International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 04 | Apr 2020 www.irjet.net p-ISSN: 2395-0072

# ANALYSIS OF FRAMEWORKS AT BACK END DEVELOPMENT FOR CUSTODY BANKING SECTOR

**Varsha S K[1], Dr.K. Badari Nath[2] and Anitha Sandeep[3]**

[1]*Department of Computer Science and Engineering, R.V College of Engineering, Bengaluru, Karnataka, India*
[2]*Assistant Professor, Department of Computer Science and Engineering, R.V College of Engineering, Bengaluru, Karnataka, India*
[3]*Assistant Professor, Department of Computer Science and Engineering, R.V College of Engineering, Bengaluru, Karnataka, India*

---

**Abstract –** *The generating of the report in the Traditional system was very slow and not much efficient and the data size was large day by day and the system works slow and they do not meet the SLA(Service Level Agreement) Example client want they report at 9AM EST but the system works till 11AM EST so we cannot achieve the time and size. So, the process is entirely migrated and redesigned to JE framework so the system works much fast and more efficient than the Traditional system. And they can keep track of data in table level and called by their parameter name. They can meet the SLA for huge amount of data receiving daily. Traditional system is used for scheduling the process execution. In this approach the number of Traditional system jobs increases. This makes difficult to understand the flow of application batch execution and maintenance of it. History of process execution is not maintained for longer period of time in Traditional system and if one fails it won't fetch the next job until it gets completed. So, the report does not get generated in time. In this paper, the key features comprising this library was analyzed and its advantages over the traditional system to the enhanced framework.*

## 1. INTRODUCTION

COUNTERPARTY STRESS TESTING:

In financial terminology, is an analysis or simulation designed to determine the ability of a given financial instrument or financial institution to deal with an economic crisis.

1. It is a computer simulated technique to analyze how bank and investment portfolios fare in drastic economic scenario.
2. Regulations require banks to carry out various stress-test scenarios and report on their internal procedures for managing capital and risk.
3. To determine whether a bank has enough capital to withstand an economic or financial crisis.

JE framework is designed to execute processes for daily/weekly runs in an application and maintaining the history of the process execution timings. Each process in JE Framework terminology is referred as job group. If a process comprising of many scripts, then the process can be conceptually split into multiple groups and have the dependency defined in the Database tables.

In most of the Risk projects, Traditional system is used for Scheduling the process execution. In this approach the number of Traditional system jobs increases along with the enhancement to current applications or addition of new applications. This makes difficult to understand and the flow of application batch execution and maintenance of it.

Also, the history of process execution is not maintained for longer time in Traditional system.

Through JE framework, the history of process execution is maintained in database tables. Hence Batch run statistics can be maintained per application record retention policy.

The application scripts can be parameterized also these parameters can be used across scripts, changes are done in table level if required. Parameter used for execution of runs has an ID and it will be tracked in the table level, this helps in identifying the issue in case of any job fails. It gets triggered and it fetches the next job.

## 2. Key features of Job Engine Framework

1. **Auto initialization and execution of job groups.**
   The job groups configured at JE Framework setup can be initialized automatically at the specified time in a day for the previous business date as well as current business date

as applicable. Autosys job will be pick up the job groups initialized. This autosys job is common to all job groups deployed in a server for a specified user.

2. **Initialization and execution of the job group for previous cob dates.**
   If a job group has to be re-run for previous business dates, JE Framework provides the facility to initialize and execute for the older dates also.

3. **Serial/Parallel execution of multiple application.**
   JE Framework provides the flexibility of configuring the job groups to run in serial or parallel.

4. **Serial/Parallel execution of jobs within the applications.**
   The jobs within a job group can be configured to achieve the serial or parallel execution.

5. **Cross server application execution.**
   JE Framework supports applications where the process specific scripts are distributed across multiple servers. In case, the autosys job for initialization and execution must running from bot the machines.

6. **Re-Running the application from a specific stage.**
   JE Framework provides the flexibility for running the job group from any stage. This is an important feature which is critical in case of any failures in the execution flow.

7. **Static Data Management.**
   JE Framework facilities to have the static data defined for each script at table level. This approach is easy to manage the static data and re-use them.

## 3. ANALYSIS METHODOLOGY

### 3.1 Analyzed framework

Traditional system was tightly coupled so that after one job completes it fetches the other job so it is hard to complete for 40000 million records so we move to the enhanced system with framework where one jobs fails it will be in pending status and then it picks up the next job so they have table level communication for the status of each jobs which is been run on the current business date. It can be modified in the table level to make an efficient flow of data.

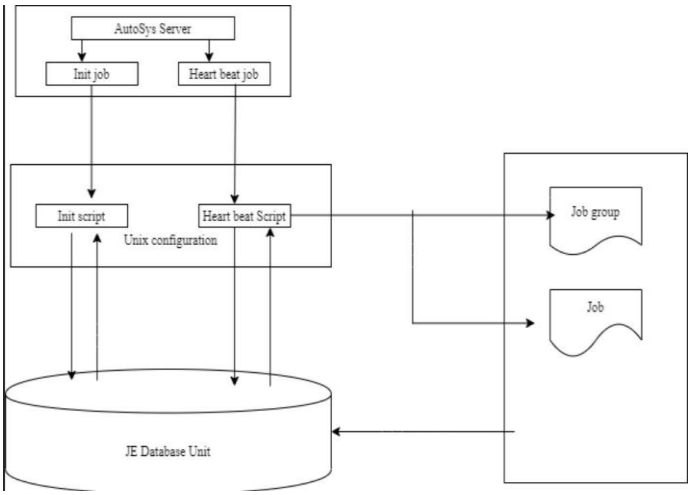


**Fig -1:** Working Architecture

**Steps for the migration from autosys to JE:**

1. Initialization and heart beat jobs start from Traditional system.
2. Initialization scripts queries DB to initialize the job group.
3. Initialization scripts keeps polling to pick the next job groups for initialization.
4. Heart beat job picks the initialized groups for execution.
5. Heart beat spawns the job groups to execute and polis for the next job group for execution.
6. Heart beat job spawns the job to execute under each job group.
7. The job group status and job status has been updated to DB.

#### 3.1.1 JE Framework components:

1. **Traditional system Configuration:**

Traditional system jobs are created to initialize and execute the job groups through JE Framework. Traditional system jobs are common for all job groups using JE Framework for running the batches from a UNIX machine.

2. **UNIX Configuration:**

Configuration at UNIX level is required to have the data sourcing and profile information. The JE shell scripts and the configuration files are placed in a separate path with respect to the application shell scripts. This provides the flexibility of using the JE framework for multiple application hosted on UNIX machine.

**3. Database Configuration:**

The JE framework tables and other Database units are deployed to the same Database where the application DB objects are deployed.

### 3.1.2 Benefits against the Traditional system

Below are the goals achieved by configuring the job groups in JE framework. These benefits both the application support teams and business users.

- Multiple Traditional system jobs are not required for scheduling the application scripts. The scripts names to be configured in the JE framework tables. These facilities to easily view the application execution flow at table level instead of going through the Traditional system dependency definition.

- Maintaining the history of the job group execution for previous dates. This helps to view the job groups execution timings/statistics and the success rate of the job group run.

- By using JE framework, the application (business logic) scripts can be parameterized. Also, these parameters can be used across scripts. This way the parameter can be changed at the table level if required. Since the parameters can be shared across the scripts, optimal number of parameters will be defined for the application.

- The parameters used for each execution run will be tracked ion the table level. This helps in identifying the issue in case of any job failures.

- Write the set of scripts for the multiple insert and select query for the different scenarios and the manipulation of the code in been done in scripts and placed it in the file watcher and stages are been loaded. send a block of statements to the database, significantly reducing traffic between the application and the database and write compact code for manipulating data.

- The status of all the jobs is been monitored and tracked in the table so that easy to understand the flow data and check the logs of data failure and complete of jobs.

### 3.1.3 Different Frameworks

1. **UNIX shell scripting:** A program which is having a series of OS commands that will be executed by the command-line interpreter or the shell in sequence. Because each command is combined to form a "script" that the shell follows and executes by its instruction.

   **Advantages:**

   - The user no need to change to entirely different syntax as the syntax and commands. They are exactly the same as those directly entered in the command line can be quickly write and start the shell scripts, interactive debugging.

   **Disadvantage**:
   - shell scripts are like slow execution, minimal data structure unlike other scripting languages, little difficult to remember all the commands, the complex flaws design

2. **PERL:** It is general purpose High-Level Programming Language developed by Larry Wall. Perl (Practical Extraction and Reporting Language).

   **Features**
   - Dynamic Programming Language
   - Used in Graphical Programming
   - used for scripting, and the Platform to create tools in System Administration
   - Network Programming, Bio-informatics and Finance are its other area of Application.
   - Camel, the symbol of PERL is not officially declared.
   - They are Procedural Programming
   - Perl borrows a lot of features from programming languages like c, Lisp, AWK, sed, etc.
   - Used for glue language, working between two distinct interfaces.
   - Often implemented as a core interpreter.

   **Advantages:**
   - Similar like Shell Language
   - Traditional Approach by using Braces for functions and Loops.
   - Powerful Programming Language

- Versatile
- Multi-purpose Language
- Mature Language
- Imperative, Procedural, Functional or Object Oriented, depends upon requirements.

**Disadvantages:**

- Plenty ways to achieve same result, means unreadable code, which in turn means untidy code
- It is slower for a lot of tasks.
- Object Oriented not implemented well
- Creates problem when the codes are larger than 200 Lines.
- Argument Handling is poor
- Not Portable
- No Interpreter Shell
- Ugly Libraries

3. **PL/SQL:** Procedural language for structure query language. Where the multiple queries can be run on basis of procedures and functions.

**Advantages:**

- Block structure: blocks of code, which can be nested within each other. Each block forms a unit of a task or a logical module. Blocks can be stored in the DB and reused.

- Procedural language capability: procedural language constructs such as conditional statements (if else statements) and loops like (FOR loops).
- Better performance: engine processes multiple SQL statements simultaneously as a single block, thereby reducing network traffic.

- Error handling: exceptions effectively during the execution of a PL/SQL program. Once an exception is caught, specific actions can be taken depending upon the type of the exception or it can be displayed to the user with a message.

**Disadvantages:**

- Proprietary to Oracle.
- Poor I/O Features.

## 4. DISCUSSION

The analysis shows that the different frameworks like Traditional system and library like JE framework well based on the following application criteria:

JE framework is feasible for applications because of the following:

- Better learning
- Binding is better
- Good number of usages
- Time reduction
- Applications that require versatility
- High speed

Traditional system is less suitable though for applications. Still some of the reasons why Traditional system should be chosen are:

- It is having a good structure
- Initial job is fetched from the Traditional system then dumped into JE frameworks.

JE framework is a concept which is important to increase the performance of the report generating by doing some
Manipulations. It is one of the many advantages of JE framework and is a really important concept.

## 5. CONCLUSION

This paper examined the use of JE framework is primary for the back-end technologies and for the creating Report generating. The back-end frameworks JE framework and Traditional system were compared and JE framework was definitely a better choice. JE framework was discussed in details and its advantages and disadvantages were listed too. The criteria for selection of specific framework were clearly identified and JE framework should be chosen in most of the cases.

Also, one more concept was discussed as well that is nothing but the different languages and its advantages on how it increases the performance of report generating and also reduces the delay in submission of report.

## 6. REFERENCES

[1]. Kernighan, Brian W.; Pike, Rob (1984), "3. Using the Shell", The UNIX Programming Environment, Prentice Hall, Inc., p. 94, ISBN 0-13-937699-2,

[2]. Unix Shells By Example, pp 7-10, Programming Perl, 5th Edition, preface "osh - manned.org". manned.org. Retrieved 2019-01-16.

[3]. Johnson, Chris (2009). [1] Pro Bash Programming: Scripting the Linux Shell, Apress, Retrieved on September 27, 2019. ISBN 9781430219989 "Shebang symbol (#)".

[4]. Robbins, Arnold; Beebe, Nelson H. F. (May 16, 2005). Classic Shell Scripting: Hidden Commands that Unlock the Power of Unix. O'Reilly Media, Inc. p. 10. Retrieved May 7, 2017.

[5]. Carling, M.; Degler, Stephen; Dennis, James (2000). Linux System Administration. Sams Publishing. p. 275. Retrieved May 7, 2017.

[6].Kumari, Sinny (November 23, 2015). Linux Shell Scripting Essentials. Packt Publishing Ltd. Retrieved May 7, 2017.

[7]. Robbins, Arnold; Hannah, Elbert; Lamb, Linda (2008). "Learning the vi and Vim Editors". p. 205.

[8]. Easttom, Chuck (2012). "Essential Linux Administration:: A Comprehensive Guide for Beginners". p.228. "Csh Programming Considered Harmful".