Sivaprasad.valluru@gmail.com

Whatsapp: +91 8088910831

LAB- batch job and oneway synch usecase

Using batch job

In this lab, you will be working on 07-onewaysynch-using-batch-solution project. So, import it

Open 03-batchprocessing.xml and understand the logic.

Run the the application and Copy src/main/resources/products.csv under c:\files\inputbatch folder

Observe the logs and analyze the behavious of batch job.

Can you tell the purpose of Batch Aggregator in step 1?

Can you tell why I didn't use batch aggregator in step2 ?

What is the purpose of configuring "accept expression" in step2?

What is the payload logged by the logger in "on complete" phase?

Sivaprasad.valluru@gmail.com

Whatsapp: +91 8088910831

Oneway synch using ontablerow

In this step, you will be working on **05-oneway-nearrealtimesync-usingontablerow-solution** project. So, import it.

Create 2 schemas with name db1 and db2 in mysql server using mysql workbench.

Open db1_product.sql and db2_product.sql under src/main/resources and execute them on your mysql

After executing scripts, u should observe that there are around 6000 products in products table of db1

Now we want a realtime sync between db1 and db2.

Open onewaysyncusingontablerow.xml and observe the flow. This is a straight forward flow which reads from db1 and write to db2

Run the application and observe that all records in db1.products are written to db2.products

Problems with above flow : For each record a mule event is generated and it is inserted to db. So, if there are 6000 records in db1, 6000 times my app is firing insert query which results in poor performance.

If the number of records in db1 are expected to be considerably less and you want near real time sync, this approach is better.

If number of records are more and u prefer to do batch inserts, we will see the second approach

Sivaprasad.valluru@gmail.com

Whatsapp: +91 8088910831

Oneway synch using scheduling and foreach

In this step, you will be working on **06-onewaysync-using-scheduling-foreach** project. So, import it

Open onewaysync-using-scheduler-foreach.xml in this project and observe the flow.

Actually, we have to use a scheduler in the source part of the flow.

But just for demo purpose I have added Http Listener.

We are using Object Store also to store and retrieve the lastupdatedtime.

Go through all the components of the flow and understand them

Click on “ForEach” scope and observe that we have given batch size of 200.

Also we have used “BatchInsert” to insert 200 records at a time.

Truncate products table in db2 now

Run this application and make a request to /start.

After you get the response, you should observe that db1 is synced with db2

Problem with above approach:
If you are using For Each, single thread will be used for iterating over all records. So, we are not achieving parallel processing of records.
Also, if there is an exception while processing one record, rest of the records are not processed.
We can keep components in for each scope in a try scope and handle errors. But again it will become too complicated.
So, better to use batch job.

Sivaprasad.valluru@gmail.com

Whatsapp: +91 8088910831

Oneway synch using batch job

In this step, you have to work on **07-onewaysynch-using-batch** project
So, import it

Truncate products table in db2 now

Open onewaysynch-using-batch.xml and observe that we have replaced foreach with batch job.

This flow is straight forward and self explanatory if u know batch.

Now start the application and give a request to http://localhost:8081/start

See logs and observe how records are parallel processed by multiple threads.

By default, for a batch job, 2*(cores on your machine) threads will be used. So, if u are using octa core processor, 16 threads at maximum will be used.

Now I want 200 threads(maximum) to process all 6000+ records.

So, Click on Batch Job and configure max concurrency as 200.

Now restart the application and give a request to http://localhost:8081/start

You should observe that nearly 200 thread will process all the records concurrently.

Problems with this approach
All the records will be processed on a single machine even though you deploy your application in a cluster.