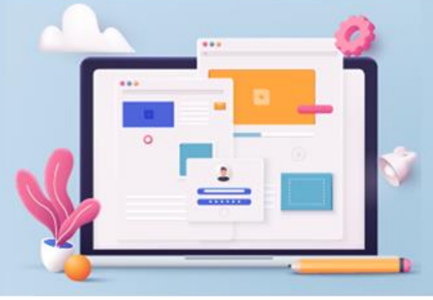

FINACLE SUPPORT CONNECT

KNOWLEDGE SERIES

January 2024 | Volume 91

Finacle Support brings you this fortnightly knowledge bulletin to augment your problem-solving capability. Every edition is put together with utmost diligence to ensure that best practices and known resolutions are shared. In this edition, you will find the following articles:

So, let's start reading!

Performance Tuning Approach for NEFT Inward Message Processing

Product: Finacle Core Banking Version: 10.x onwards

Banks having a high volume of **NEFT** inward messages are not able to process all inward **NEFT** messages within the stipulated time window.

For **NEFT** inward message processing, there are two services configured:

1. Service to pick the message from MQ/middleware and push the message to Finacle and insert to the **SMH** table (mqneftin → swmqssrv), logic similar to **HUPLPMSG** CORE menu which is File Based
2. Service to pick the inward message in ready status from **SMH** (swiftsrv), for the creation of transaction

The below points can be implemented to improve performance for the first service which picks messages from MQ/ Middleware and inserts them into the **SMH** table.

Split logic can be based on:

- Single app server and multiple queues: The bank can maintain different queues by balancing the load by putting one type of messages or any defined logic and each queue can have one service mapped to it so that the assigned service will pick the message from the dedicated queue. The parameter **INPUT_QUEUE** is to be defined as per queue name in each config defined

- Multiple app servers and multiple queues: Service split can be app-wise with different queues configured with queue manager. **INPUT_QUEUE** can be defined accordingly

Caution – each queue configured should have unique messages.

Introduction to Microservices

Product: Finacle Digital Engagement Hub Version: 11.5.x and above

Microservices are small services that work together. Each microservice runs its process and communicates in a lightweight manner. Each microservice has its own business layer and database structure. Changes made to one microservice do not affect other microservices. A monolithic **FEBA** or eBanking application has been segregated into multiple services in **DEH** and has independent microservices like Authentication, Limits, Direct Banking, and Consent Management.

Benefits of Microservices

- Each Microservice has its own functionality and can be considered as one single module and hence can be deployed independently
- Since Microservices are independent of each other, the cost of scaling is less when compared to a monolithic architecture. Upgrading of one microservice does not mean that the entire set of services also needs to be upgraded

- The addition of new services or functionalities can be encapsulated into other Microservices to not complicate the existing microservice since microservices are independently manageable
- A Microservice that is being used more frequently can be deployed on multiple servers to enhance performance since there will be a scenario where few services are being used more than the others. So, to fine-tune the performance, demanding services can be deployed on multiple servers
- Microservices are technology-independent. In **Finacle DEH**, Microservices are based on **Java**.
- Failure of a single microservice will not affect the entire application. There will be some kind of impact which means if the limits Microservice is not working properly, then transactions will not work but other modules independent of that microservice can still be accessed

Challenges of Microservices

There is a chance of failure during communication between different services. To ensure smooth communication between services, here are some proactive set ups that can be followed:

- Server clustering technology or retry the operation after a certain delay or with an exponential backoff strategy. This can be particularly useful when there are network glitches or resource unavailability
- Caching can be used at the database or microservice level for repeated requests
- Monitor microservice regularly to find latency issues and based on the findings, the database queries and code can be optimized
- Network setups can be upgraded to reduce latency

New PSP Release Notes Added to FSC!

The Product Service Pack (PSP) release notes contain Product-wise consolidated notes, Menu Mapping Sheets, and ReadMe Word documents for patches and bug fixes for select versions of Finacle. Finacle Support Center has now updated this resource with new release notes for the following versions:

Finacle Version	PSP Version
10.2.18	PSP28

Click here to visit FSC and view the artifacts.

What's New in FSC?

770+ incident resolutions have been added to the portal. Click here to visit the portal and view the artifacts.

Do you have the most useful Finacle URLs handy? Bookmark these Finacle links:

- **Finacle Support Center:** https://support.finacle.com/
- **Finacle Knowledge Center:** https://content.finacle.com
- **TechOnline:** https://interface.infosys.com/TechonlineV2/base/globallogin

Hope you like this edition. Is there anything that you'd like to see in the forthcoming series? We'd love to hear from you! Write to us at finaclesupport@edgeverve.com

Infosys® | Finacle Better Inspires Better www.finacle.com