Sivaprasad.valluru@gmail.com

Whatsapp: +91 8088910831

LAB- Understanding thread pools in a non-transactional flow

In this exercise, you have to work on **03-threadpools-nontransactional-solution**

So, import the corresponding jar given to you

Before going further, make sure that you have mysql8 server and Mysql Workbench installed on your machine.

In Mule runtime 4.2.x, there were 3 thread pools at runtime level names as CPU_LITE,CPU_INTENSIVE and BLOCKING_IO.

In Mule Runtime 4.3 or above , there is only one thread pool named "UBER".
In the logs you will observe that threads from UBER threadpool only will be used for executing CPU_LITE,CPU_INTENSIVE and BLOCKING_IO operations

Open threadpoolsdemo.xml and observe the flow and its components.

In this Flow, logger does CPU_LITE Operation, DB Select and Insert operations perform BLOCKING_IO operations and "Transform Message" performs CPU_INTENSIVE operation

Open log4j2.xml under src/main/resources and observe that I have enabled DEBUG level logs and for most of the packages, I have configured level as OFF

Now, run the application and give a request to http://localhost:8081/test

Observe the logs from various threads.
Observe how threads are switched for various types of Operations

Sivaprasad.valluru@gmail.com

Whatsapp: +91 8088910831

Logs will look like below:

```
INFO  2024-01-26 09:01:23,476 [[MuleRuntime].uber.04:
[03-threadpools-nontransactional-solution].threadpoolsdemoFlow.CPU_INTENSIVE
@2aecb4ba] [processor: ; event: 60de4970-bbfb-11ee-940e-a46bb69b43fe]
org.mule.weave.v2.model.service.DefaultLoggingService$: Got PID as  - 1 as Number
{class: "java.lang.Long"}
```

In the above log, `[[MuleRuntime].uber.04` tells the thread number is 04 and it is from uber thread pool and `CPU_INTENSIVE` tells that this operation is CPU_Intensive operation

Can you tell why the log written by “Set Variable” is labeled as BLOCKING_IO?

Now Wrap Set Variable inside Async scope , redeploy the application and give a request to http://localhost:8081/test

Now the log written by “Set Payload” is labeled as CPU_LITE

Congratulations!! You understood how Thread pools are used in a mule flow when there is no existing transaction

Understanding thread pools in a transactional flow

In this lab , you will be working on 04-threadpools-transactional-solution project

Open muletrainingdb.sql given in src/main/resources and observe it.   Execute this .sql file on your mysql8 database

After executing the .sql file observe that there is only one product in products table.

Open dbdemo.xml and understand the flow

Click on try scope and observe that it is configured to always start a **new local transaction**.

Sivaprasad.valluru@gmail.com

Whatsapp: +91 8088910831

Run the application.

Observe that polling done by a thread not belonging to uber thread pool. It is names like "`_pollingSource_dbdemoFlow/executor.01`"

Observe that single thread from "uber" thread pool is used for the whole flow.

Congratulations!! You understood how thread pools are used in a transactional flow!!