International Journal of Mathematics and Statistics Invention (IJMSI)
E-ISSN: 2321 – 4767 P-ISSN: 2321 - 4759
www.ijmsi.org Volume 5 Issue 3 || March. 2017 || PP-31-36

Job Failure Analysis in Mainframes Production Support

Ranjani KV[1], R. Roseline Mary[2]
[1](Department of Computer Science, Christ University, Bangalore India)
[2](Department of Computer Science, Christ University, Bangalore India)

Abstract: *A major part of batch processing on mainframe computers consists of several thousand batch jobs which run every day. This network of jobs runs every day to update day-to-day transaction. There are frequent failures which can cause a high delay in the batch and also degrade the performance & efficiency of the application. Permanent solution can be done to frequently occurring job failures to avoid the delay in batch and to improve performance & efficiency of the application. In this paper, we have analyzed the frequently occurring batch job failure recorded in Know Error Databases (KEBD) for past one year based on different categories. Frequently failed jobs obtained are categorized based on application, failure-type, job-runs and the resolution. Different results are obtained in the weka tool based on the different categories. From the various results obtained it can be concluded that the frequent failures are occurring in MSD application. On further analysis on this frequently failed jobs showed that data and network issue are causing the major job failures in which most of the jobs were daily processing jobs. In order to fix the failure the jobs was resolved by restarting the job from the overrides or by restarting the job from the top.*

Keywords: *Batch jobs, Failure analysis, Know Error Databases (KEDB), Resolution, weka*

I. INTRODUCTION

Batch processing on mainframe computers consists of several thousand batch jobs which run every day. This network of jobs shows the day-to-day business transaction that are updated during the night with interrelations requiring scheduling and prioritizing the jobs to assure all batch jobs run in the scheduled order within Service Level Agreement(SLA). The job scheduler helps in identifying the times at which the job runs on specific days and the dependencies of the batch job will also be seen in the scheduler. Scheduled jobs run on specific days and at different times ensures updating on business holidays as well without any loss of data. The execution of some jobs is dependent on the other jobs because output data from first job is used as an input data to second data. This data dependency is also associated with various upstream and vendor. Scheduler also helps in identifying the status of the job i.e. under the execution, waiting for other jobs to complete, arriving of data or file from the upstream or vendor, Error if the job has failed etc. There would be a delay or postponing in the job run due to dependent job in failed state which further delays application batch and downstream jobs dependent on the failed jobs. The resolution of that failed job is fixed by referring to the Know Error Databases (KEDB). KEDB contains previously failed job details which include the Job Name, Application, Return code, Error Message, Resolution and the person who resolved it. If the job details are not present in the KEDB, then it is first time failure for particular job and the record is added for future reference. Based on the previously occurred error in the production the job failures are categorized based on input arrival times and the type of failure occurred.

Based on the failure occurred that are stored or updated in the KEDB, the failure is categorized into different types like technical issue, network issue, contention, space issue, data issue, cancellation, new jobs, developer mistake or incorrect scheduling. Based on these categorizations the analysis is made on, how many times the job failed due to same error and the resolution done to fix the error.

To analyze the pattern of the job failures, The KEDB for the previously failed jobs for the past one year is obtained. With this the frequent job failure are analyzed for following categories:

- The type of the job that has failed (like the processing, transmission, database)
- The job failure type (like data issue, network issue, database, deadlock)
- The action taken to fix the job failure at that time (like the job was restarted from top, restarted from failed step, marked the job as complete).
- The job names.

Based on the categorization, analysis is done to improve performance and avoid the recurring failures of the system by implementing or suggesting the permanent fix for the specific job. This analysis is done with the help of weka tool. The job failure dataset obtained for the past one year is taken and inserted into the weka for the analysis and the respective results obtained are tabulated are shown in this research paper.

II. PROBLEM DESCRIPTION

A batch jobs on mainframe will often run every day to update day-to-day transaction and there are frequent failures. These failures can cause a high delay in the batch as well as degrade the performance and efficiency of the application. In order to avoid the delay in batch, degradation of performance & efficiency of the application a permanent solution could be done to frequently occurring job failures. This is done by analyzing the pattern of the job failure and the resolution steps taken to fix the failure. Once the pattern frequency of the failure and the resolution steps to fix are analyzed, the failure can be avoided in future by fixing it permanently by analyzing the pattern of the job failure with the historical failures recorded by the support team in a project for keeping record of the failed jobs.

III. LITERATURE REVIEW

In [1] ,the paper "Job Failure Analysis and its implications in a Large-scale Production Grid" determines an analysis of job failures in large-scale data intensive grid. Job failures in large-scale heterogeneous environments are due to a variety of possible causes, system problems which was due to node, disk or network issues. Errors also occurred at different levels as the software stack was more and more complex. Based on the job failures they are represented in three periods in the production, characterize the inter-arrival times and life spans of failed jobs. Different failure types are distinguished and the analysis is carried out. Based on the failure pattern, historical failure is taken into account in decision making. Based on the analysis the cooperation and accountability issues are briefly addressed, evaluated the effectiveness and feasibility.

In COBOL application (Banking system) critical outages were due to common causes. " Towards Assuring Non-Recurrence of faults Leading to Transaction outages – An Experiment with Stable Business Application" [2] determines that, to reduce the cost and efforts for maintaining a legacy business application was a challenge for capturing faults at early stage of software – known to prevent defects in production. Analysis was performed on these common causes to detect the causes using structured and COBOL flow analysis. From an structural analysis and control flow analysis techniques the faults was automatically detected using the all occurrences of the faults which would potentially lead to multiply failures during production.

In [3]" Towards a Training-Orientated Adaptive Decision Guidance and Support System" determines that, strategic approaches are needed to troubleshoot system failures by first identifying the component causing failure to solve the problems. In this paper they have addressed the domain of administration of DB2 on z/OS Mainframes system. The framework dynamically extracts knowledge from various correlated data sources containing system related data from the problem solving procedures of human experts. The research paper applies text and data mining techniques for knowledge extraction a rule based system for knowledge representation and problem categorization and case based system for providing decision support.

Based on the error codes categorization for the job failures" Getting back Up: Understanding How Enterprise Data Backups fail" [4] determines that, the jobs that run on each system are monitored and checked if they are completed successfully. Error characteristics are done based on the production, development and test, Number of unique error codes Number of most frequent error codes. Error causes are due to misconfigurations, system error and information messages (unusual). The above characterizations are done with the historical data and the analysis performed for decision support.

IV. METHODS AND MATERIALS

To analyze the pattern of the batch job failures in mainframe computers, the KEDB for the previously failed jobs for the past one year is obtained. The most frequent job failures were recorded and analyzed with the weka tool upon these categorization:

- Application
- Failure-type
- Resolution
- Job Type
- Job runs

1. Application: In this research paper the failed jobs are categorized into application based on the different servers the job runs.

2. Failure-Type: The job failure are classified as below Data, Network, Delay, Deadlock

a) Data: The failure is classified as data issue is there is any discrepancy in the file received. Examples of data issues could be due to the following reasons:

- Incorrect file received from the upstream or the vendor.
- Junk values in the file which has caused the data inconsistency.

- The format of the file is not as expected for example; the data or the values in the files are not in the correct format as expected.
- Missing values in the file or the file is empty.

b) Network: The interaction with batch processing is mainly through a network of transmitting or receiving the files from either one server to another or through DB2 tables. The failure is classified as network issue when there is any issue in interacting with the batch processing. Examples of the network issue could be due to the following reasons:

- Server unavailability to while extraction the file or uploading the file during the job execution.
- Resource unavailability, this could be due to the file or the table is been used by other jobs.
- System or the application down while the user is trying to access the data.

c) Delay: When there is any feed delay from the upstream or the vendor the batch jobs go into the failed status. The delay would happen due to the following reasons mentioned below:

- The file is very huge (that is, contains more number of records than the expected.
- The scheduled release activity either in our application or in the upstream.
- The jobs going into contention waiting for the files which are used by the other jobs.

d) Deadlock: When the batch is executing concurrently, a deadlock can happen when one job is trying to access the file or database and is waiting on the other job for release the lock on that file or the database.

3. Job-Type: The job type specifies the type of job failures that happen while the batch is running. The failed jobs in past year recorded are categorized into the following:

a) Processing: The mainframes batch jobs that run during the night updates the data or the transactions that has happened in the day time. The processing jobs have failed due to the following reasons:

- Insufficient storage: While updating the day-to-day transactions into a dataset or DB2 tables.
- Null values: When there is an empty file or dataset received or missing data in a specific column.
- Incorrect data formats: When the data received is not in the same format the values usually received like the date formats or the characters in place of numeric values.
- Load Balancing: When the jobs are automatically submitted on different CPU's where the specific application jobs do not have access to run on that CPU.

b) FTP Transmission: File Transfer Protocol (FTP) transmission is process of transmitting the files between servers.

c) NDM: Network Data Mover (NDM) transmission is process of transmitting the files between servers with a direct connect by installing the server details at both the end before transmitting any details between the servers. This type of transmission is much faster and more secure while comparing with the FTP transmission.

The FTP transmission and NDM transmission jobs have failed due to the following reasons:

- Connectivity/Access Issue: When the server details are not installed at both the ends so the jobs fail due to access issue while accessing the servers.
- File unavailability: When the file is not available as the file generation at the upstream is still in progress.

d) Database: The database jobs are the jobs that append the latest data into the database and take the backup of those databases into the datasets for future reference. These jobs failed due to the following reasons:

- Resource/ Datasets Unavailability: Where the specific database or the table is in use by other batch jobs.
- Space Issues: While updating the day-to-day transactions into a dataset or DB2 tables.

4. Job-Runs: The Job-Runs categorized with Daily, Weekly, Monthly jobs based on the pattern the job runs.

a) Daily: The jobs are categorized into daily where the jobs runs daily that is Mon-Friday or Tuesday to Saturday.

b) Weekly: The jobs are categorized into weekly where the job runs any one day of the week.

c) Monthly: The jobs are categorized into monthly where the job runs once in month.

5. Resolution: The resolution steps taken to resolve the job failure when it has failed are categorized in the following categories:

a) Restarted-the-job-from-top: The failed jobs are restarted from the top when they have failed before executing either due to access issue or network connectivity issue.

b) Restarted-the-job-failed-step: The failed jobs are restarted from the failed step when the processing of the jobs is interrupted due to the resource unavailability, network issue, unexpected return code from the job, etc.

c) Marked-the-job-as-complete: The failed jobs are marked as completed when all the processing completed and the job has thrown any acceptable return code.

d) Restarted-the-job-with-the-overrides: The failed jobs are restarted with the overrides when the job has failed due to syntax errors, space issue or file not available due to delay or any other reasons where the job needs any manual modification to complete the job successfully.

V. RESULTS AND DISCUSSION

The various results obtained with the above categorization with Weka tool for the records are shown as below with the respective observations.

1. Application: Based on Application, below figure Fig.1 shows the classification for different categories. The MSD application has the more number of job failures compared to the IMS application jobs in the past one year. 78.61% of the times the jobs have failed in MSD application and 21.38% of the times the jobs have failed in IMS application.

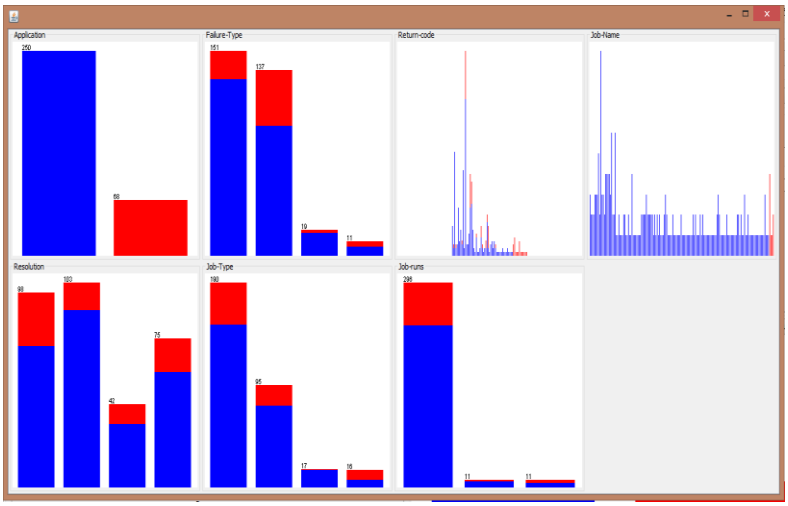

Fig. 1: Application classification for different categorizes

2. Failure- Type: Based on Failure-Type, below figure Fig.2 shows the Failure-Type classification. It shows that there are 47.48% of the times the jobs are failing due to the data issue, 43.08% of the times the jobs are failing due to Network issue, 5.74% of the times the jobs are failing due to delay and 3.45% of the times the jobs are failing due to deadlock.

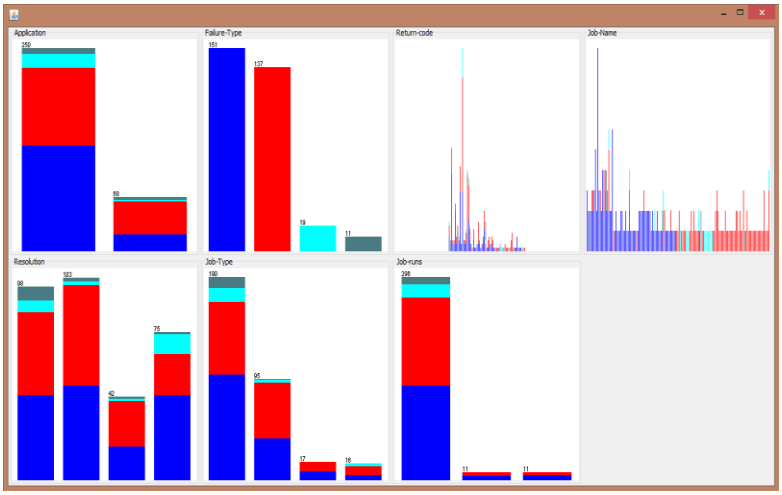

Fig. 2: Failure-Type classification for different categorizes

3. Resolution: Based on Resolution, below figure Fig.3 shows the Resolution-Type classification. It shows that there are 30.81% of the times the failed jobs have resolved by restarting the job from the top, 32.38% of the times the failed jobs have resolved by restarting the job from failed step, 13.20% of the times the failed jobs have resolved by marking the job as complete and 23.58% of the times the failed jobs have resolved by restarting the job with the overrides.

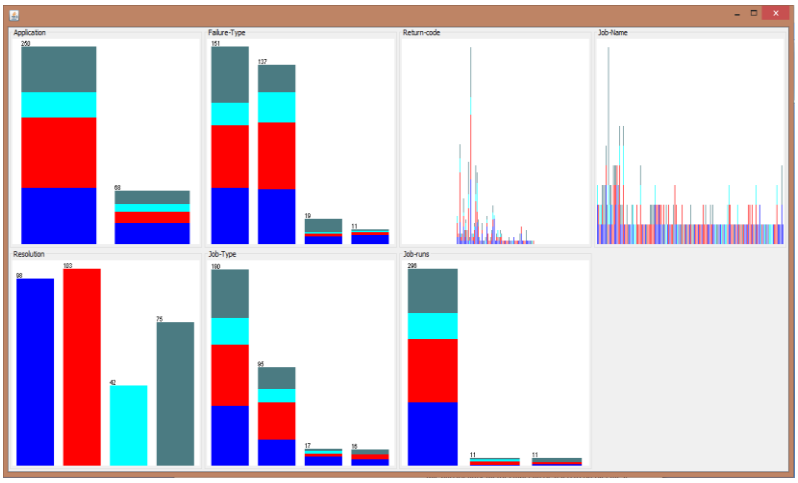

Fig. 3: Resolution classification for different categorizes

4. Job-Type: Based on Job-Type, figure Fig 4 shows the Job-Type classification. It shows that there are 59.74% of the times the Processing jobs have failed, 29.87% of the times the FTP Transmission jobs have failed, 5.34% of the times the Database jobs have failed and 5.03% of the times the NDM Transmission jobs have failed.

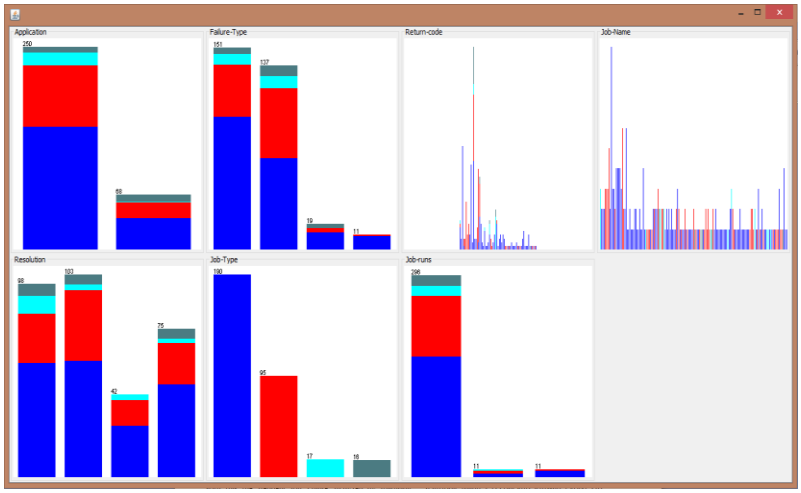

Fig. 4: Job type classification for different categorizes

5. Job-Runs: The below figure Fig 5 shows the Job-Run classification. It shows that there are 93.08% of the times daily jobs have failed, 3.45% of the times weekly jobs have failed and 3.45% of the times the monthly jobs have failed for the job failure taken for the past one year.

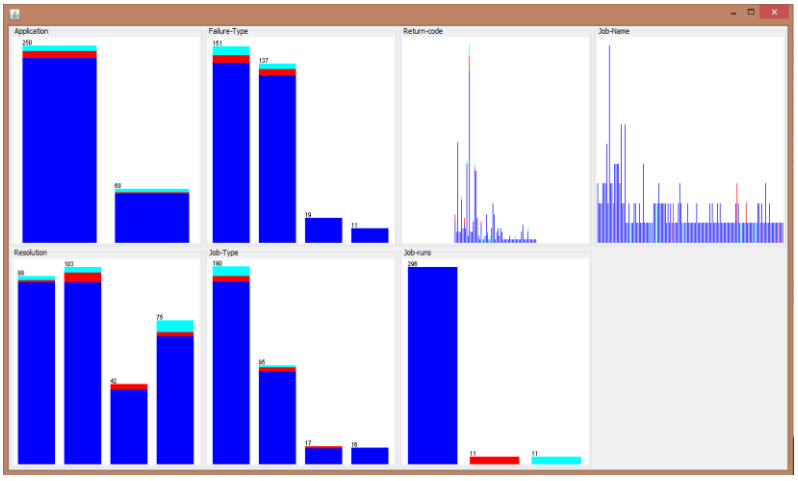

Fig. 5: Job Runs classification for different categorizes

VI. CONCLUSION AND FUTURE ENHANCEMENT

With the analysis of the pattern of the job failures recorded in KEDB for the previously failed jobs for the past one year are obtained into different categories. The results obtained in the weka tool for the frequent job failures based on the different categories are tabulated as below:

Categories	**Classifications**	**Job Failure in %**
Application	MSD	78.61
Failure-Type	Data Issue	47.48
	Network Issue	43.08
Resolution	Restarting the job from the top	30.81
	Restarting the job from failed step	32.38
Job-Type	Processing	59.74
Job-Runs	Daily	93.08

The above result implies, job failure behavior based on the various classifications and the percentage of job failures with same error. These results help us understand the major failures occur during night batch processing, with the type of failure occurring and the resolution preformed to fix the failure.

For further enhancement, the results could be analyzed with respect to the specific jobs. By analyzing the specific job failure with the same error can provide more accurate results to reduce the frequent job failures. These recurring failures can be permanently fixed by implementing the changes in source code so as to avoid the future job failures for the same error. This would help in increasing the performance, efficiency and stability of the system.

REFERENCES

[1]. H. Li, D. Groep, L. Wolters and J. Templon, "Job Failure Analysis and Its Implications in a Large-scale Production Grid", Proceedings of the Second IEEE International Conference on e-Science and Grid Computing (e-Science'06) 0-7695-2734-5/06 $20.00 © 2006.

[2]. A. Agrawal and R. Naik, "Towards Assuring Non-recurrence of Faults Leading to Transaction Outages – An Experiment with Stable Business Applications", *ISEC'11, February. 23–27, 2011, Thiruvananthapuram, Kerela, India. Copyright © 2011 ACM 978-1-4503-0559-4/11/02…$10.00*, 2011.

[3]. F. Zulkernine, P. Martin, S. Soltani, W. Powley, S. Mankovskii and M. Addleman, "Towards a Training-Oriented Adaptive Decision Guidance and Support System", 2010.

[4]. G. Amvrosiadis and M. Bhadkamkar, "Getting Back Up: Understanding How Enterprise Data Backups Fail", 2009.

[5]. S. Kavulya, J. Tan, R. Gandhi and P. Narasimhan, "An Analysis of Traces from a Production MapReduce Cluster", *10th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2010). May 17-20, 2010, Melbourne, Victoria, Australia.*, 2010.

[6]. "Differences Between NDM and FTP", Difference Between, 2017. [Online]. Available: http://www.differencebetween.net/technology/differences-between-ndm-and-ftp/.

[7]. "7 Benefits of Using a Known Error Database (KEDB)", *The ITSM Review*, 2017. [Online]. Available: http://www.theitsmreview.com/2012/04/7-benefits-of-using-a-kedb/.