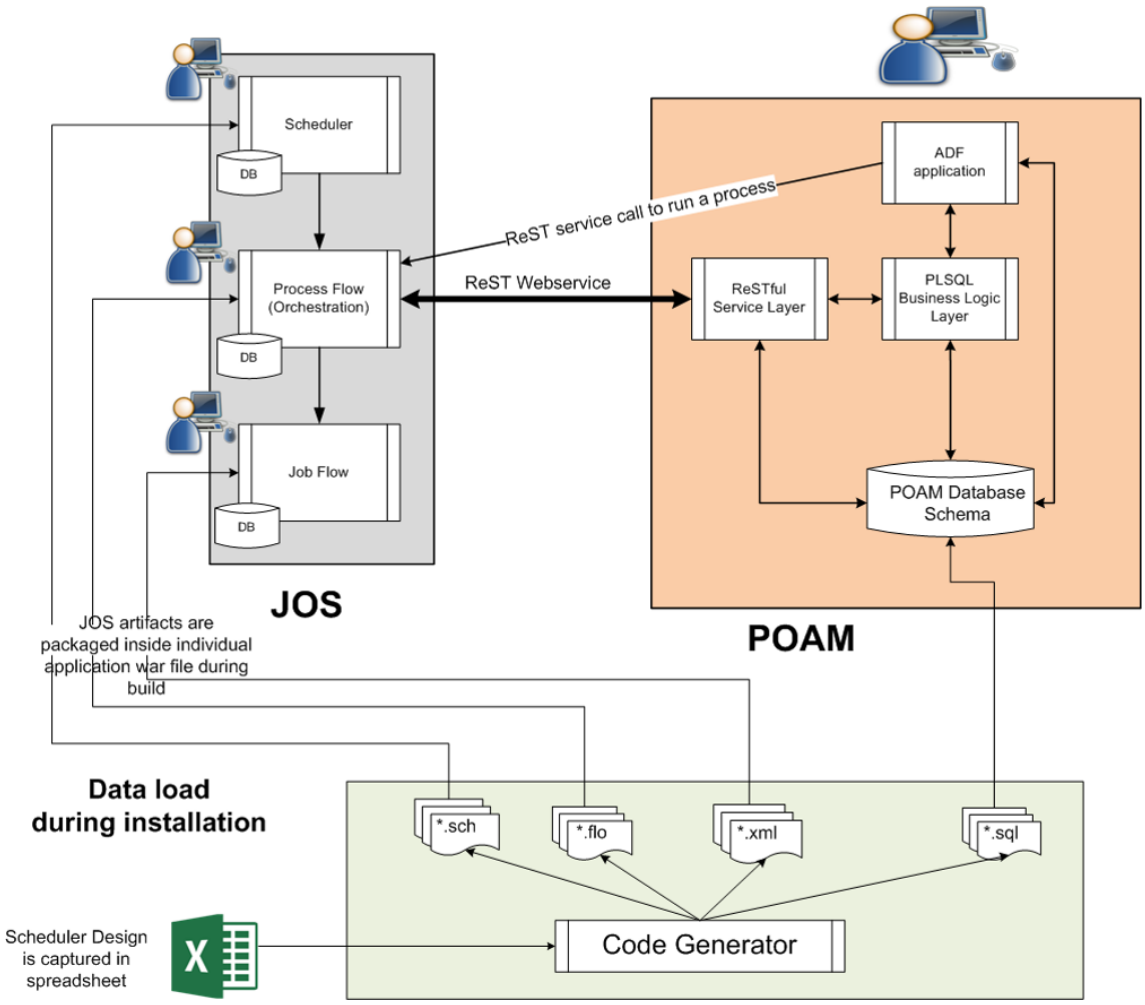

Figure 2.2: Integration of POM and JOS

2.3.2 New approach with POM Execution Engine

- Execution Engine of JOS is replaced with New light weight execution engine which directly talks to POM Database.
- No rest service communicationbetween POM and JOS for status sync.
- Status is maintained in POM only
- Dependency changes can be done in POM without a patch.
- No change in existing business logic
- Dynamic parameter change limitation while invoking the service from external system is addressed.

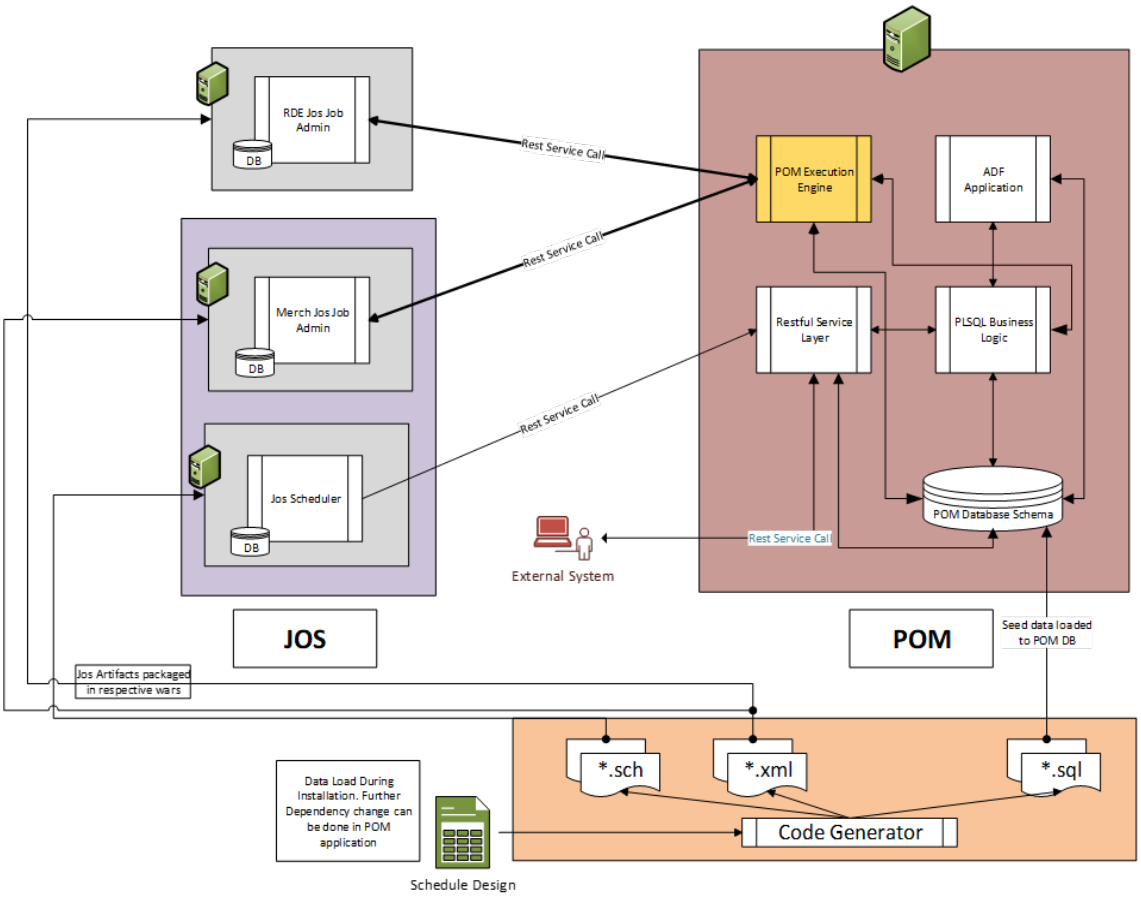

Figure 2.3: New integration of POM and JOS

2.3.3 Advantages of new approach

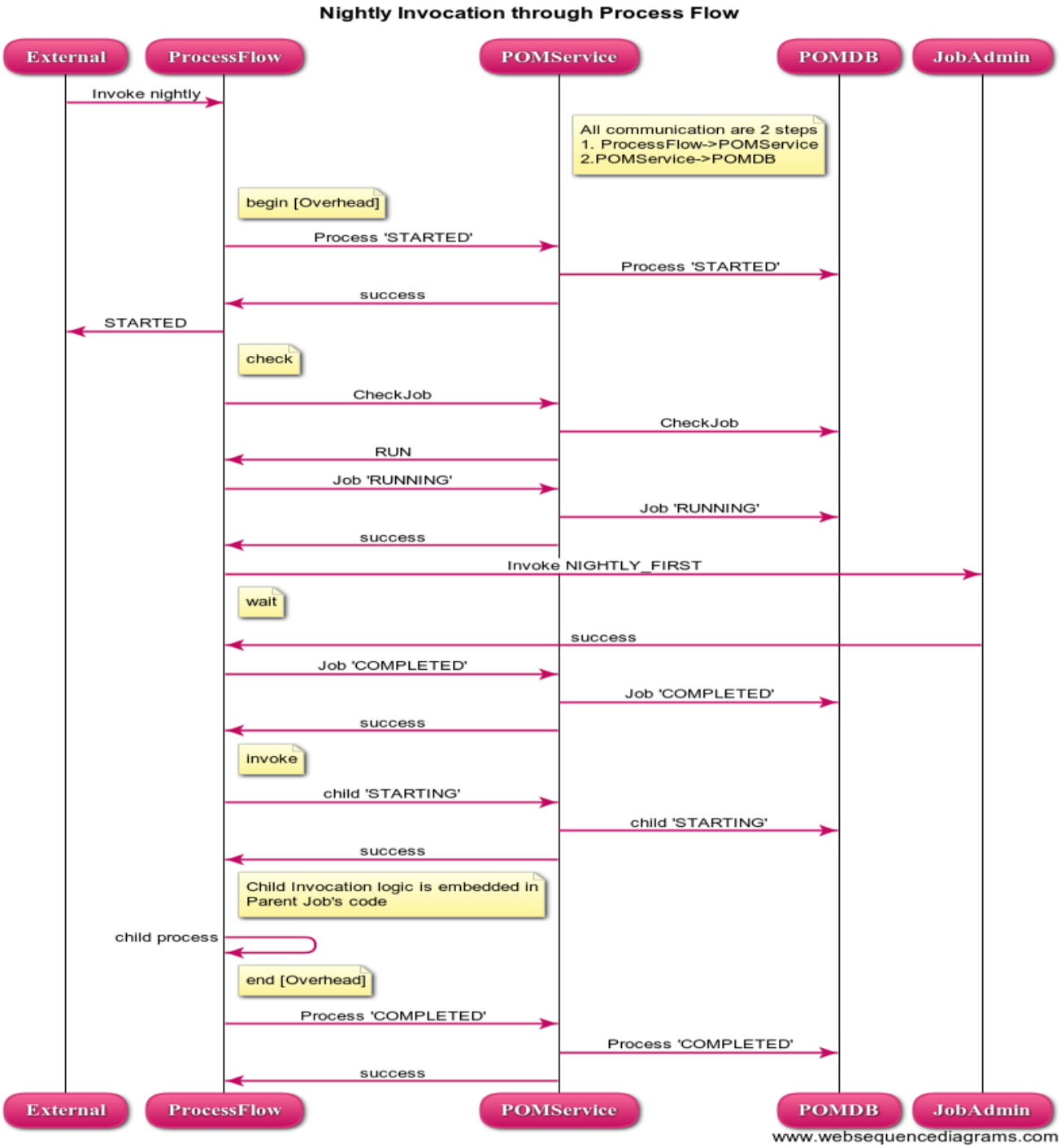

Figure 2.4: Nightly invocation using old approach

- Single source of truth. Batch Execution Engine directly talks to POM DB. Status syncing is not required.
- Minimal communication involved as shown in Figure 2.4 and Figure 2.5.
- Better way of invoking a cycle. It only needs to know the cycle number and name

and scheduler day. It is based on eternal single threaded controller which keeps polling the runnable jobs in POM DB. All runnable jobs are attempted to be executed on job admin parallel in different managed threads. Once a job is executed its status is written on POM DB.

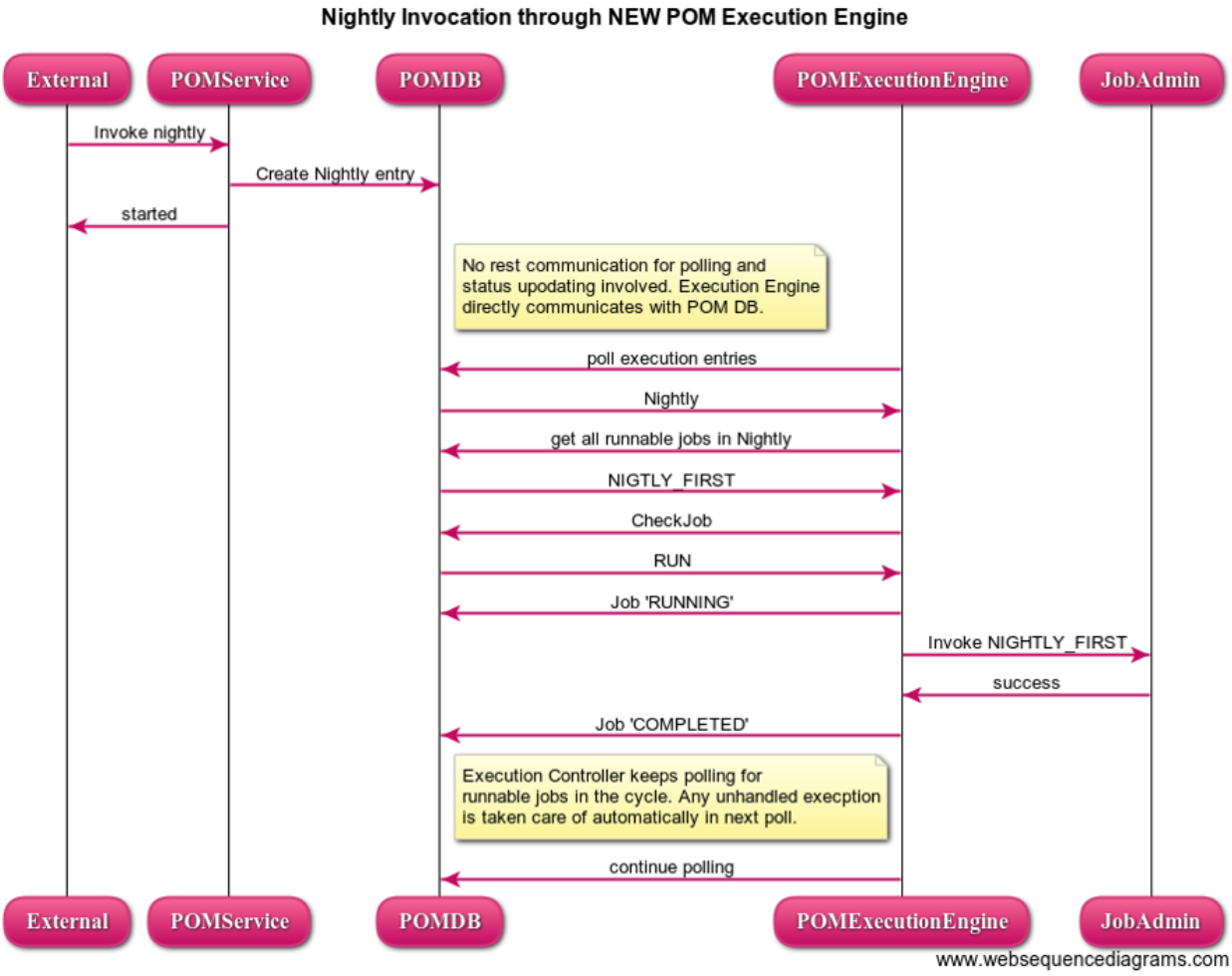

Figure 2.5: Nightly invocation using new approach

Chapter 3

Artifact Generator

3.1 Need for AG

The POM requires *.sql files as an input, JOS requires *.xml and *.sch for Job Admin and JOS Scheduler respectively. The same input data needs to be converted into multiple formats. All the format inherits XML specifications. So, if the user wants to run a batch with 50 jobs then the user have to generate 50 XML files to be fed to the POM and JOS. With the advent of AG, the task is reduced to just entering the jobs in a single spreadsheet and giving it as an input to AG. The AG takes spreadsheet with XLSM as an input and generates the required input formats for POM and JOS.

3.2 Artifact Generation Process

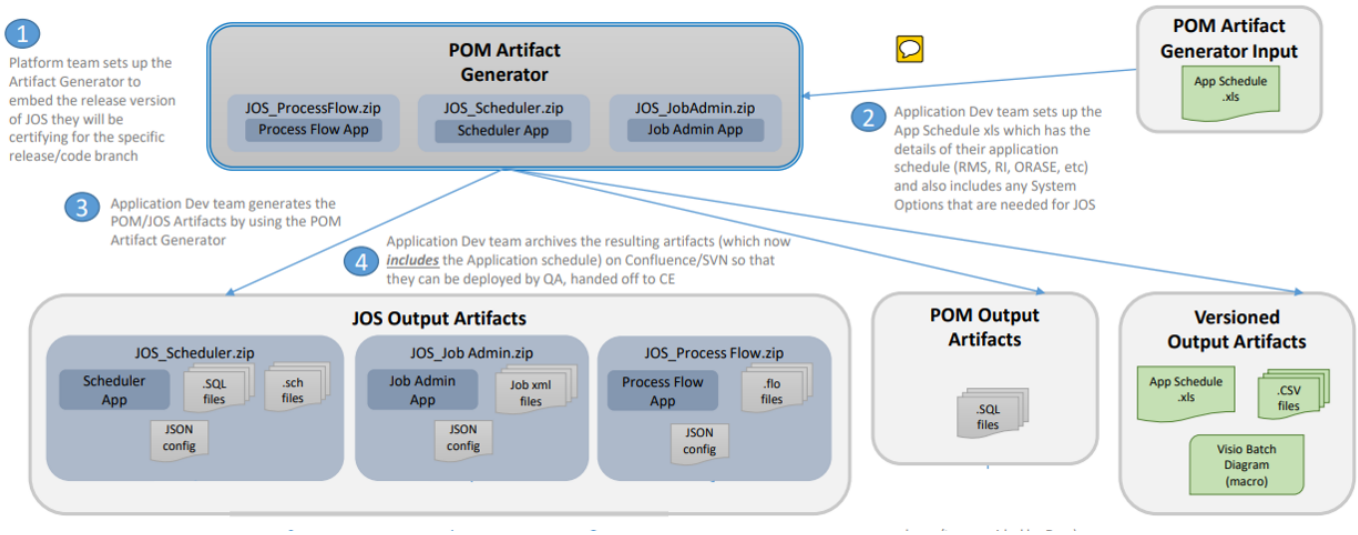

Figure 3.1: Artifact Generation Process

3.3 Implementation

3.3.1 Input with Macros Validations

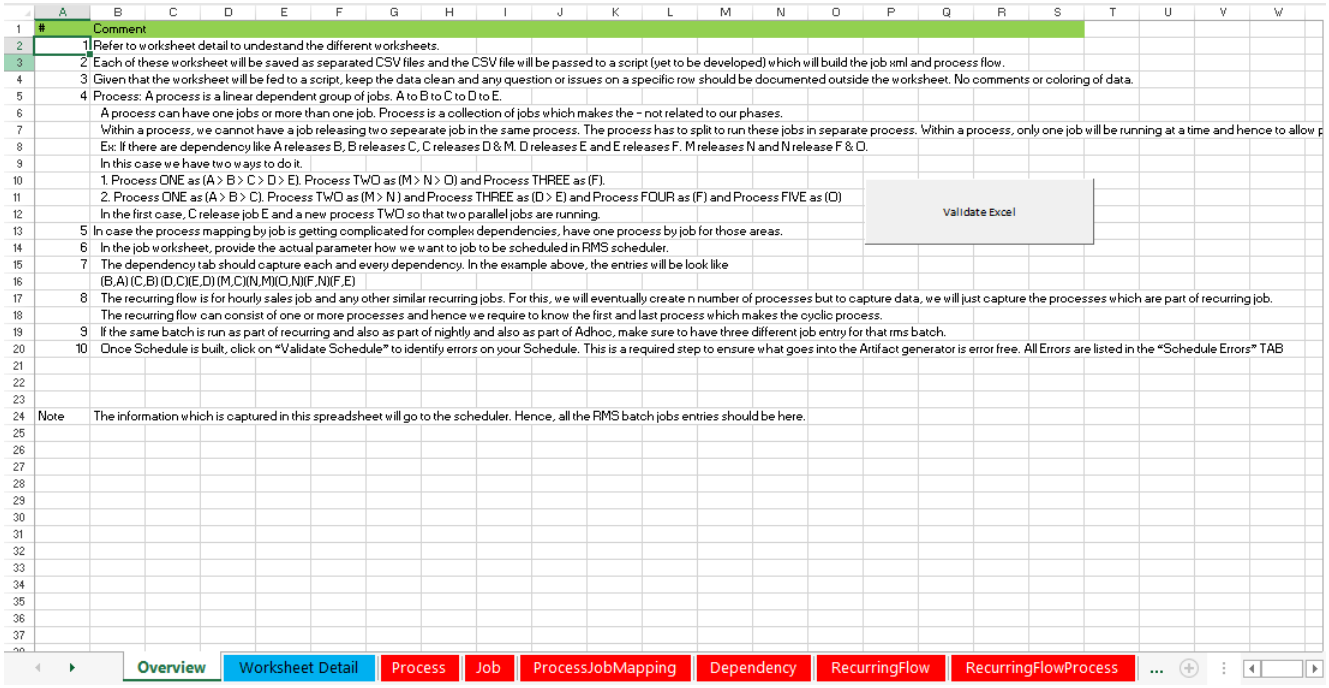

Figure 3.2: Sample spreadsheet shown for Merch-Batch-Schedule

3.3.2 Algorithm

The AG is built using Batch Processing Framework in Java-EE. All the applications in the framework includes XML files along with the java classes. The purpose of XML files in the application is to specify the structure of a job. It is basically in terms of batch artifacts and the relationships established between them. The steps of processing are as follows:

1. Design the batch application. Create a directory on remote server and place batch_application_name.jar and the input spreadsheet in same directory.
2. Define the application in terms of jobs. Job will consist of various steps along with decision elements. Determine the dependencies between them and store it in a TreeMap data structure.
3. Define the sequence of execution by forming terms of transitions between steps.
4. Identify steps that can run in parallel if any so that they can be executed using more than one thread.

5. Form the java class based on batch artifacts by implementing the specified interfaces by the batch processing framework.
6. Define jobs, steps, and their execution flow in XML files using the Job Specification Language. Each element present inside the xmp maps to the classes for the batch artifacts. The batch artifacts can be used to access properties defined inside the XML files like naming related details and the database connection related logic.
7. In order to execute the batch application, use the API given by the batch runtime.

3.3.3 Using the Job Specification Language

- Elements of Batch Job

  A batch job can contain one or more of the following elements:

  1. Steps: They can be chunk-oriented or task-oriented. Chunk-oriented steps can be partitioned steps. In a partitioned chunk step, the processing of one item does not depend on other items, so these steps can run in more than one thread.
  2. Flow: It is a sequence of steps that execute as a unit. A sequence of related steps can be grouped together into a flow.
  3. Split: A split is a set of flows that execute in parallel; each flow runs on a separate thread. The split transitions to the next element when all its flows complete.
  4. Decision elements: They use the exit status of the previous step to determine the next step or to terminate the batch job. [8]

- Job Instances and Job Executions

  A job definition can have multiple instances, each with different parameters. A job execution is an attempt to run a job instance. The batch runtime maintains information about job instances and job executions. [8]

- Example of JSL

Value	Description
STARTING	The job has been submitted to the batch runtime.
STARTED	The job is running.
STOPPING	The job has been requested to stop.
STOPPED	The job has stopped.
FAILED	The job finished executing because of an error.
COMPLETED	The job finished executing successfully.
ABANDONED	The job was marked abandoned.

Table 3.1: Batch Status Values

The Figure 3.3 shows the sample code for generation of XML file for jobs in RPAS schedule. The tokens are replaced by the values taken from XLSM sheet thereby generating the XML output.

```
<job id="$TokenName$" version="1.0" xmlns="http://xmlns.jcp.org/xml/ns/javaee">
  <properties>
    <property name="description" value="$TokenDescription$"/>
  </properties>
  <step id="batchlet-step">
    <batchlet ref="BatchShellCommandRunnerBatchlet">
      <properties>
        <property name="externalCommand"
value="#SysOpt.rpasDir/retl_scripts/rpas/rfx/src/$TokenScript$ #{jobParameters['jobParameters']}"/>
      </properties>
    </batchlet>
    <end on="COMPLETED"/>
  </step>
</job>
```

Figure 3.3: Sample of job template for RPAS Schedule

3.3.4 Output generation

- Steps to generate and test the output for populating the tables:

  1. Take complete maven build. This will generate the output .jar file for each cluster and AG.
  2. Create a directory on remote server and place BatchArtifactsGenerator.jar and the XSLM sheet in same directory.
  3. Run the .jar file.

4. Once the artifacts are generated success fully then run the seed-data-file in any of the database engine like SQL Developer.

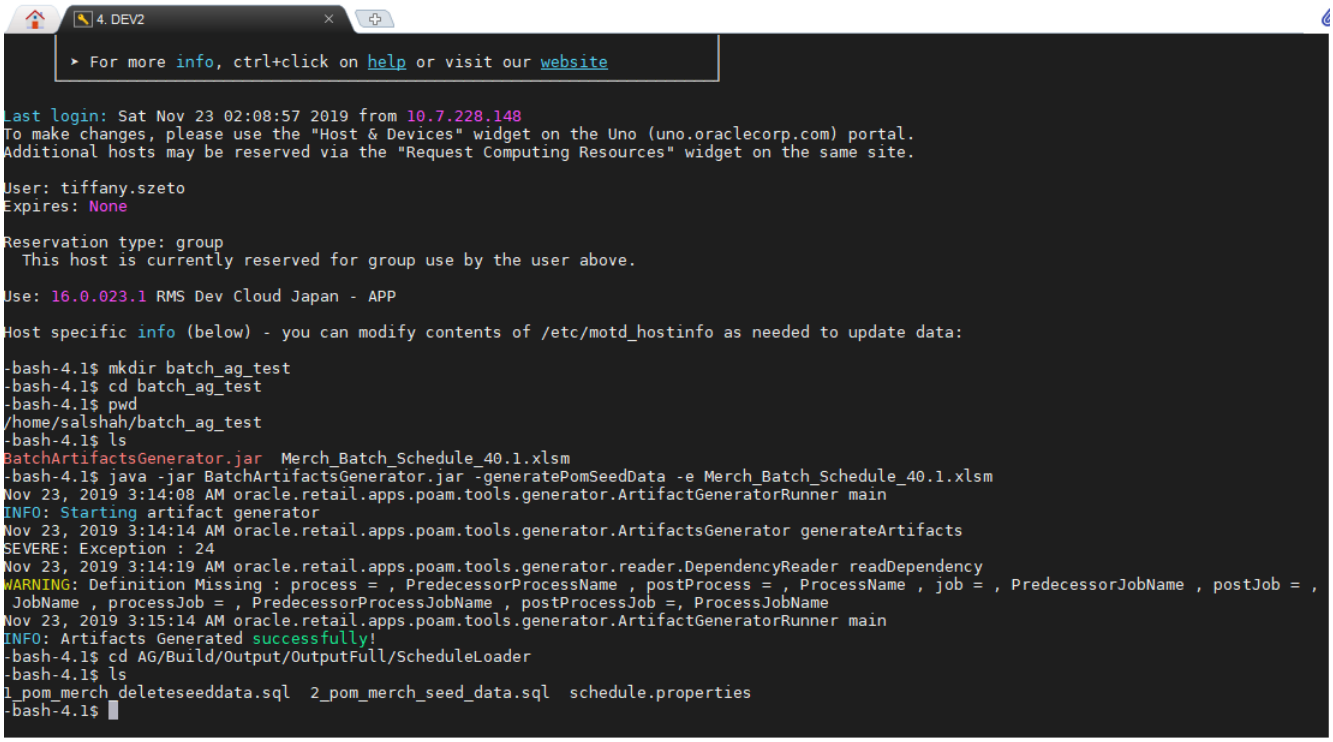

Figure 3.4: Generating output for AG

Chapter 4

Upgrading Batch Scheduler

4.1 Introduction

The Batch Scheduler, which is a part of JOS product suite, assists in scheduling of batch processes to run at predefined configured intervals. It basically determines when a job, a process, or any program must be executed along with the frequency of execution. From the implementation point of view, the Sheduler is based on a container-managed Java EE timer service for execution of schedules and utilizes Oracle Weblogic Server's implementation and management of timer service when deployed on a server. The Scheduler supports various schedules ranging from simple interval schedules such as hourly, daily, and so on, to advanced cron-like scheduling. It also supports the calling of REST services. The Console (Admin UI) enables runtime monitoring and administration of schedules where the user can view, create, edit, and delete schedules, manually run a schedule, enable or disable a schedule, set up notifications for schedules, and so on. [6]

- Schedule Types

  A schedule can be an interval-based schedule or a calendar-based schedule.

  1. Interval Schedules

     An interval-based schedule is a schedule that repeats at fixed interval of time starting from a specific time, for example, hourly, daily, weekly, every five minutes, and so on.

  2. Calendar Schedules

     A calendar-based schedule is a cron-type of schedule that specifies different

times that the schedule runs. More complex schedules that can be specified as a cron expression are defined as calendar-based schedules. The following parameters define a calendar-based schedule. These are the same as the parameters in a cron expression: Minutes, Hours, Day of Week, Day of Month, and Month.

- Scheduling Mechanisms

  1. Simple Scheduling

     Simple schedules are predefined schedule frequencies that are available as options for the user to choose. The following are the simple schedules that the Scheduler supports.

     (a) Hourly

     (b) Daily

     (c) Weekly

     (d) Monthly

     (e) Weekday [Mon-Friday]

     (f) Weekend [Sat-Sunday]

     (g) Saturday

     (h) Sunday

     (i) First day of every month

     (j) Last day of every month

     (k) One time only (run once)

     (l) User-specified frequency with intervals in the units minutes, hours, days, or weeks [6]

  2. Advanced Scheduling

     The JOS Scheduler supports advanced scheduling, which is cron-like scheduling. Calendar-based schedules that can be expressed in cron-format can be set up with the advanced scheduling capability of the Scheduler. Advanced scheduling is defined with the following parameters (similar to that of cron expression) and the corresponding range of values:

     Minutes: 0-59

Hours: 0-23 (12:00 a.m. - 11:00 p.m.)

Day of Week: Monday - Sunday

Day of Month: 1-31

Month: 1-12 (January - December)

If a schedule is created with multiple values for the above parameters, then the schedule will repeat at all those specified times. [6]

4.1.1 Salient Features of Batch Scheduler

- Batch configuration
- Batch Monitoring
- View Job Executions
- Batch execution for multiple retail applications
- Mail notifications and alerts
- External dependency/Inter-Schedule Dependency
- Execution Links
- Batch Throttling [6]

4.2 Auto-monitoring facility

4.2.1 Goal

POM and JOS are used by all the product teams of Oracle Retail. To run the batches, the product team will provide an XSLM sheet as input to AG. AG will generate the artifact and these artifacts will be given as input to Batch Scheduler. There are high chances that the product team might make mistakes in filling up the XSLM sheet. When the sheet with incorrect data is fed to the Scheduler, the software will waste 15 mins in processing the request and then will prompt the message of failure or some misleading transactions might occur in database.

The main objective of adding the auto-monitoring facility to the Batch Scheduler is to save the time of user and provide them support for early detection of mistake. Here,

Job No.	Job Name
1	RMS_CLEAN_UP_JOB
2	RMS_START_JOB
3	RMS_DAILY_JOB
4	RMS_DAILY_PRE_JOB
5	RMS_DAILY_POST_JOB
6	RMS_ADHOC_JOB
7	...
8	...
9	RMS_COMMIT_TRANSAC_JOB
10	RMS_PURGE_TEMP_JOB
11	RMS_COMPLETE

Table 4.1: Sample of jobs in a batch

the goal is to recognize the anomaly based on historic sequence patterns in the batches given by different products teams. Table 6.1 shows the dummy data for list of jobs in a sequence. Suppose the purge_job is mentioned in the between or the pre and post processioning of the job is out of sequence, then this might lead to inconsistent data entries or even failure sometimes.

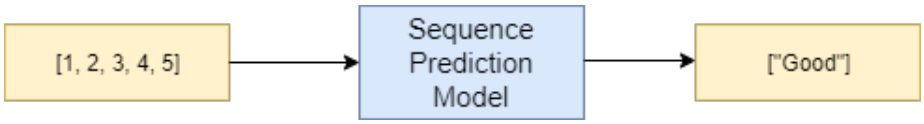

Figure 4.1: Example Model of Sequence Classification Problem

Sequence Classification problem states that given an input sequence, sequence classification problem involves predicting a class label. In Figure 4.1 the input numbers are mapped with the Job No. in Figure 6.1.

For example:

Given: 1 (job_1), 2 (job_2), 3 (job_3), 4 (job_4), 5 (job_5)

Predict: “good” or “bad”

The main goal of sequence classification is to create a classification model using a labeled dataset of batches which can be used to predict the class label of an unseen sequence. The input sequence may contain real values or discrete values. In the latter case, such problems could also be termed as discrete sequence classification. Here, we can use Anomaly Detection which an example of sequence classification problem. The problem for Anomaly detection is to predict whether the sequence is anomalous or not based on the given a sequence of observations.[11]

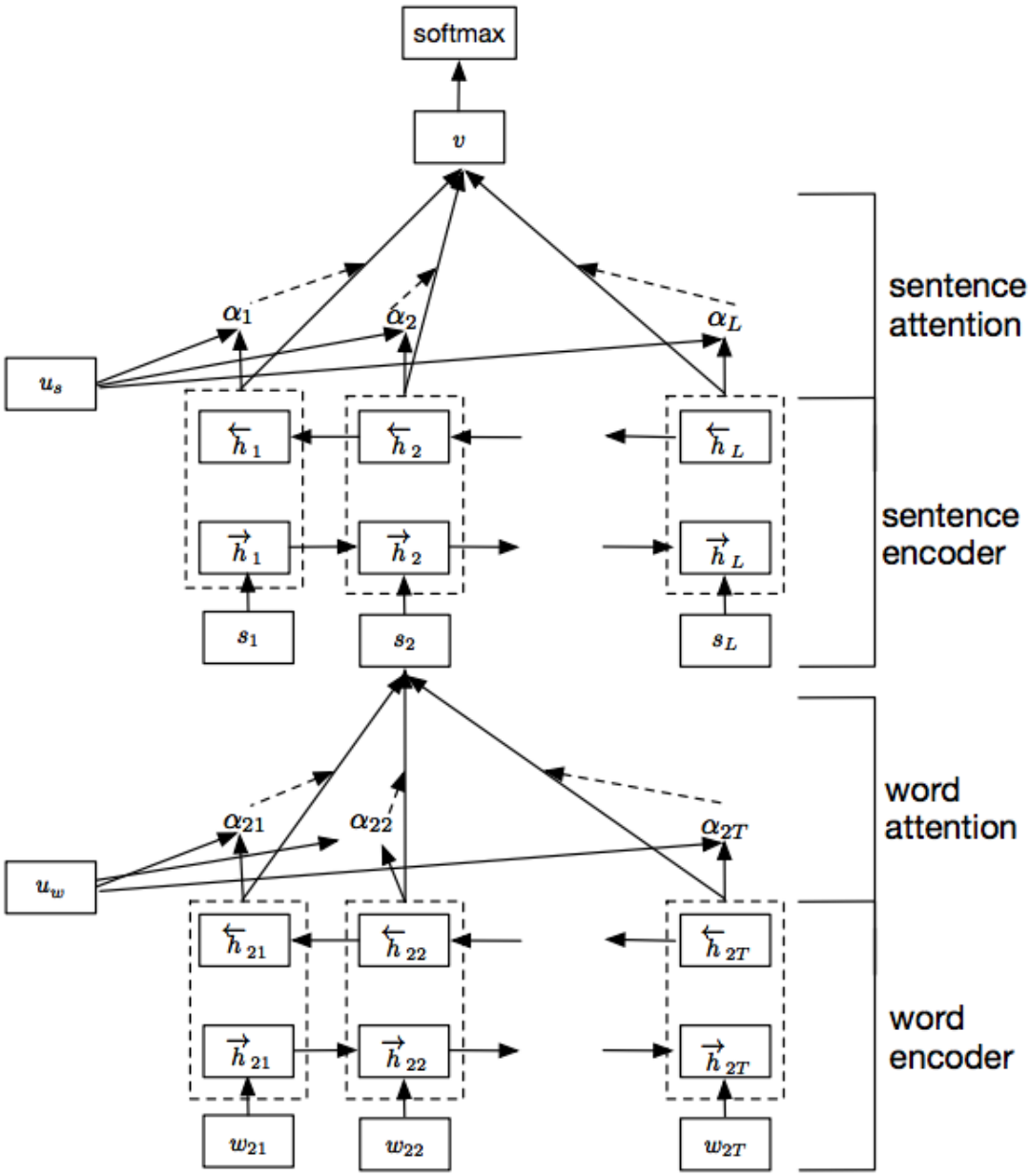

Figure 4.2: Hierarchical Attention Neural Network Architecture

4.2.2 Proposed Methodology

As per the analysis shown in the table 4.2 Hierarchical Attention Neural Networks outperforms in sequence classification. We can detect the malware using this algorithm and also classify it as a sample application. Here the term malware or malicious indicates to some of the harmful soft-wares or computer programs such as viruses, ransomware, spyware, adware, and others which are usually unintentionally installed and executed without the awareness of the user. In order to find such viruses, a typical pattern or sequence to analyze could be a set of disk access actions that the program has taken. To

Author	Methods	Yelp '13	Yelp '14	Yelp '15	IMDB
Yang et al. (2016)[13]	HN-ATT	68.2	70.5	71.0	49.4
Yang et al. (2016)[13]	HN-AVE	67.0	69.3	69.9	47.8
Tang et al. (2015)[12]	Paragraph Vector	57.7	59.2	60.5	34.1
Tang et al. (2015)[12]	SVM + Bigrams	57.6	61.6	62.4	40.9
Tang et al. (2015)[12]	SVM + Unigrams	58.9	60.0	61.1	39.9
Tang et al. (2015)[12]	CNN-word	59.7	61.0	61.5	37.6

Table 4.2: Accuracy analysis for various possible approaches

analyze software without running it, we can treat series of assembly instructions in the disassembled binary as sequences to be classified in order to identify sections of the code with malicious intent. In our novel approach, we apply techniques typically used to discover structure in a narrative text to the data that describes the behaviour of executable. This approach is potentially applicable to other use cases that involve sequential data, such as application logs analysis (detecting behaviour patterns), network logs analysis (detecting attacks), classification of streams, events, etc.[7]

4.2.3 Related work

The table 4.2 shows the comparative analysis based on the accuracy measure for 4 different data-sets for sentiment analysis: Yelp'13, Yelp'14, Yelp'15 and IMBD.[9] It clearly indicates that the attention based model outperforms in gaining the efficient result. An noteworthy comparison can be derived from the Yang et al. (2016)[13] model and a CNN. The support for extraction of automatic position-sensitive features is provided by both the models. However, after this analysis we can say that for our problem CNN would not be a good choice as it gives less efficient outcome. Each and every sentence is required to read twice that is one time in the forward direction and another time in the backward direction. The LSTM encoding can also support extraction of features of arbitrary length, because any aspect of the sentence context might be mixed into the token's vector representation. The same procedure is simply applied again to construct the document vector.[9]

Another essential factor driving the accuracy of the model is bidirectional LSTM encoder which is used for creating the position-sensitive features. This can be examined by replacing the attention mechanism with the men or average pooling pooling. The model still outperforms the previous state-of-the-art on all benchmarks. However, the attention mechanism improves performance further on all evaluations.[9]

Chapter 5

Migration Activities

5.1 Need to migrate from ARAF to JRAF

Technological drift is very frequent in the recent days. Within a fraction of second the present technology becomes obsolete. Be it from a larger scale like an entire technology or to a smaller scale like a small software tool, it becomes obsolete within no time. Hence enterprise firms are now migrating their applications to cloud. Java is the leading language used in most of the enterprise applications, and most of the Retail applications were developed using ADF. ADF is completely based on Java. But now, JavaScript is rapidly picking up prominence and by numerous records is the second most famous language today. JavaScript is extremely well known for front end improvement of web interfaces, and the respective development teams also had a keen interest in leveraging the advantages of JavaScript in their user interface. Hence as per industry standards, a toolkit named Oracle Javascript Extension Toolkit was developed. Using this, the Retail Application firms have developed JRAF that uses the JET for its UI and REST services completely at the backend. JET and ADF can integrate with JET being able to access the REST services that are exposed from the ADF Business Components.

5.2 Implementation of ReST end-points

For all the screens present in the application, the ReST endpoints are to be built. Till now, I have worked on three screens: Execution Engine Screen, Batch Monitoring Screen and Batch Scheduler Admin Screen.

Sample output for of two endpoints of Batch Scheduler Admin Screen:

1. /scheduler/schedules/scheduleName/cycles/cycleName/flows/flowName/tasks

   To fetch the Schedule Tasks for a given schedule name , cycle name and flowname.

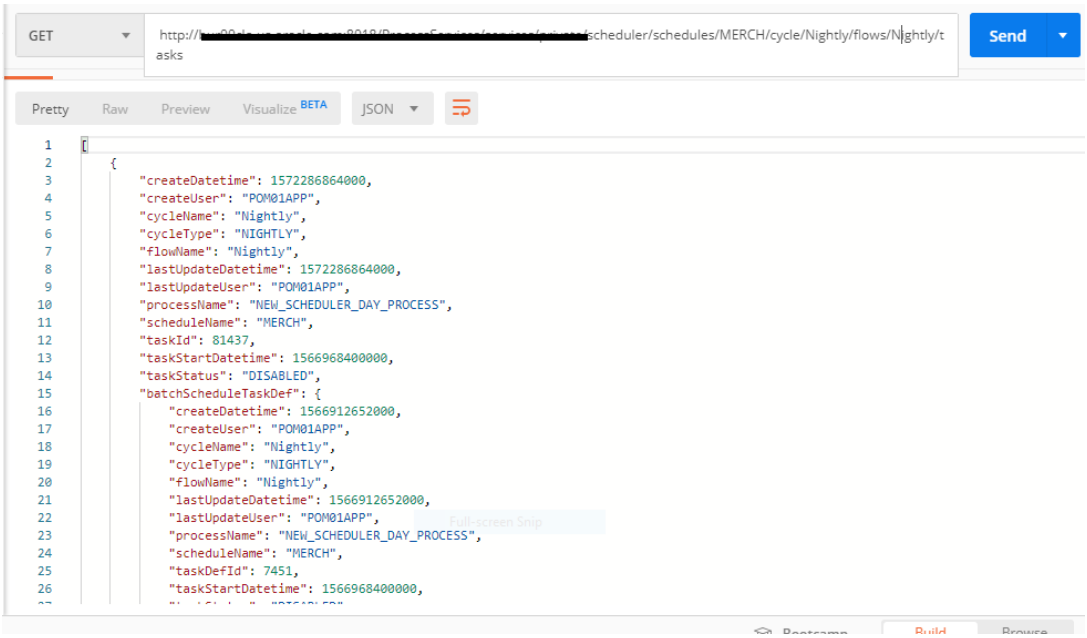

Figure 5.1: JSON response for endpoint 1

2. /scheduler/schedules/scheduleName/cycles

   Count of Schedule Tasks for all cycles for a given Schedule, by cycle.

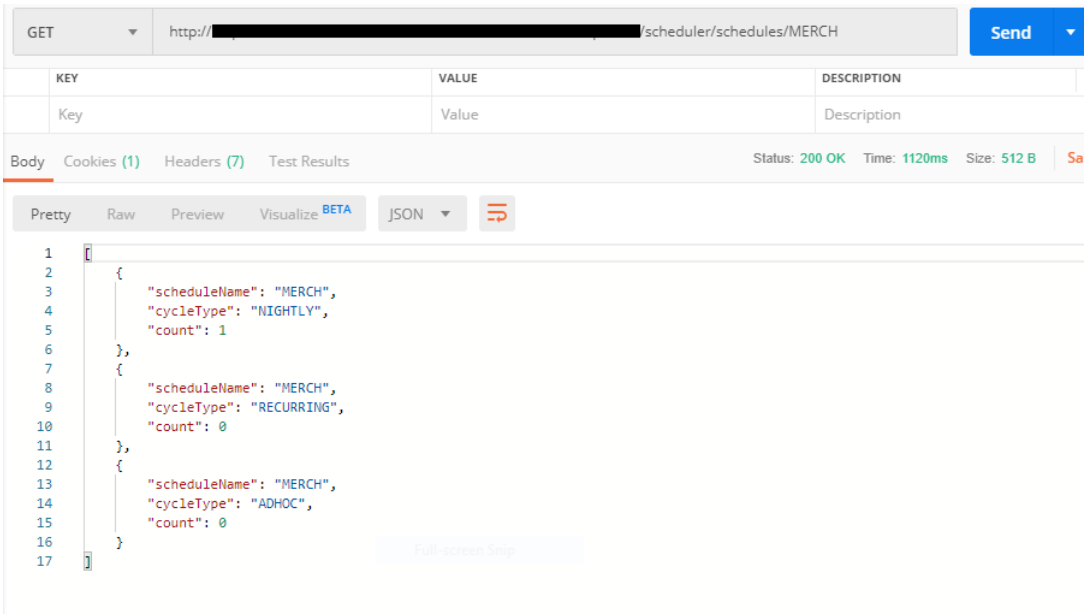

Figure 5.2: JSON response for endpoint 2

Chapter 6

Optimization Analysis

6.1 Selection of Server

6.1.1 Performance Analysis of Weblogic and Tomcat Server

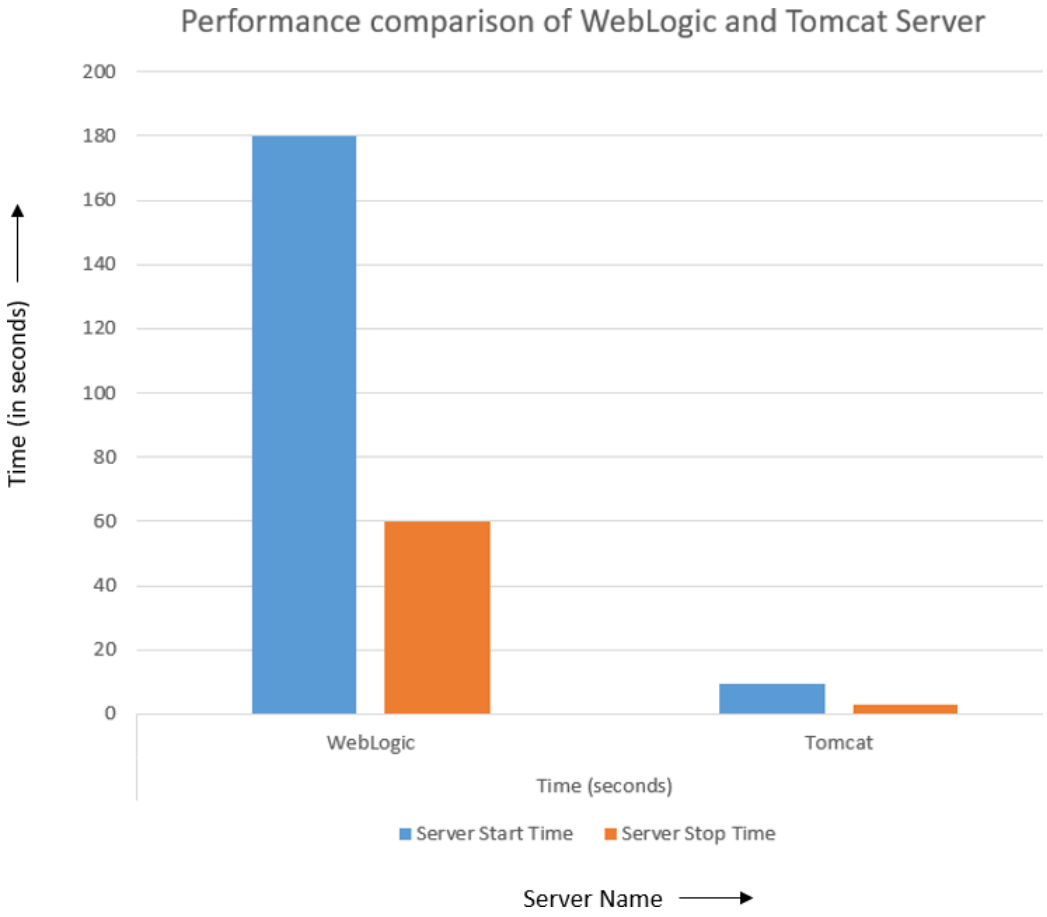

Figure 6.1: Performance analysis of WebLogic and Tomcat Server

WebLogic server was previously used as the deployment container for the Retail Platform

Performance Metrics	Weblogic	Tomcat
Time to start the server (mins)	3.0	0.16
Time to stop the server (mins)	58	2
Average execution time of GET request (ms)	2069.8	508.8
Average execution time of POST request (ms)	755.5	285
Average execution time of PUT request (ms)	508	371
Average execution time of DELETE request (ms)	691	60

Table 6.1: Performance analysis of WebLogic and Tomcat Server

services. Due to several issues with WebLogic, TomEE - a flavour of Apache Tomcat has been replaced as the deployment container. With the usage of TomEE server, the performance of the Retail Platform services has been boosted by 50-55% on an average. Figure 6.1 clearly shows the difference in the time to start and stop the server, with respect to both WebLogic server and TomEE server. It can be easily observed that the start time for TomEE is about nine times lesser than that of WebLogic.

6.1.2 Performance Evaluation of Retail Platform Services

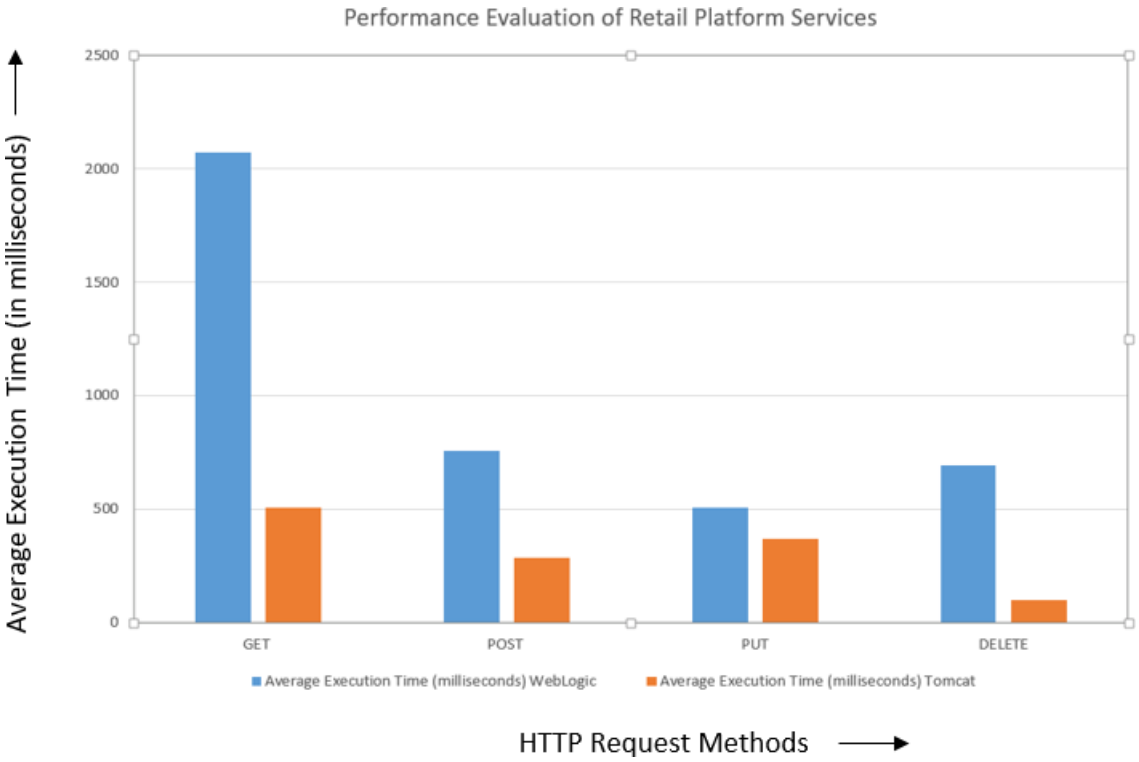

Figure 6.2: Performance evaluation of Retail Platform Services

Figure 6.2 shows the average time of execution of each of the REST service request methods. The average execution time is calculated for both the WebLogic as well as the TomEE server. It can be seen that the improvement in the performance of the Retail

Platform Services on an average is 50% – 55%. The inference that can be drawn is that the Retail Platform Services show excellent performance with respect to execution time of the resource requests on using the Apache TomEE server as the deployment container.

Chapter 7

Conclusion

In Batch Fusion Development, our aim is to automate a process that requires countless steps as per demand with minimum changes in code and towards better performance by optimizing the overall response time. After working with the Development team, I learnt to work with ReST API, different data formats like XML and JSON, brief about AI techniques, ADF and Spring Boot. I also learnt to maintain effective backend service to improve scalability and security for SaaS using agile and scrum methodologies.

Chapter 8

Future work

- Developing the auto-monitoring feature for Batch Scheduler to to include facilities like suggesting priority based on the type of jobs using dynamic scheduling algorithm.
- Integrating Job Admin and Process Flow into POM to reduce the number of ReST calls.

References

[1] https://www.tutorialspoint.com/restful

[2] https://www.tutorialspoint.com/xml

[3] https://www.w3schools.com/js-json-intro

[4] https://www.javatpoint.com/restful-web-services

[5] https://www.oracle.com/database/technologies/developer-tools/adf

[6] https://docs.oracle.com/cd/E8208501/16021/JOS

[7] https://machinelearningmastery.com

[8] https://docs.oracle.com/javaee/7/tutorial/batch-processing002.htm

[9] https://explosion.ai/blog/deep-learning-formula-nlp

[10] https://devblogs.microsoft.com/cse/2018/03/06/sequence-intent-classification/

[11] Thiago Andrade, Rita P. Ribeiro, Joao Gama, Wesllen Sousa Lima. *Anomaly Detection in Sequence Data: Principles and Case Studies*, Feb 2019.

[12] Rie Johnson, Tong Zhang, *Semi-supervised Convolutional Neural Networks for Text Categorization via Region Embedding*, Oct 2015.

[13] Yang, Zichao, Yang, Diyi, Dyer, Chris, He, Xiaodong, Smola, Alex, Hovy, Eduard, *Hierarchical Attention Networks for Document Classification*, April 2016.