Log – Analyzer

Steps to configure Eclipse:

- Import the project into eclipse with Log-Analyzer.zip file.
- All the dependencies will be added with the help of pom.xml file except database driver jar. Please add respective jar in Build Path. (I have used ojdbc8_g.jar)
- Please update the /WEB-INF/dbConnection.properties file with corresponding database details.
- Add respective Web/Application server in Targeted Runtime Path (I have used Apache Tomcat v9.0)
- Add a table in database to store models. (sql file attached)
- Now start the server and run the application on this server.
- Now you will get a simple web page which will ask you to upload file on the page.
- Please select a file and click submit.
- The information available in Log File will get stored in data base and you will get a success page.
- In case of exception or file validation error, you will get a very simple exception and error page respectively.
- First upload after restart the server could take few seconds extra, after that processing speed would be good enough.

Approach Taken: (Flow Chart also attached)

As we have a Log file with bundles of records and each record has same format. Also as per given example every unit of record is starting from date. So I have used regular expressions (which can be differ for different format of log file) to find out the unit blocks and sending those unit blocks for further processing.

Now from those Unit blocks I am taking each key, value pair and converting this unit block in java model object with this key, value pairs.

Once I get the model object, I add them to List of model objects which has a fixed buffer size (in my case 1000). Once the buffer limit reaches this list will be added to data base using batch insertion and list will be reset to take other object models.

To support multitasking I have used ThreadExcecutorService, so with help of this we can process the unit blocks immediately we receive it. So we don't need to store this unit data in our heap. So we can prevent our program from throwing Out Of Memory Error, even for a file with 2 GB size.

Also I have used two ThreadExcecutorService(s) in my program to boost the speed of the process. So that few threads are working only on data insertion and few are working only on token generation.

I have used Oracle 12c to store the data in data base, so that it can be fetched easily in any order type at any time.

An video has also been attached to show the working of the applications.

State any improvements that you would like to make if more time was available for implementation: In my opinion we can use Elasticsearch database as it provides scalability, no schema and real-time indexing. So for this application, it can provide better performance as compare to Orcale (RDMS) database when numbers of records approach to a very large number.

Any assumptions made: As of now I am excepting only text/plain file. So application will not accept any other file than this. Also I have restricted this application for max file size of 2 GB. For a file with 2 GB it is working very fine, we can exceed the file size if it is required.

Find JAVADOC for this application at with below link:

.../Log-Analyzer/javadoc/index.html