

# MEP 5 -- Refactor query(search & retrieve) workflow

Current state: Under Discussion

ISSUE: [#6493 Query Nodes that only load segments will get stuck when searching](#)

PRs:

Keywords: QueryNode / Search / Retrieve

Released:

## Summary

Currently we have only one service time which would block query(search&retrieve) on multiple channels and historical. While historical doesn't need to wait for new tSafe, and current channel doesn't need to wait for tSafe from other channels. So we need to refactor query(search&retrieve) workflow and decouple the dependencies between historical and channels.

This proposal is about

- The details on how to refactor query workflow.
- Test plan of new query workflow.

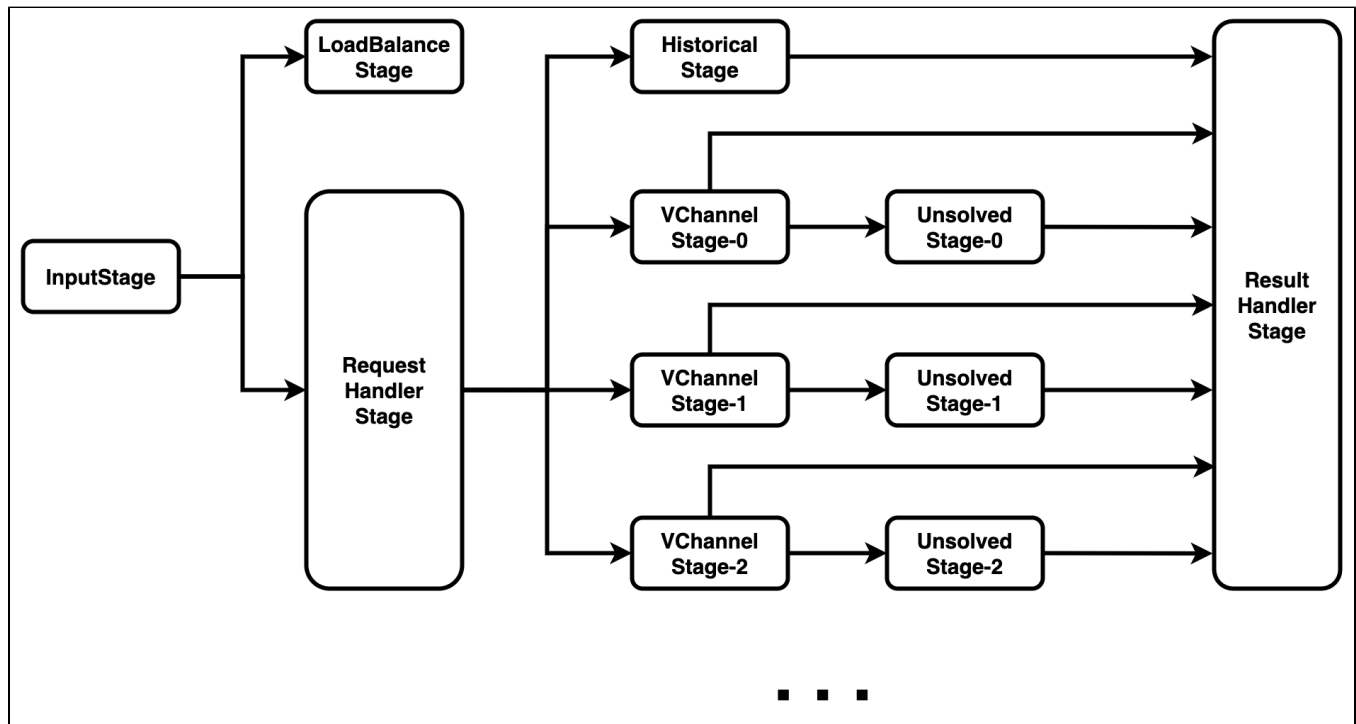
## Motivation

1. **Improve query performance::** Decoupling the query dependencies between historical and channels, query would be more efficient.
2. **Fix issue that query would get stuck:** A bug fix on issue #6493.

## Design Details

Make query workflow into multiple stages, they are InputStage, LoadBalanceStage, RequestHandlerStage, HistoricalStage, VChannelStage (VChannelStage have one or more, depending on the number of VChannel in the current Collection), UnsolvedStage and ResultHandlerStage

Stages are connected to each other by Golang's channel. Each stage will run one or more goroutines in the background to excute the query task. The query workflow design is shown in the following figure:



Stages are independent of each other, that is, after the current query task is done, the next query task can be excuted immediately, without taking care of the status of other stages. The details of these stages are as follows:

1. InputStage: Consume from queryChannel of msgStream, and group messages by the message type (query, loadBalance) and send them to loadBalanceStage or RequestHandlerStage.
2. loadBalanceStage: Excute loadBalance task.
3. RequestHandlerStage: Query task preprocessing (such as parsing DSL, creating query plan, etc) and send it to HistoricalStage and VChannelStage.
4. HistoricalStage: Query in Historical, get the query result and send it to ResultHandlerStage.
5. VChannelStage: Check the query time, when the service time doesn't get to the query time, the current task would be sent to UnsolvedStage, then return and excute the next query task immediately; when the service time has been get to the query time, excute the query task immediately and the query result is sent to ResultHandlerStage.
6. UnsolvedStage: A for-loop triggered by the update of TSafe that will continuously check the query task time, and execute the tasks which service time has been get to the query time, and send the result to ResultHandlerStage.
7. ResultHandlerStage: Reduce query results and produce the results to query result channel of msgStream.

## Compatibility, Deprecation, and Migration Plan

### Test Plan

1. Unit tests about all stages and unit test of simple query on queryCollection.
2. Test of query on the QueryNode which only have historical data.
3. All ci test cases about query.

### Rejected Alternatives

### References