Foreach We can set batch size and we can store data to Db using Bulk Insert

But With Parallel For Each we cannot use bulk insert why because we cannot set the batch size in Parallel for Each

Foreach cannot modify the Actual Pay load.

Also we cannot access the Foreach data outside foreach we want we can declare the Variable out side and Append the Data Inside the for each and then we can use it outside the For Each.

Below is the DW expression to form the Scatter gather response to single Array of objects.

flaaten(payload..payload)

We have to set scatter gather request to variable and then pass to the calls in scatter gather

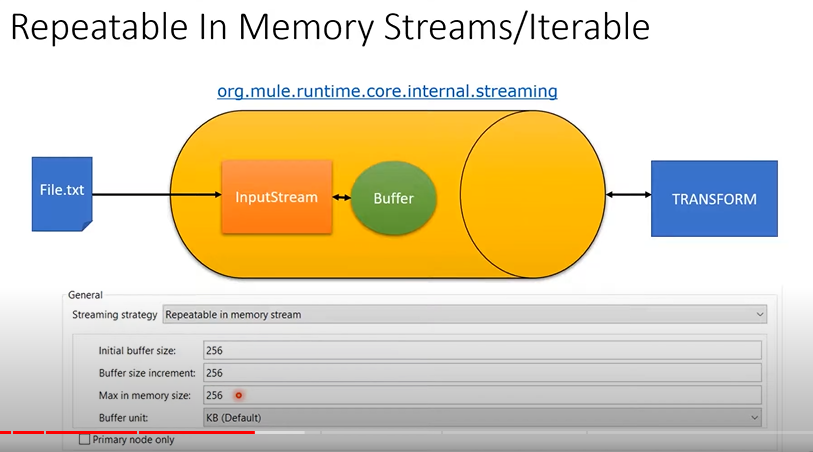
1. Insert
2. Select
3. Bulk Insert
4. ForEach|Parllel ForEach
5. Choice
6. Scatter Gather
7. Cache
8. Try

**Streams in Mule:**

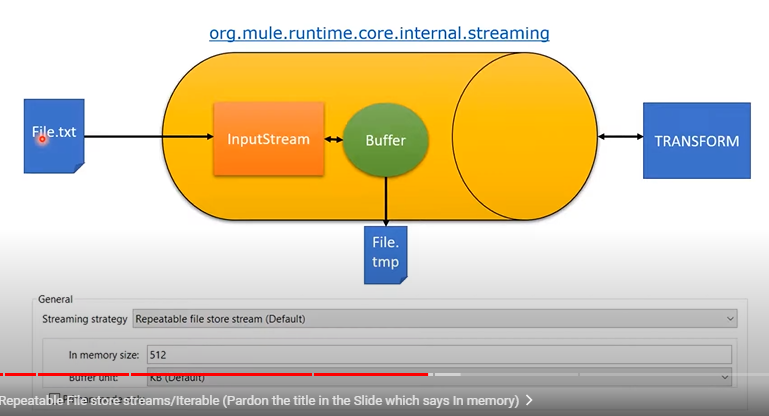
1. For Reading less Data from DB or file it is good to store in a **collection**
2. For Little more Data we can go for **Pagination** but still same time if we get 100s of requests then still we may get memory issue.
3. Streams Strategies will solve the problem in Mule

**1.Non Repeatable:** It will available for only immediate consume but not available to next consumer. It will be read only once and Memory Efficient.

**2.Repeatable:** It will be stored in a InMemory or File storage for available to process next components.



**Storing data to File System:**



it increase’s 256 then extra 256 will get store data then Max is 512 then next data over flows.

* Initial Buffer Size: 256
* Buffer size Increment: 256
* Max in Buffer size: 512

Buffer Unit: in MILLIS

* 1. In Memory
  2. file Storage

**Streams:**

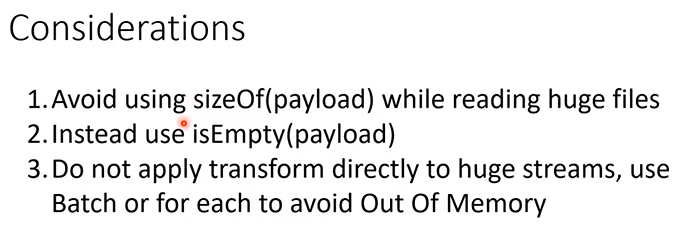
1. DB: Select
2. HTTP: Listener ,Request
3. File: List,Read,On New or Update File

**Stream Strategies:**

**Streams :** Http/On New or Update on File

**Itarables:** DB/File/FTP/SFTP

1. If we have 5 to 10 BM of data then it is good to have repeatable Stream with in memory
2. Always The components will selected with Repeatable default why because it should be available to next connectors to process data.
3. If we have more than 100 MB then go for File store and not good to use in memory.
4. If we are consuming millions of records then **We should not immediately** use the Message transformer instead of it we should use for Each or Batch Processor then use Message Transform



**Around 10000000 records with 10 columns with minimum 10 size string values = 1gb**

**MQ vs VMQ:**

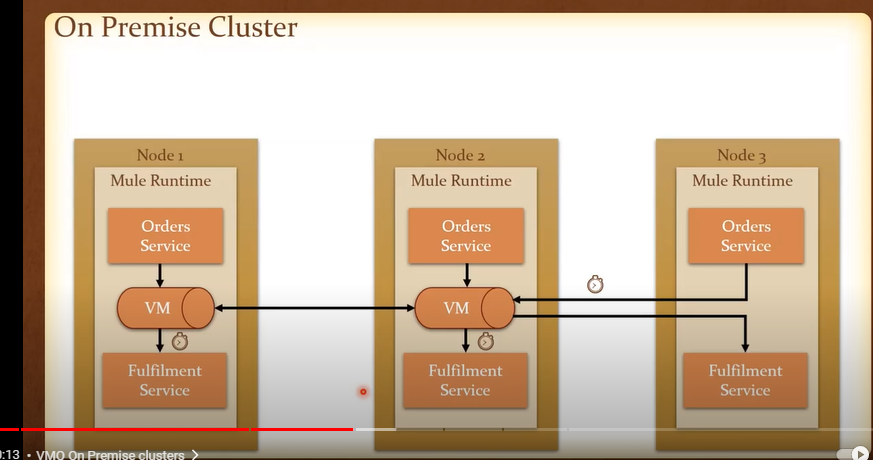
VMQ vs MQ may get shut down because of some reasons.

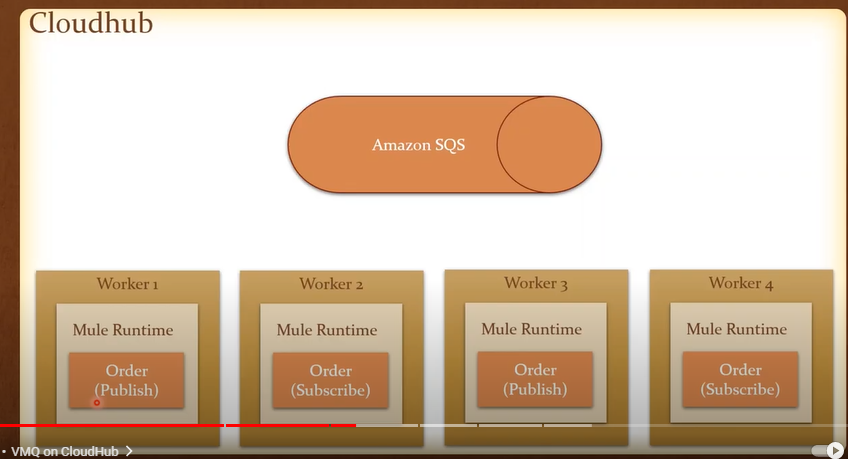
If we uses out side MQ then Our Published service and Consumer services are up but the AMQ Broker

VMQ is a part of Mule Runtime and uses Mule runtime Inmemory .

Transient [Inmemory] and Persistence[Disk or File system] VM Queues.

Always On Prmise uses Persistance VM Queue





In On premise 2 applications can access the single queue

But in Cloud it is not possible one application can access only one Queue But 1 application deployed in different nodes can access the same Queue

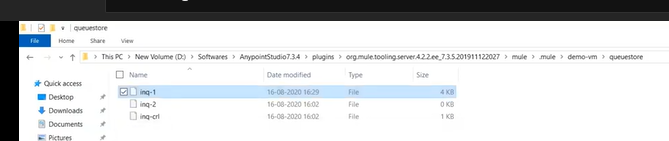
Always Mule will uses the Persistence Virtual Messaging Queue to process Data like Batch Process uses Persistence Queue.

But always Outside the Companies who uses Queue in Mule uses the outside queues because of Support and Better way of handling messages.

There is no URL to see the message count or message in VMQ.

Also we can set the time out for the queue .Suppose Time out is 1 hr then message comes in at 10AM then it will get expires on 11 AM.

We can see the VMQ message under this folder but we need to deserialize and check.



1. Publisher: Publish the message to Queue
2. Listener: Consume the message from Queue
3. Publish Consumer: It will expect the Response from the Listener .If Listener wont listen the we will get message like No Listener to consume message.

**APIKit Router and Console:**

1. Yes APIKit autogenerates your flows and routing based on your api spec. It generates the backend impl flows, error handling and status code mapping and more.
2. The router is for routing the HTTP request to the correct flow based on the method, path, uri Params etc. It introspects the raml and maps the request to the correct flow.(you can also provide manual mappings)
3. The apikit console is not mandatory. It provides an auto generated interactive HTML console for interacting with your api for testing etc. It is auto generated from your RAMl/api spec. It
4. In studio it will pop up as a view pane when you start the app containing the console flow and you can use the url you defined to access the console in a browser also. Usually something like /api/v1/console It’s the same type of console that is in Exchange and API designer for interacting with your api.

