FIS Modern Banking Platform 25 million account performance benchmark report

Modern Banking Platform 2024

Contents

Contents continued

Contents continued

1. Preface

1.1. About Modern Banking Platform (MBP)

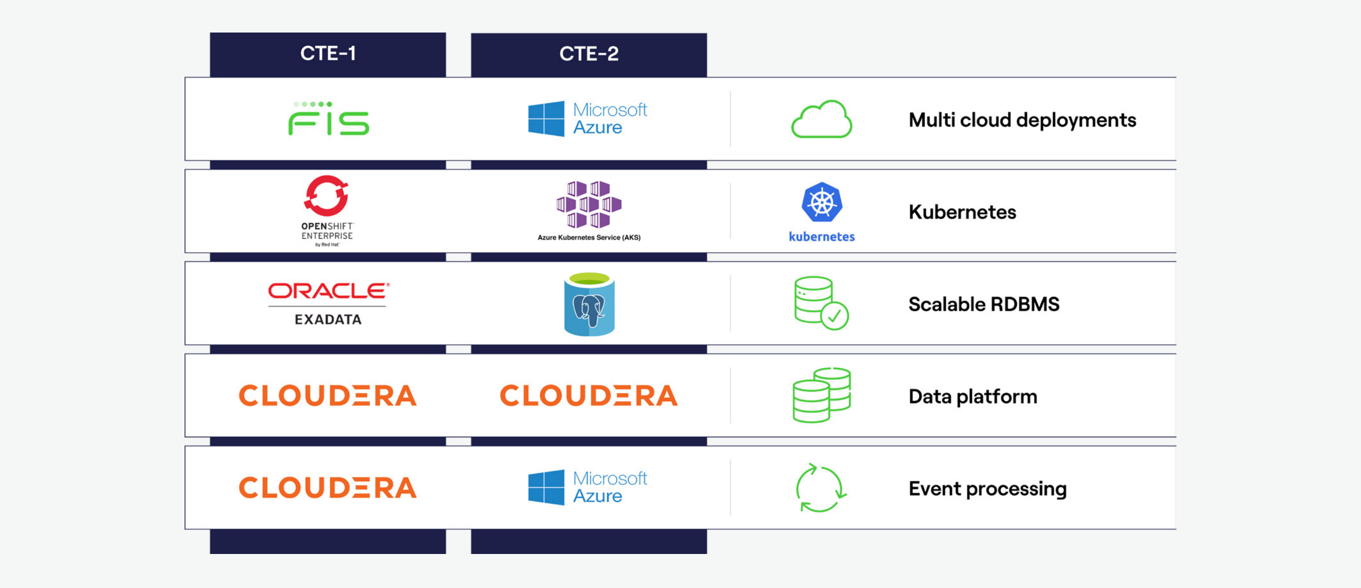
FIS has designed the Modern Banking Platform to be modular, allowing for solutions to be tailored by combining the appropriate set of application components needed to support a given solution. This benchmark specifically focused on the MBP 3.11 USA Retail Deposits Solution.

The major functional components tested in this benchmark include:

- **Retail Deposits:** This component handles the creation and transaction processing for various types of accounts such as Checking & Savings accounts, Retail Term Deposits, and Individual Retirement Accounts. Retail Deposits is built on the CAPE financial processing engine, which serves as the account processing engine of the FIS Modern Banking Platform
- **Common Arrangement Processing Engine (CAPE):** CAPE provides essential software engineering features that are common across many financial services markets, including product and pricing definition, account definition, charge processing, transaction processing, and accounting patterns that can be reused
- **Enterprise Customer:** This solution centralizes customer information, consolidating and unifying data across multiple channels to provide a comprehensive single view of customer and relationship data
- **Reporting & Analysis System (RAS):** CAPE RAS is the reporting and analysis solution for CAPE. It can consume data from various Modern Banking Platform components, including CAPE Retail Deposits, CAPE Commercial Deposits, CAPE Lending, Collateral, EFT, Enterprise Customer, and Enterprise Org.

MBP is designed to support multiple cloud deployment models, referred to as Certified Technology Environments (CTE). There are currently two supported CTEs: **CTE-1** for FIS private cloud deployments in our own data centers, and **CTE-2** for Azure public cloud deployments.

This Benchmark Report focuses on the FIS Private-Cloud Certified Technology Environment (CTE-1).



The four major technology platform's details for CTE-1 are as follows:

1. **Kubernetes – Red Hat OpenShift.** The Kubernetes platform acts as the primary application processing tier for Modern Banking Platform. Kubernetes is designed to manage containerized workloads and services. It provides a framework for running distributed systems resiliently, taking care of scaling, failover, and deployment patterns.
2. **Relational Database (RDBMS) – Oracle Database on Exadata.** The relational database is used for supporting online transactional processing for MBP.
3. **Kafka – Cloudera Data Platform for Kafka.** MBP architecture includes asynchronous event processing between solution components.
4. **Data Platform – Cloudera Data Platform for Hadoop/Spark.** The MBP Reporting and Analysis System is built on the data platform ultimately supporting data warehouse functions including report and data extract processing.

1.2. Benchmark Overview

MBP v3.11 is designed to meet the demands of banks of all sizes, including the world's largest. This benchmark focuses on the MBP v3.11 Retail Deposits Solution in the USA within the FIS private cloud technology environment at a scale of 25 million deposit accounts and 15.5 million customers.

The Benchmark Model Bank was created using an empirically derived product and transaction mix representative of a typical retail deposit processing bank served by FIS. Processing includes API processing. It also includes End of Day and End of Month processing which spans both MBP Core and MBP Reporting and Analytics System (RAS).

The primary focus of the Benchmark was to assess the performance, scalability, and stability of the applications under the defined workload while ensuring they met the pre-defined Service Level Objectives (SLOs).

1.3. Benchmark Technology Profile

Certified Technology Environment (CTE)	FIS Private Cloud Hosting (CTE-1)
Solution Release Version	Modern Banking Platform v3.11
Solution	USA Retail Deposits Solution
Component Applications	Retail Deposit; Enterprise Customer; Enterprise Organization; Reporting & Analysis System (RAS)
Test Type	Online Load Test; EOD Batch Test
Volume	25 million Deposit Accounts; 15.5 million Customers
Technology Platforms	Platforms: • Kubernetes: OpenShift 4.12 • RDBMS: Oracle 19c / Exadata X10 • Kafka: Cloudera Data Platform 7.1 • Data Management: Cloudera Data Platform 7.1

1.4. Benchmark Artifacts

Key Artifacts:

- **Performance Metrics Reports:** These reports demonstrate observed API response times and throughput during load and soak tests, as well as End of Day Periodic Processing workflow processing times
- **Infrastructure Bill of Materials:** This document outlines the necessary infrastructure to run the Model Bank with 25 million accounts.

Architectural Details:

- **Optimized Platform Configurations:** These configurations are designed to meet the Service Level Objectives (SLOs)
- **New Architectural Design Implementations:** These implementations aim to overcome technical blockers and include a roadmap document with target release dates
- **Backlog of Additional Architectural Opportunities:** This backlog is used for internal teams to outline opportunities for future scalability improvements.

1.5. Service Level Objectives

The Service Level Objectives (SLOs) for the MBP v3.11 focus on two main areas including Critical EOD Periodic Processing Workflow and API Response times.

Type	Sub-Type	Service Level Objective
Critical Periodic Processing	End-Of-Day (EOD)*	Workflow < 6 hours
	End-Of-Month (EOM)*	Workflow < 6 hours
API Response Times	2023 Request Mix	Average <1,000 milliseconds
	2024 Request Mix	Average <1,000 milliseconds

1.5.1. System API SLO

SLO: API Response Time is less than **1,000 milliseconds** average across the full transaction mix.

This time is measured from when the applicable API call arrives at FIS's point of demarcation to when it leaves back to the caller, excluding any time elapsed between the caller and FIS's point of demarcation.

1.5.2. Periodic Processing SLO

SLO: MBP EOD **Critical periodic processing** flows shall complete in **less than 6 hours after EOD** processing initiation.*

* Intraday pre-processing and non-critical post-processing occur outside this time window.

1.6. Benchmark Foreword

To properly set the context of the details provided in the full document, this foreword provides some high-level context on the scope, architecture, infrastructure, and setup of the benchmark environment.

1.6.1. Model Bank

For the benchmark to be credible, it must demonstrate that it closely represents the processing of a typical bank of this scale. The Model Bank attributes emphasized include Online API Request Types, API Volumes, Customer Profile Change Volumes, Monetary Transaction Volumes, Account to Customer Relationships Ratios, Account Types, Interest Capitalization Methods, and Transactional History.

For EOD processing, the Model Bank used the anniversary date configuration to determine when to capitalize interest. Some banks may opt to use Month End capitalization configurations which is expected to increase the level of resources required to achieve service level objectives.

1.6.2. Data Partitioning

To effectively achieve performant processing, we must apply the pattern of horizontal scaling across the platform. This includes strategies around data distribution on the database, allowing the database to process efficiently. System configuration must be established ahead of time to support proper data loading which distributes accounts to these horizontal partitions.

Horizontal partitioning improves query performance by distributing data across multiple database nodes. Each node handles a smaller portion of the data, reducing the load on any single node and speeding up query execution. By aligning the processing on the node to specific data partitions, the database can reduce the expensive and contentious task of synchronizing global cache data across nodes.

1.6.3. Data Load

The data load approach focuses on the use of APIs and EOD processing to grow the platform to 25 million accounts including 30 days of daily history. This approach allows for the organic creation of data across the platform.

Enrollment of customers and accounts is accomplished by executing daily API requests that mimic online processing. The data load suite is configured to establish the customer to account relationship ratios and product mixes defined in the Model Bank. The enrollment suite is executed in a way that incrementally adds accounts volumes resulting in distributed anniversary dates which becomes important to the Anniversary Date Capitalization processing model.

Daily Monetary Transactions workloads are designed to mimic the normal transaction processing related to the Model Bank. In a live production system, monetary transactions can come from many different sources and on various ad hoc schedules. This Model Bank attempts to best capture the average daily transaction volumes for Day End processing as part of the online API workload mix.

Transactional history for this benchmark is generated by executing EOD processing for 30 days to create a month's worth of time series data. This approach acknowledges the importance of history in representing real-world processing scenarios. Understanding that creating real-world history can be complex, the benchmark aims to provide a realistic representation by establishing 30 days of history as sufficient to meet its primary objectives.

1.6.4. Online API Profile

The MBP Benchmark team provides an ongoing analysis of production workloads to refine its standard Model Bank benchmark profile. These profiles improve and become more sophisticated over time. In the last 10 million account benchmark, the mix was reflective of the 2020 API Profile. Early in the 25 million account benchmark project, the 2023 API Profile was used as the basis. Later in the 25 million benchmark cycle, the team introduced the latest revised 2024 API Profile, which primarily increased the Request Per Second (RPS) volume with a heavy concentration of monetary transactions observed to be initiated from other ancillary sources. This benchmark report references both the 2023 and 2024 API profiles.

1.6.5. Infrastructure Kit

The 25 million account benchmark for the MBP v3.11 aimed to demonstrate performance and scale in a production-equivalent FIS private cloud data center, unlike the previous 10 million account benchmark which was run in Azure. This objective was achieved by establishing an isolated ACI network segment and leveraging standard production private cloud infrastructure necessary to support each of the required platforms. The Bill of Materials document resulting from the benchmark was based on the resources observed to be used by each of these major platforms.

1.6.6. Testing Scenarios

Each day of MBP processing includes three significant processing segments:

1. **Peak Online API Processing:** This involves APIs for inquiries, enrollments, profile updates, and monetary transactions. The benchmark included metrics for both the 2023 and 2024 API profiles.

2. **Intra-day Processing:** This processing is newly available with MBP 3.11. It enables the platform to pre-process work that can run prior to the start of the EOD workflow schedule which ultimately reduces the remaining work to be executed in the critical EOD processing window.

3. **EOD Periodic Processing:** This is the time-critical processing workflow that runs each day while also simultaneously processing reduced online API traffic. Both End of Day and End of Month metrics are provided. Note that additional non-critical processing extends beyond this window which includes activities not tied to standards SLAs.

Each scenario includes the observability of critical event processing metrics associated with scaling each workload.

1.6.7. Bill of Materials

One of the primary outputs of the benchmark is to determine the infrastructure requirements necessary to support the target workload. The summary of requirements is noted as the Bill of Materials (BOM). The BOM represents the baseline processing capacity required to achieve the benchmark results. Assumptions are documented to describe how the benchmark observations were used to arrive at these defined values. This BOM does not include any requirements associated with ancillary applications surrounding MBP.

It's important to note that the BOM does not account for platform-specific High Availability (HA) redundancy designs, which need to be considered separately. Additionally, the BOM assumes that the Control Plane for each platform, such as OpenShift, is accounted for separately.

2. Summary of Performance Results

API Profile refinements are made annually to reflect real-world observations from production traffic. For this benchmark, both 2023 and 2024 mixes were included. Traffic is bifurcated between two major components, being Retail Deposits (RD) and Enterprise Customer (EC).

2.1.1. System API Findings

Test Profile	Requests per Second (RPS)		Response Time Average		Duration	
	Target	**Result**	**Target**	**Result**	**Target**	**Result**
2023 Combined	4,000	4,069	<1,000 ms	✓ PASS	1 hour	1 hour
2023 Mix – RD	2,700	2,742	< 1,000 ms	✓ PASS	1 hour	1 hour
2023 Mix – EC	1,300	1,327	< 1,000 ms	✓ PASS	1 hour	1 hour
2024 Combined	**4,700**	**4,766**	**< 1,000 ms**	**✓ PASS**	**1 hour**	**1 hour**
2024 Mix – RD	3,100	3,136	< 1,000 ms	✓ PASS	1 hour	1 hour
2024 Mix – EC	1,600	1,630	< 1,000 ms	✓ PASS	1 hour	1 hour

Both the 2023 and 2024 API Profiles exceeded the target objectives. The System API SLO was met in both cases including processing the 2024 Mix at a combined 4,766 **API requests per second (RPS)** in less than 1,000 ms average.

✓ **3,100 RPS** for Retail Deposits; **98 ms** average response time
✓ **1,600 RPS** for Enterprise Customer; **66 ms** average response time

2.1.2. Periodic Processing Findings

End of Day workflows are scheduled to run each day of the year including special activities that can occur at End of Month. For this benchmark, metrics have been collected to allow us to recognize any possible impact on End of Month processing.

End of Month Processing Results*

Test Profile	Duration	
	Target	**Result**
MBP End of Day	Checkpoint	3 hours, 0 minutes
Solution Day End	< 6.0 hours	5 hours, 35 minutes

* EOD processing SLO was met while completing all processing in 5 hours and 35 minutes, which is less than the combined 6-hour SLO. End of Month processing was also tested, with no increase in batch duration.

EOD processing is a reference to the periodic workflow scheduled for each day to finalize and data processing that cannot otherwise be completed in real-time. It consists of a series of interdependent jobs and sub-tasks. The job sequence is divided into Critical Job Processing and Non-Critical Job Processing segments. The Critical Job Processing designation primarily relates to the series of jobs that lead to outputs feeding downstream time-critical dependencies.

The infrastructure scaling profile of these jobs is roughly correlated to the number of customers and accounts on the platform and their associated activity. As a result, infrastructure resources need to be adjusted to scale to the volumes associated with customer profile updates, monetary transactions to be processed, and other periodic processing associated with each account.

In addition, the benchmark mimics the online API activity typically running during EOD processing. The Model Bank analysis approximates this load to represent around 25% of the Peak Hour or 1,200 API Requests Per Second.

During this period, online API activity responded within the SLO and maintained its average response times in line with the other reported mid-day online testing result.

2.1.3. Benchmark Comparison – 10M vs 25M

The table below compares key API processing metrics for the 10M and 25M Benchmarks, along with select infrastructure details used in each Benchmark. Benchmark API Profiles are periodically updated based on observations from active MBP Production clients.

Consequently, API Request Profiles and the relative throughput of Retail Deposit and Enterprise Customer API requests differed between the 10M and 25M Benchmarks. For the 25M Benchmark, there was a substantial increase in target throughput for both Retail Deposits and Enterprise Customer API requests, as well as improved response times.

A notable observation in comparing the Benchmarks is the newer version of the solution demonstrated the ability to process proportionally higher throughput with less resources while also improving overall response times.

Benchmark	10M	25M	Delta
MBP Version	**3.7.1 MM5**	**3.11 SP5**	
Total API Throughput	1,600 RPS	4,766 RPS	298%
Retail Deposits Throughput	1,400 RPS	3,136 RPS	224%
Retail Deposits Average Response Time	468 ms	98 ms	478%
Enterprise Customer Throughput	200 RPS	1630 RPS	815%
Enterprise Customer Average Response Time	171 ms	66 ms	259%
Oracle Database Required*	128 vCPU	2 X 115 vCPU	150%
OCP Required**	712 Cores	1,035 Cores	154%
RAS Platform (Hadoop)***	192 Cores	448 Cores	233%
Oracle Database Type	Azure VM	Exadata X10	

* See detailed Bill of Materials specification, in section 9.1
** See detailed Bill of Materials specification, in section 9.2
*** See detailed Bill of Materials specification in section 9.3

3. Model Bank Details

The concept of a Model Bank is the definition of several characteristics of a representative banking client including environmental configuration, product setup, processing profile, and transaction mix.

The product and customer-to-account relationship distributions are detailed below.

The Model Bank transaction mix was implemented using automated Online Transaction Load Profiles, simulating peak API loads expected for a bank with approximately 25 million accounts.

Additionally, Periodic Processing was setup using a standardized workflow segmented into Critical and Non-Critical Processing. The capitalization model implemented was based on anniversary date processing configuration, allowing for uniformed processing each day of the month.

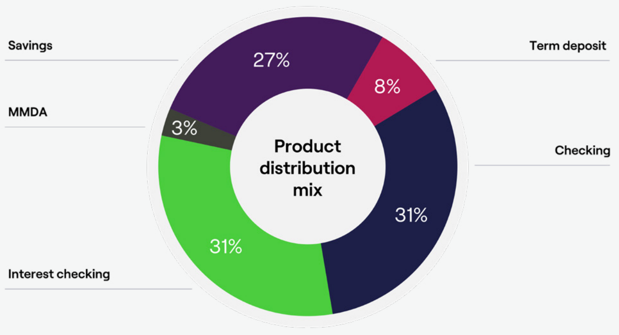
Finally, both Day End and Month End profiles were simulated to analyze any potential differences that may occur due to month end processing.

3.1. Account Type Production Distribution

The products in the Model Bank include a mixture of savings, term deposit, and checking account types, reflecting the product mix typically found in a retail bank.

Products are defined within each of these product types:

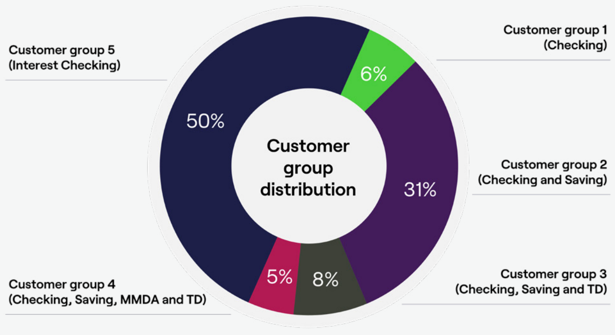
Product Type	% Distribution
Term Deposit	8%
Checking (non-interest)	31%
Interest Checking	31%
MMDA	3%
Savings	27%
Total	**100%**



3.2. Customer Group Distribution

Five distinct customer groups were defined for the Model Bank, each with a specific number and type of related accounts. The distribution ratio of these customer groups is based on an analysis of existing MBP production clients hosted in the FIS Datacenter.

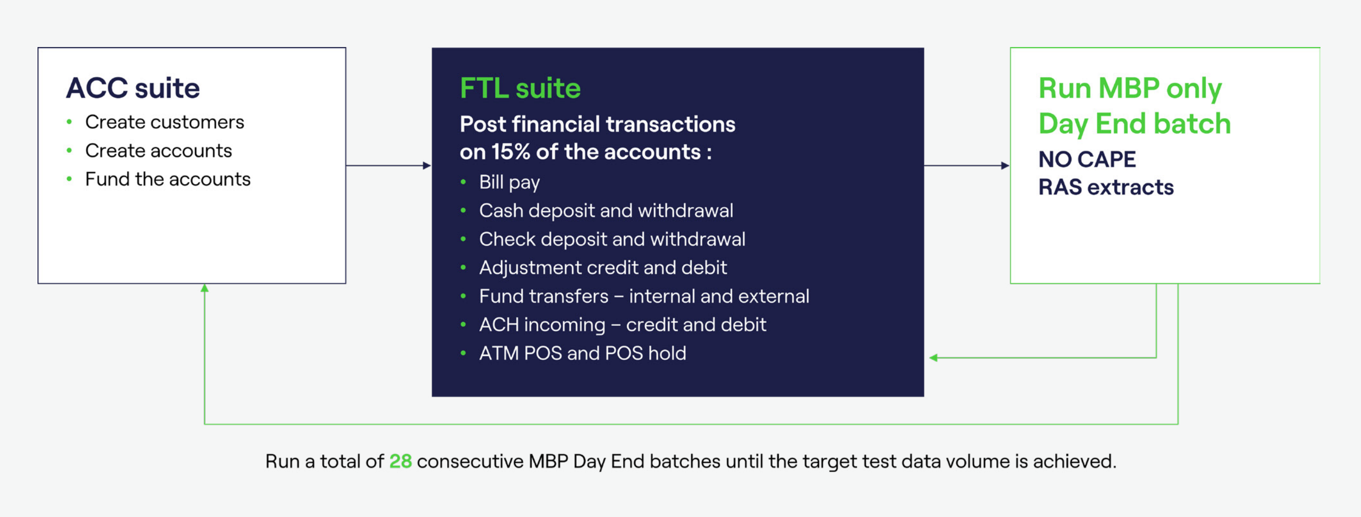
Customer Group	% Distribution
Customer Group 1 (Checking)	6%
Customer Group 2 (Checking & Saving)	31%
Customer Group 3 (Checking, Saving & TD)	8%
Customer Group 4 (Checking, Saving, MMDA & TD)	5%
Customer Group 5 (Interest Checking)	50%
Total	**100%**



3.3. Model Bank Setup - Data Seeding Steps

Model Bank data was created using scripts for Account and Customer Creation (ACC) and Financial Transaction Load (FTL) to generate account transaction history. At each processing date, the ACC script was executed for a predetermined customer and account volume. Then, the FTL script was executed to add financial transactions to 15% of all the accounts loaded to date, selected randomly. Next, EOD processing was run to update balances, accrue and capitalize, and progress the system processing date. This process was repeated until the final target volumes were reached.

The flow below shows the overview of the data seeding process:


ACC suite
- Create customers
- Create accounts
- Fund the accounts
FTL suite
Post financial transactions on 15% of the accounts :
- Bill pay
- Cash deposit and withdrawal
- Check deposit and withdrawal
- Adjustment credit and debit
- Fund transfers – internal and external
- ACH incoming – credit and debit
- ATM POS and POS hold
Run MBP only Day End batch
NO CAPE
RAS extracts
Run a total of 28 consecutive MBP Day End batches until the target test data volume is achieved.

3.4. Model Bank – Final Account and Customer Volumes

Below is the final volume of customers and accounts after the Data Seeding phase of the Benchmark, along with the number of accounts for each product type:

- Total Customers, Target: 15,500,000
- Total Customers Created: 15,790,722
- Total Accounts, Target: 25,000,000
- Total Accounts Created: 25,086,934

Product Type	Count	% of Total
Term Deposit	1,994,106	8%
Checking (non-interest)	7,823,384	31.2%
Interest Checking	7,788,705	31%
MMDA	751,172	3%
Savings	6,729,567	26.8%
Total	25,086,934	100%

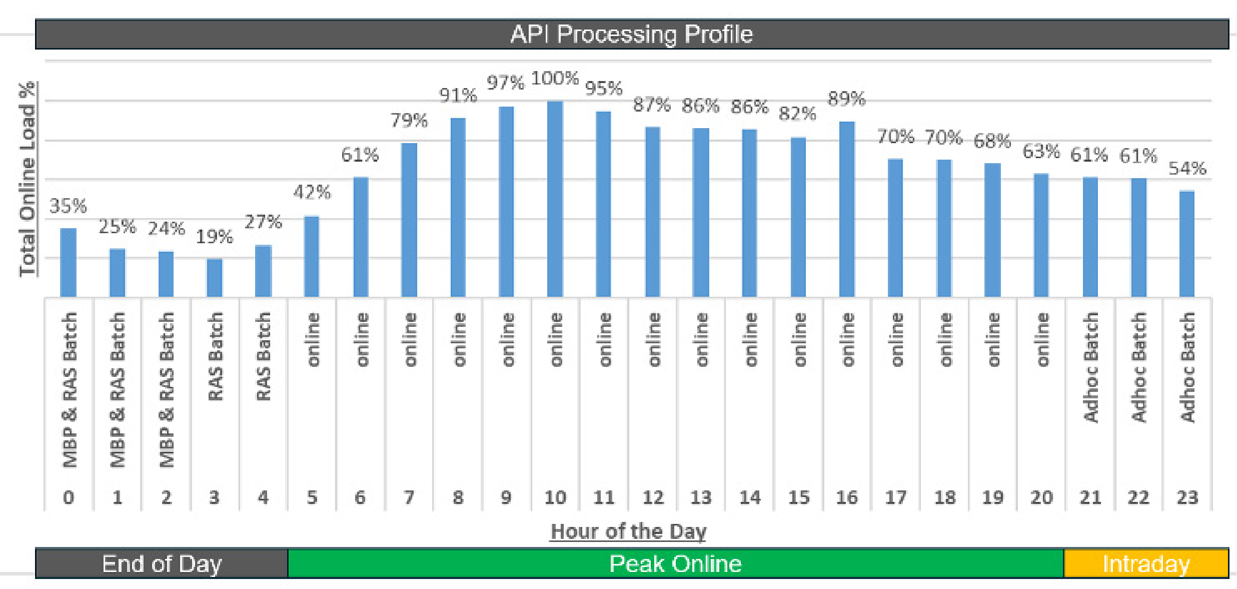
3.5. Model Bank – Periodic Processing Profile

The Model Bank is configured to perform capitalization and monthly processing, such as fees and statements, on the same date as account creation, in monthly increments. On average, capitalization and monthly processing occurs for 1/30 of accounts each day of the month.

As a result, Month End processing is expected to perform similarly to regular EOD workflows. Each EOD workflow is preceded by a Batch Transaction Load, which generates financial transactions to around 20% of all checking and savings accounts achieving Model Bank targets for each day.

Additionally, customer maintenance transactions are included. Intraday processing is run prior to the EOD processing flow under a simulated evening-time concurrent API load at a level corresponding to **2,440 RPS**, driving additional transactions on top of those created by the Batch Transaction Load.

Intraday processing runs for a period between 2 and 3 hours during our test. The EOD processing flow is executed while the system is under a simulated night-time concurrent API load corresponding to **1,220 RPS**. Please see the following diagram for Model Bank assumptions around traffic volumes as a percentage of the daily maximum and the Intraday and EOD processing windows:



3.5.1. Model Bank – Batch Transaction Load Profile

EOD Batch is preceded by a Batch Transaction Load to simulate the daily financial activity of a bank. This Batch Financial Transaction (BTL) Load applies transactions to approximately 20% of all Checking and Savings accounts, with some accounts receiving more than one transaction. In addition to this BTL load, customer maintenance API traffic is executed. Customer maintenance and financial processing events are delivered to the RAS as a predecessor to EOD processing. The full profile of the Batch Financial Transaction Load is provided below:

Batch Transaction Load Profile	% Distribution	Transaction Count
Service	100%	6,496,667
POS Purchase (Checking)	29.8%	1,935,254
ACH Incoming Debit (Checking)	19.9%	1,289,954
ACH Incoming Credit (Checking)	10.9%	709,376
POS Withdrawal (Checking)	8.9%	580,957
ACH Incoming Credit (Savings)	8.1%	526,824
ATM Withdrawal (Savings)	4.5%	291,334
Remote Deposit Check (Checking)	2.6%	170,870
Bill Pay (Checking)	2.0%	128,800
POS Hold (Checking)	2.0%	128,800
Cash Deposit (Checking)	1.0%	64,724
Create External Account (Checking)	1.0%	64,724
External Immediate Funds Transfer (Checking)	1.0%	64,724
Check Withdrawal (Checking)	0.9%	56,956
Cash Withdrawal (Checking)	0.8%	49,836
Cash Deposit (Savings)	0.7%	48,572
Create External Account (Savings)	0.7%	48,572
External Immediate Funds Transfer (Savings)	0.7%	48,572
Debit Adjustment (Checking)	0.5%	32,362
Credit Adjustment (Checking)	0.5%	32,362
External Scheduled Funds Transfer (Checking)	0.5%	32,362
Internal Immediate Funds Transfer (Checking)	0.5%	32,362
Internal Scheduled Funds Transfer (Checking)	0.5%	32,362
Internal Immediate Funds Transfer (Savings)	0.5%	31,608
Remote Deposit Check (Savings)	0.4%	29,078
Internal Scheduled Funds Transfer (Savings)	0.4%	24,232
External Scheduled Funds Transfer (Savings)	0.3%	21,072
Cash Withdrawal (Savings)	0.2%	10,536
Check Withdrawal (Savings)	0.1%	9,482

3.6. Model Bank – API Request Profile

The MBP Model Bank API Request Profile is periodically updated based on observations from active MBP Production clients and augmented with other industry sources of information. We update an internally published API Request Profile annually, which is used in various testing stages of the MBP Benchmarks. The CTE-1 25M Benchmark executed API tests for the 2023 API Request Profile and added tests for the 2024 API Request Profile after it was finalized. The Request Profiles include the following types of requests:

- **Retrieve Services:** Account and Customer Inquiry APIs
- **Search Services:** Account and Customer Search APIs
- **Record Services:** Account and Customer records update APIs
- **Monetary Services:** POS, Transfers, Deposits, Withdrawals, etc
- **Account Open Services:** Account on-boarding APIs
- **Customer Open Services:** Customer on-boarding APIs

3.6.1. 2024 and 2023 Online Request Profile – Details

Number of different requests for each of the Request Types in the 2024 Online Request Profile:

Request Type	Deposits	Customer
Retrieve Services	13	13
Search Services	4	3
Record Services	5	11
Monetary Services	13	-
Account Open Services	3	3

Filter values for select services:

Service	Balance	Charge Balance Detail	Overdraft	Ownerships	Restrictions
Account Inquiry - Filter 1	✓	✓			
Account Inquiry - Filter 2	✓	✓		✓	✓
Account Inquiry - Filter 3*	✓	✓	✓	✓	✓
Account Inquiry - Filter 4*	✓	✓	✓	✓	✓

* Account Inquiry – Filters 3 and 4 differ in the order of filter parameters in the API request.

4. Online Transaction Load and Soak Test - Results

4.1. 2023 Transaction Mix

The total target rate for the 2023 Transaction Mix was 4,000 Requests Per Second (RPS) across Deposit and Customer domains. The 2023 Transaction Mix benchmark test successfully achieved 4,069 RPS without any notable impact on target API response times.

4.1.1. 2023 Transaction Mix – Retail Deposits

- Target Rate – 2,700 RPS
- Achieved Rate – 2,741.9 RPS
- Average Response Time – 108 ms

Note that API times exclude any external centralized API Gateway that may be used for routing.

Retail Deposit Services	Throughput (RPS)	Average Response Time (ms)	Median Response Time (ms)	95% Response Time (ms)
Total	**2,741.9**	**108**		
Account Inquiry, Filter 3	774.0	159	149	238
Account Hold Inquiry	363.3	50	39	126
Transaction Search by Date	341.0	70	58	144
Retrieve Account List	230.9	46	37	118
Retrieve Account Detail	147.7	123	115	196
Account Inquiry, Filter 1	146.1	153	143	231
Monetary Transaction Search	115.5	42	33	113
Retrieve Scheduled Fund Transfers	94.0	26	21	68
Retrieve Account Periodic Balance	94.0	55	47	124
Retrieve Account Balance	83.2	108	100	179
POS Withdrawal - Checking	61.8	222	222	320
ATM Withdrawal - Savings	61.7	223	225	323
Account Inquiry, Filter 2	57.2	154	144	231
Account Inquiry, Filter 4	49.7	154	145	232
Retrieve TD Account Detail	40.3	64	55	135
Transaction Search by Reference Number	24.2	50	41	121
Periodic Balance Inquiry	18.8	108	98	181
Create External Account	9.7	146	132	215
Create Clearing Transaction	5.4	182	185	279
External Account Details Inquiry	4.3	407	389	533
Customer Accounts Balance Inquiry	2.7	70	61	143
Search Transaction Exception	2.7	35	27	92
Create Checking/Savings Transaction	2.2	188	172	293
Create Customer	1.6	254	241	331
Create Account	1.6	537	523	680
Account Pre-Closure Details Inquiry	1.6	68	58	139
Close Account	1.6	454	443	570
Create CASA Account	1.6	513	496	668
Retrieve Account Pre-Closure Detail	1.6	51	44	117
Create Closure Transaction	1.6	427	413	547
Create TD Account	0.8	559	531	733

4.1.2. 2023 Transaction Mix – Enterprise Customer

- Target Rate – 1,300 RPS
- Achieved Rate – 1,326.9 RPS
- Average Response Time – 54 ms

Note that API times exclude any centralized API Gateway that may be used for routing.

Enterprise Customer Services	Throughput (RPS)	Average Response Time (ms)	Median Response Time (ms)	95% Response Time (ms)
Total	**1,326.9**	**54**		
Customer Inquiry	790.1	36	28	92
Retrieve Customer - Account Relationship	118.5	62	48	134
Retrieve Account - Customer Relationship	92.0	78	63	148
Customer - Account Relationship Inquiry	75.6	79	65	150
Customer - Account Relationship Inquiry, Exclude Closed Account	64.3	83	66	156
Life Cycle Status Update	54.2	50	38	121
Customer - Customer Relationship Inquiry	41.6	93	78	162
Customer - Account Relationship Inquiry	26.5	92	79	159
Customer Search by Tax Id	6.9	102	87	172
Create Customer	6.9	261	247	343
Create IP	6.3	151	137	226
Retrieve Account	5.0	202	187	270
Customer Update	3.8	146	133	213
Create Contacts	3.8	186	171	258
Update Customer Preferred Details	3.8	66	53	135
Create Customer Preferred Contact	3.8	59	46	127
Retrieve Customer Tax Id	3.2	35	25	95
Retrieve Customer Preference	3.2	42	32	111
Retrieve Customer POC	3.2	70	56	136
Retrieve Customer Name	3.2	68	55	137
Search for Customer	3.2	100	84	169
Create Customer Tax Details	3.1	77	65	143
Customer to Account Relationship Update	2.5	109	93	175
Create External Account	2.5	138	127	205

4.2. 2024 Transaction Mix

The total target rate for the 2024 Transaction Mix was 4,700 Requests Per Second (RPS) across Deposits and Customer. The 2024 Transaction Mix benchmark test successfully achieved 4,766 RPS without any notable impact on API response times.

4.2.1. 2024 Transaction Mix – Retail Deposits

- Target Rate – 3,100 RPS
- Achieved Rate – 3,136.3 RPS
- Average Response Time – 98 ms
- Steady State Duration – 1 hour

Note that API times exclude any external centralized API Gateway that may be used for Enterprise routing. Alternatively, the benchmark uses the embedded internal microgateway for this purpose.

Retail Deposit Services	Throughput (RPS)	Average Response Time (ms)	Median Response Time (ms)	95% Response Time (ms)
Total	**3,136.3**	**98.0**		
Account Inquiry (Checking) - Filter 3	475.5	115	110	173
Get POS Holds (Checking)	363.3	38	34	60
Get Transactions by Posting Id (Savings)	309.5	41	37	66
Get External Accounts (Savings)	220.1	283	256	408
Account Transaction Search by Date (Checking)	190.2	52	46	105
Get External Fund Transfers (Savings)	182.1	34	31	55
Get Internal Fund Transfers (Savings)	182.1	29	26	46
POS Purchase (Checking)	144.5	161	146	246
Account Transaction Search (Savings)	141.2	50	45	104
POS Withdrawal (Checking)	116.5	158	142	244
ACH Incoming-Debit (Checking)	96.0	157	140	240
Account Inquiry (Checking) - Filter 2	57.5	91	85	143
Account Inquiry (Checking) - Filter 1	57.5	107	101	165
Account Inquiry (Checking) - Filter 4	57.5	95	89	147
Account Periodic Balance Inquiry (Savings)	55.6	99	94	155
ACH Incoming Credit (Checking)	52.8	160	144	243
Account Inquiry (Term Deposit)	46.1	81	76	135
Account Transaction Search by Reference Number (Checking)	45.2	46	41	80
Account Inquiry (Savings) - Filter 2	42.1	121	117	177
Account Inquiry (Savings) - Filter 1	42.1	140	135	199
Account Inquiry (Savings) - Filter 3	42.1	126	121	182
Account Inquiry (Savings) - Filter 4	42.1	123	118	179
Account Balance Inquiry (Savings)	35.7	82	76	137
Transaction Exception Search (Savings)	28.1	51	47	100
Account Balance Inquiry (Checking)	18.5	84	78	140
Remote Deposit Check (Checking)	12.8	162	149	247
ACH Incoming Credit (Savings)	12.8	214	214	297

Retail Deposit Services	Throughput (RPS)	Average Response Time (ms)	Median Response Time (ms)	95% Response Time (ms)
Get POS Holds (Savings)	12.4	36	33	56
Bill Pay (Checking)	9.5	160	144	244
ATM Withdrawal (Savings)	7.1	195	199	289
External Immediate Funds Transfer (Checking)	4.8	157	141	241
Check Withdrawal (Checking)	4.3	126	120	180
Cash Withdrawal (Checking)	3.8	160	143	242
Credit Adjustment (Checking)	2.4	120	114	177
Debit Adjustment (Checking)	2.4	121	115	177
Internal Immediate Funds Transfer (Checking)	2.4	266	265	354
Create External Account (Checking)	1.9	252	233	333
Get Pre-Closure Details (Checking)	1.4	57	52	107
Create Account (Checking)	1.4	530	501	712
Delete Account (Checking)	1.4	304	285	389
Update External Accounts (Checking)	1.0	111	97	174
Update Account Basic Details (Savings)	1.0	109	115	179
Cash Deposit (Savings)	1.0	207	208	287
Create External Scheduled Fund Transfer (Savings)	1.0	116	121	192
Delete External Scheduled Transfer (Savings)	1.0	49	44	101
Update Account Business Details (Checking)	0.6	121	101	205
Create Additional Accounts (Checking)	0.5	48	44	90
Cash Deposit (Checking)	0.5	156	140	236
Create Related Accounts (Checking)	0.5	71	53	126
Get Related Accounts (Checking)	0.5	54	41	106
Transaction Exception Search (Checking)	0.5	46	42	81
Cash Withdrawal (Savings)	0.5	132	126	193
Create External Account (Savings)	0.5	249	229	341
Create Account (Savings)	0.5	533	506	707
Create Account (Term Deposit)	0.5	532	472	937
External Immediate Funds Transfer (Savings)	0.5	133	127	190
Internal Immediate Fund Transfers (Savings)	0.5	214	199	305
Remote Check Deposit (Savings)	0.5	218	220	299

4.2.2. 2024 Transaction Mix – Enterprise Customer

- Target Rate – 1,600 RPS
- Achieved Rate – 1,630.0 RPS
- Average Response Time – 65.7 ms
- Steady State Duration – 1 hour

Note that API times exclude any centralized API Gateway that may be used for routing.

Enterprise Customer Services	Throughput (RPS)	Average Response Time (ms)	Median Response Time (ms)	95% Response Time (ms)
Total	**1,630**	**65.7**		
Customer-Account Relationship Inquiry (Savings)	497.7	74	57	134
Account-Customer Relationship Inquiry (Savings)	387.5	89	70	148
Customer Inquiry (Checking)	203.0	33	30	49
Customer Inquiry (Savings)	108.4	31	28	46
Get Organization Settings	94.6	32	21	57
Customer-Account Relationship Inquiry (Checking)	142.7	75	58	136
Get Customer Summary (Checking)	79.9	53	39	113
Customer Names Inquiry (Savings)	27.1	51	39	108
Update Customer Life Cycle Status (Savings)	20.4	56	42	113
Get Customer Notes (Checking)	14.3	95	76	157
Get Customer Contacts (Checking)	10.0	63	49	119
Customer-Customer Relationship Inquiry (Checking)	10.0	87	67	147
Get Customer Due Diligence (Checking)	5.7	55	41	113
Customer Search by Name (Savings)	4.8	54	40	112
Customer Search by Tax Id (Savings)	4.8	47	34	97
Customer Search by Name (Checking)	2.4	56	42	117
Customer Search by Tax Id (Checking)	2.4	47	34	99
Create Customer (Checking)	1.4	182	157	254
Create Customer Notes (Checking)	1.4	60	45	119
Create Customer Contact Preferences (Checking)	1.4	62	46	116
Create Customer Contacts (Checking)	1.4	159	138	233
Create Customer Tax Information (Checking)	1.4	85	67	144
Update Customer Preferences (Checking)	1.4	52	40	106
Search for Account (Savings)	1.4	37	26	77
Update Customer (Checking)	0.5	148	128	208
Update Customer Life Cycle Status (Checking)	0.5	54	39	109

Enterprise Customer Services	Throughput (RPS)	Average Response Time (ms)	Median Response Time (ms)	95% Response Time (ms)
Create Customer (Savings)	0.5	171	152	243
Create Customer Notes (Savings)	0.5	56	41	116
Create Customer Contact Preferences (Savings)	0.5	58	46	112
Create Customer Contacts (Savings)	0.5	164	139	237
Create Customer Tax Information (Savings)	0.5	85	67	143
Update Customer (Savings)	0.5	153	125	213
Update Customer Preferences (Savings)	0.5	54	40	101

4.3. Soak Test

- The Soak Test involved executing the 2023 Traffic Mix at a steady 70% of maximum volume for 8 hours to identify the impact on system components due to long-running production-like workload timeframes
- The transaction mix for the Soak Test was based on the 2023 Transaction Mix Profile described previously, with a sustained throughput of 2,840 RPS
- This transaction load was executed in a steady state for 8 hours, with an average response time of 73.5 ms
- CPU and memory usage across deposit and customer service applications remained steady throughout the test
- Graphs of CPU and memory usage are presented in section 7.1.5.

4.4. Concurrent Online Transaction Load

As described in section 3.5, an online transaction load using the **2023 transaction mix at 30% of peak throughput** was executed during Periodic Processing tests.

Results for the concurrent online transaction load are taken from the Month End Periodic Processing test. They are shown below, with a comparison to the peak online transaction load results. SLOs for API response time were met, with all APIs achieving average response times of less than 1,000 ms.

Concurrent Online Transaction Load – Target and Achieved Throughput:

Retail Deposits			Enterprise Customer		
Target Throughput (RPS)	**Achieved Throughput (RPS)**	**Average Response Time (ms)**	**Target Throughput (RPS)**	**Achieved Throughput (RPS)**	**Average Response Time (ms)**
810	827	81	390	400	37

Concurrent Online Transaction Load – Comparison to Peak Online Load:

	Retail Deposits		Enterprise Customer	
	Throughput (RPS)	**Average Response Time (ms)**	**Target Throughput (RPS)**	**Average Response Time (ms)**
Peak Online Load Test	2,742	108	1,327	54
Concurrent Online Load Test (30% Load)	827	81	400	37

5. Periodic Processing Schedule – Overview and Results

The 25M Benchmark included separate Periodic Processing cycles for Day End (having Processing Date of one day prior to the end of the month) and Month End (having Processing Date of the end of the month).

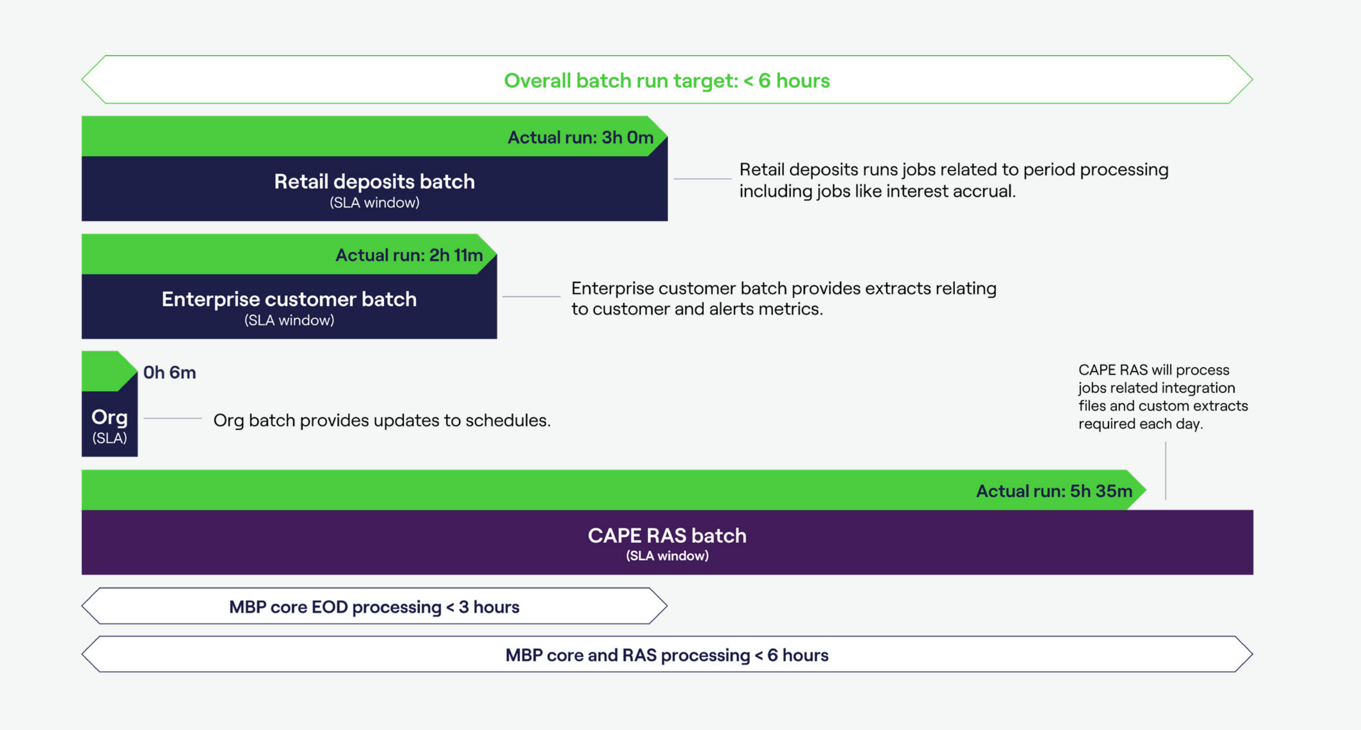
As job schedules and the number of accounts capitalizing on the End of Day and Month End tests were identical, and to avoid redundancy, Periodic Processing results included in this report are from the Month End Periodic Processing cycle.

5.1. Total Run Time and SLOs

Critical EOD processing completes in less than the 6-hour service level objective (SLO) while handling aligned night-time API traffic of 1,200 Requests Per Second (RPS).

- EOD service level objectives incorporate all critical processes from the start of MBP to the final critical-path extracts completing in CAPE RAS
- SLOs are consistently maintained via horizontal scaling across each platform. This allows us to meet objectives consistently from smaller workloads up to 25 million accounts
- MBP Core and RAS EOD processing SLO is 6 hours
- The EOD processing time achieved in this benchmark test was 5 hours 35 minutes.

Note: RAS Processing jobs generally process after specific Customer and Deposits dependencies.



Retail deposits runs jobs related to period processing including jobs like interest accrual.

Enterprise customer batch provides extracts relating to customer and alerts metrics.

Org batch provides updates to schedules.

CAPE RAS will process jobs related integration files and custom extracts required each day.

6. Scalability – 5M vs 25M Performance

During the data load up to 25M volume, we paused at 5M volume to perform an initial performance assessment. This provided the basis for a scalability and performance comparison at different account and transaction volumes.

6.1. Online Transaction Load - Scalability Assessment

The scalability comparison for Online Transactions is based on the 2023 traffic mix. The results of the 25M Benchmark compared favorably with the 5M checkpoint, with similar response times observed. The 5-fold increase in both account volume and API throughput was successfully compensated for with a linear corresponding increase in database and OpenShift pod capacity.

Benchmark	MBP Release	Account Volume	Customer Volume	DB Size
5M	3.11 RC12	5,000,000	5,000,000	CPU count: 96 SGA + PGA: 64 GB
25M	3.11 GA	25,000,000	15,500,000	CPU count: 230 SGA + PGA: 325 GB

Benchmark	EC Throughput (RPS)	EC Average Response Time (ms)	RD Throughput (RPS)	RD Average Response Time (ms)
5M	263.4	59	551.4	106
25M	1,326.9	54	2,741.9	108

6.2. Periodic Processing – Scalability Assessment

For large jobs in Periodic Processing, the workload was partitioned across multiple pods, each representing a Job Partition. Additionally, for the 25M tests, specific Retail Deposit Batches were 'pinned' to one of 58 database partitions on alternating RAC nodes, as described in section 8.6.2. While the 25M test completed comfortably within the 6-hour SLO, periodic processing at 25M had some notable differences as compared to 5M, which is a result of unproportional batch pod counts for each test.

It is necessary to consider the non-equivalent ratio of job partitions to total accounts used for 5M and 25M. 14 Retail Deposit Batch pods were used for the 5M periodic processing execution, but only 58 pods rather than 70 (5x14) were used for 25M to align exactly with the number of database partitions needed for DB node pinning. The use of proportionally fewer Batch pods for 25M (83% of 70) explains almost exactly the entire increase in MBP-only run times: 180 minutes for 25M vs. 153 minutes for 5M (85% of 180 min).

On the CAPE RAS side, job schedule changes, including event reconciliation jobs, were integrated between running the 5M and 25M Benchmark tests, preventing a full side-by-side comparison. The method of measuring and summing run times was adjusted to be more realistic for the full 25M test, for example by not excluding brief execution pauses between jobs introduced by the Batch scheduler.

Benchmark	MBP Release	Account Volume	Customer Volume	DB Size
5M	3.11 RC12	5,000,000	3,100,000	CPU count: 96 SGA + PGA: 64 GB
25M	3.11 GA	25,000,000	15,500,000	CPU count: 230 SGA + PGA: 325 GB

Benchmark	Run Time (MBP Only)	Run Time (MBP and RAS Critical Path)
5M	2 hours, 33 minutes	4 hours, 27 minutes
25M	3 hours, 0 minutes	5 hours, 35 minutes

7. Platform Metrics

The metrics in this section contribute to the Bill of Materials calculations in section 9. Please review that section for further details. Analysis of each testing profile seeks to determine the peak processing resources required to successfully process the target workloads. The results of this analysis are then provided in the final Bill of Materials which acts as the requirements for infrastructure procurement.

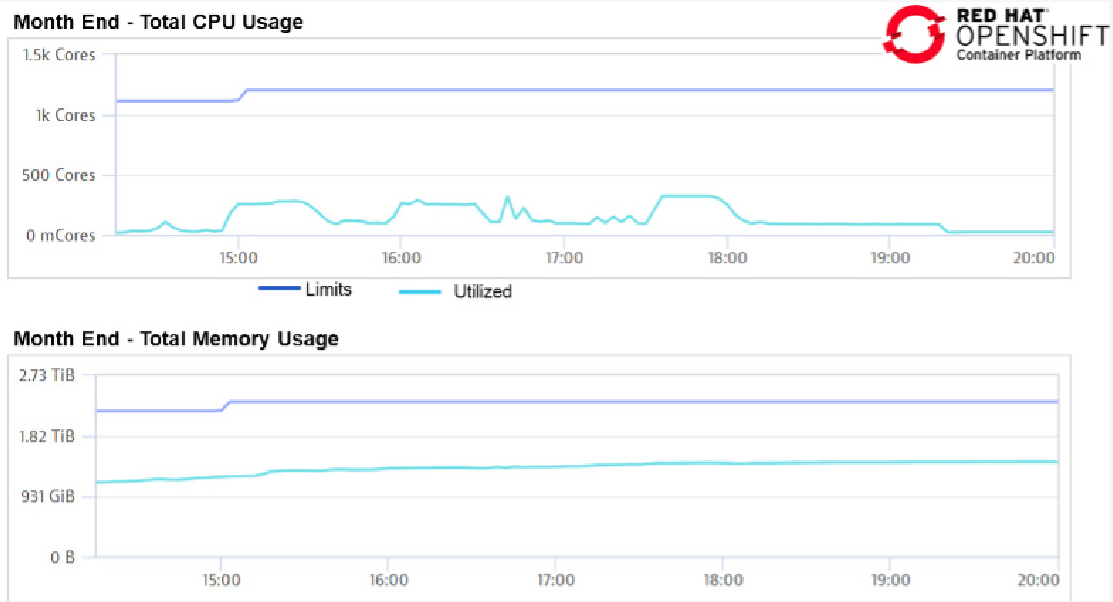
7.1. Kubernetes Metrics

Kubernetes Platform Type: Red Hat OpenShift 4.12

7.1.1. Month End – Key Container Usage

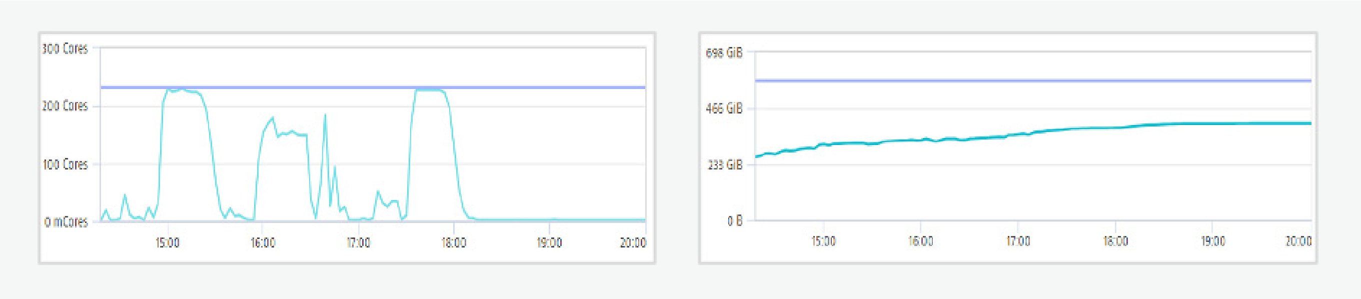
At the overall namespace level, there is sufficient spare CPU capacity, but different applications display peak usage at different times throughout the periodic processing window. Retail Deposits Batch fully utilizes its CPU capacity during various high-volume Batch job executions.

In terms of memory, RAM utilization of Batch pods continues to grow, but stays below the allocated limit. Batch pods are spun down after completion of the periodic processing window.



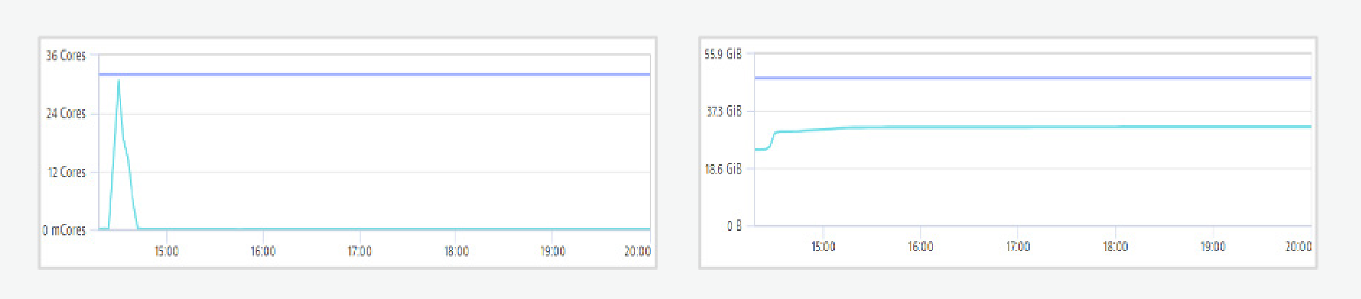
Spikes in Deposit Batch App CPU usage is noticed during various high-volume Batch job executions. At these times, Deposit Batch App CPU usage reaches 100% of allocated capacity. Some memory leakage was noticed in the Deposit Batch App, but memory usage stabilized before the end of the test.

Deposit Batch App – CPU/Memory Usage



Most EC batch processes occur within the first hour after periodic processing begins. After completion of these EC processes, Customer Batch App CPU usage is near zero. Memory usage was stable throughout the test.

Customer Batch App – CPU/Memory Usage



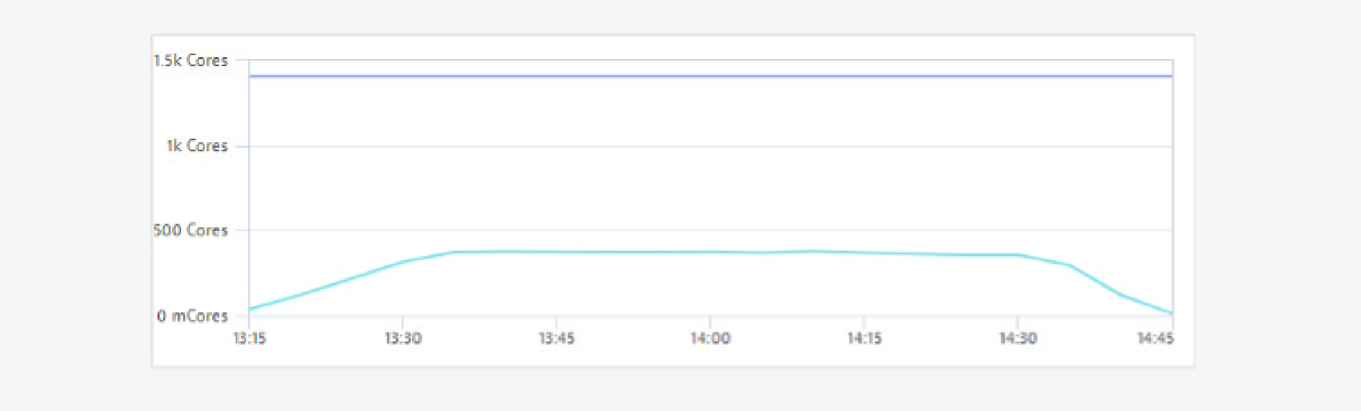
7.1.2. Online Transaction Load – Key Container Usage

MBP is deployed as a solution where each component application implements several container applications. The observations below focus on the most relevant container applications.

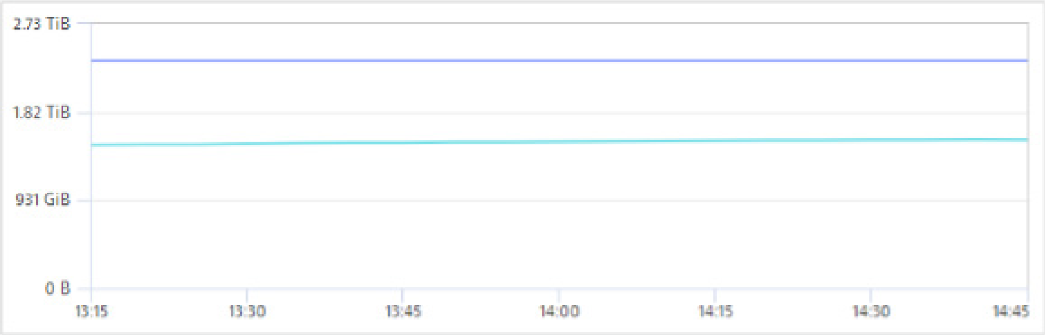
Online Transaction Load – Total CPU Usage

For the 2024 Traffic Mix peak online execution window, less than 50% of the available CPU resources were utilized. There are likely opportunities for further resource optimization at the application level, but experimental reductions in available pod counts did result in marginally slower response times.

Memory utilization remained consistent and stable during peak online traffic runs as well as the 8-hour soak test, without notable growth.



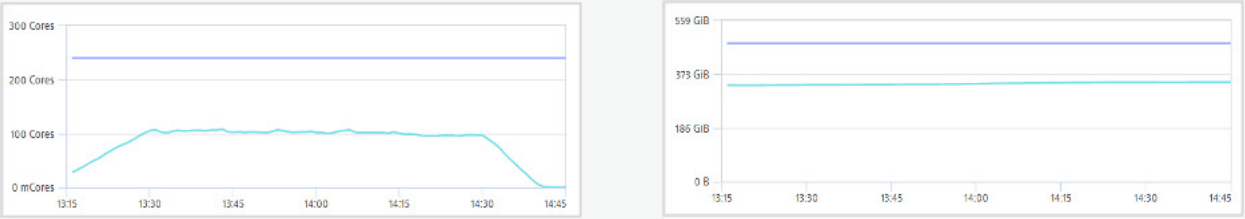
Online Transaction Load – Total Memory Usage



The deployment architecture includes individual MBP API micro-service containers which delegate down to a common Deposit Services App. The graphs below focus on the aggregate load processed at this level. In doing so, it is utilized by every Retail Deposit API during Online Transaction Load testing.

Throughout testing, CPU usage remained steady at below 50% of allocated capacity. Memory usage was also stable.

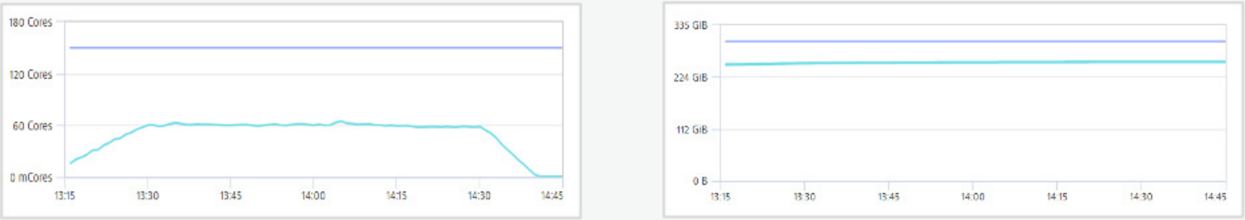
Deposit Service App – CPU/Memory Usage



The deployment architecture includes individual MBP API micro-service containers which delegate down to a common Customer Services App. The graphs below focus on the aggregate load processed at this level.

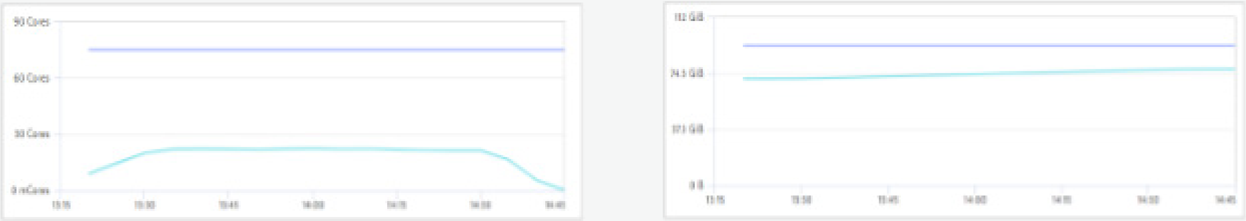
Throughout testing, CPU usage remained steady at below 50% of allocated capacity. Memory usage was also stable.

Customer Service App – CPU/Memory Usage



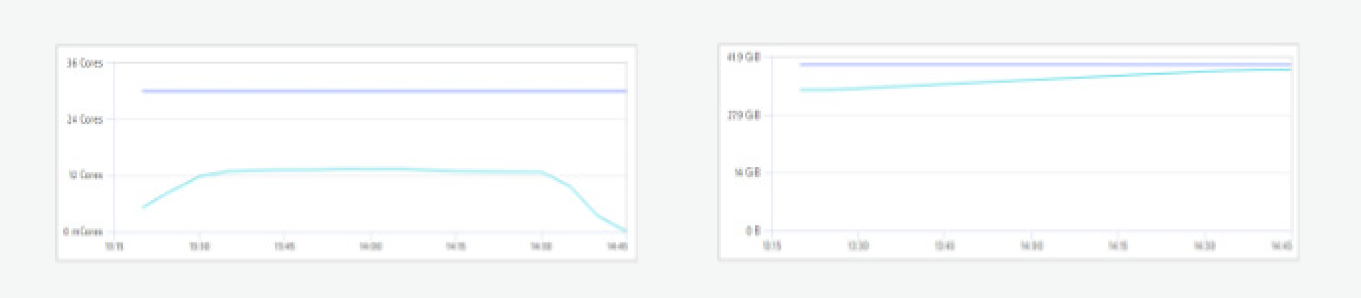
One of the highest volume MBP APIs highlighted here is the mbp-api-casa services which supports CASA Account Inquiry, Update Account Business/Basic Details, and various monetary transaction services.

mbp-api-casa - CPU/Memory Usage



APIs that used the mbp-api-rel application include Customer-Customer Relationship Inquiry, Account-Customer Relationship Inquiry, and Create Related Account.

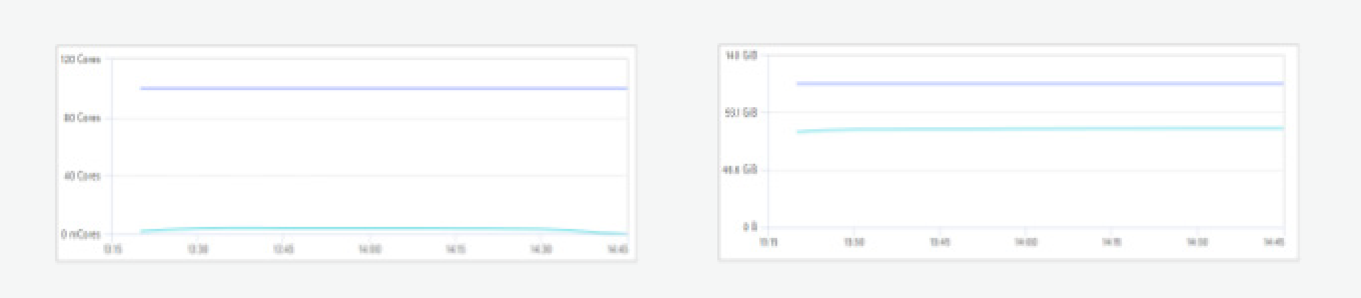
mbp-api-rel - CPU/Memory Usage



The last highlighted high-volume service is the Turbo Customer API which was instrumental in achieving EC API response times well within SLAs (see section 4 for full results). For more details on the Turbo Customer API, see section 8.5.

APIs that used the mbp-api-customer (turbo) application include Customer Inquiry, Update Customer, and Create Customer.

mbp-api-customer (turbo) - CPU/Memory Usage

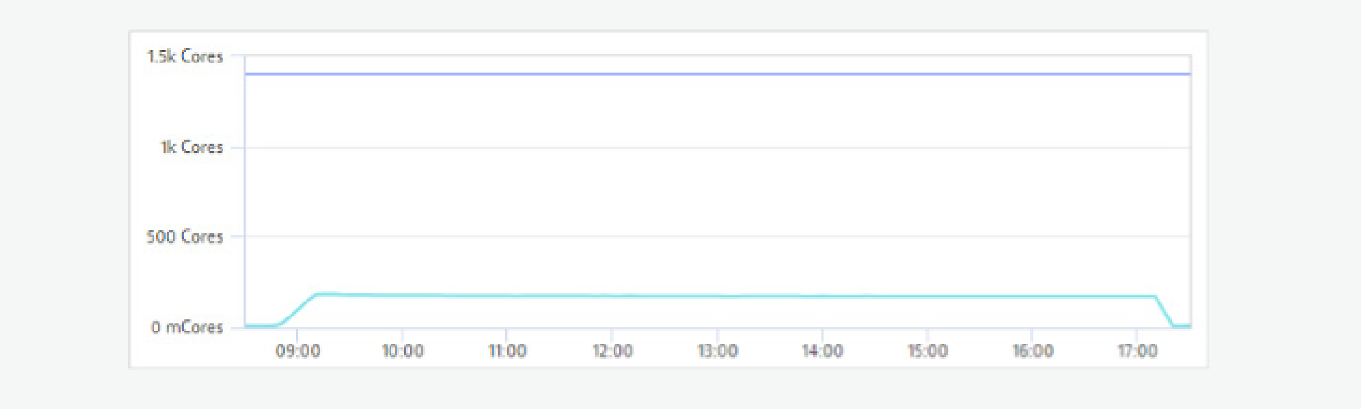


7.1.3. Soak Test – Key Container Usage

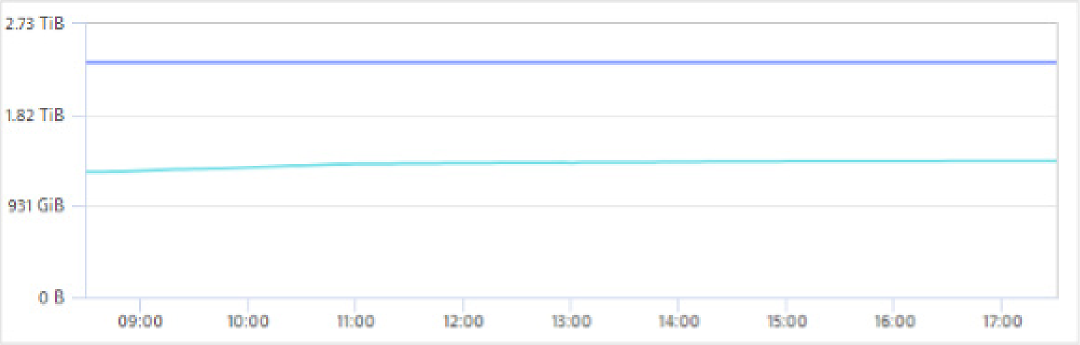
The Soak Test involved running the 2023 transaction mix profile, at 70% of max steady-state throughput (2,840 RPS), for 8 hours.

CPU usage was steady after the warm-up phase and throughout the test period. Memory utilization remained consistent and stable during the 8-hour test, without notable growth.

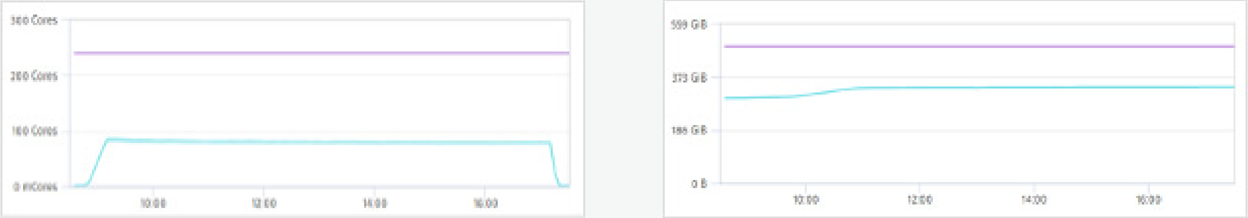
Online Transaction Load – Total CPU Usage



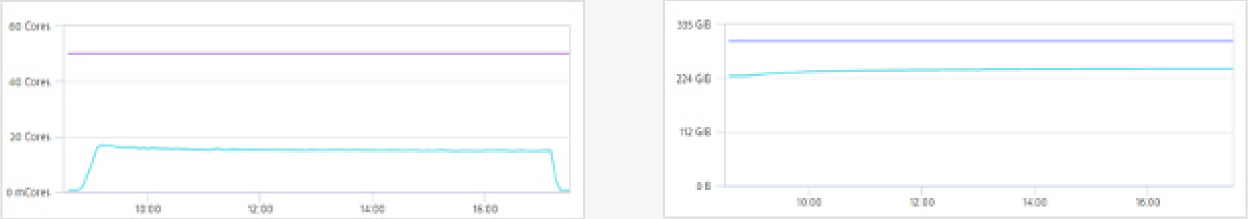
Online Transaction Load – Total Memory Usage



Deposit Service App – CPU/Memory Usage



Customer Service App – CPU/Memory Usagea



7.2. Oracle Database Metrics

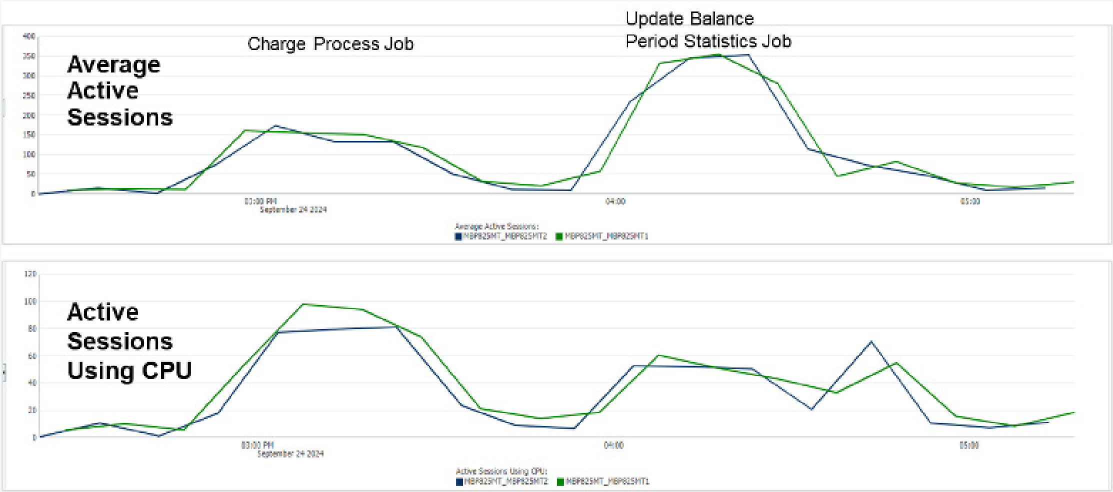
Platform: Oracle Exadata X10; Oracle Database 19c

The peak workload on the database during EOD processing occurred during the "Update Balance Period Statistics" job. Note that EOD processing is designed to pin workloads to specific nodes based on account distribution across partitions. It is recommended to have a 30% buffer beyond the utilized CPU during the benchmark.

Application	Usage (Node 1)	Usage (Node 2)
Version	Oracle 19c	Oracle 19c
RAC	Yes	Yes
Archive	Yes	Yes
Server Type	Exadata X10	Exadata X10
vCPU	98	81
Memory	1,068 GB	1,022 GB
SGA	240	240
PGA	31	29
IO (Req/sec)	373,949	321,202
IO (MB/sec)	5,052	4,175

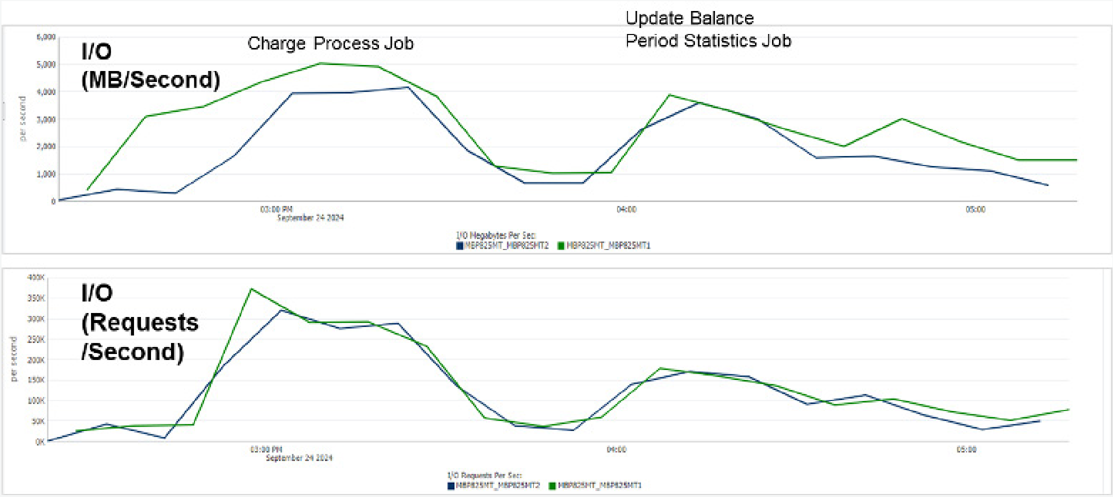
7.2.1. CPU Graphs

Active Sessions using CPU is the metric used to calculate CPU demand. 98 vCPU is consumed on Node 1 and 81 vCPU on Node 2.



7.2.2. I/O Graphs

Disk utilization is measured at 374K IOPS and 5K MB/Sec.



7.2.3. Database Performance

Test Type/Instance		I/O Requests/ Second	I/O MB/ Second	Average Active Sessions	Active Sessions using CPU	Max PGA in Use	Max Processes	Max Sessions	Total REDO Log Switches
2023 API Mix	**Node 1**	62,357	3,112	25.7		31	3,008	2,543	15
	Node 2	62,188	2,984	22.0		29	2,992	2,511	
2024 API Mix	**Node 1**	94,458	1,265	37.6	23.0	37	3,844	3,352	13
	Node 2	97,455	1,148	32.0	20.7	38	4,158	3,672	
Day End	**Node 1**	365,943	4,482	263.5	100.9	31	3,354	3,082	106
	Node 2	352,170	4,307	258.3	81.6	31	3,341	3,058	
Month End	**Node 1**	373,949	5,052	355.8	98.2	31	3,336	3,290	91
	Node 2	321,202	4,175	353.8	81.2	29	3,332	3,308	

7.2.4. Key Database Stats

Test Type/Instance		Execute count (Total)	Execute count/ Second	Redo Size (bytes Total)	Redo Size (bytes/Second)	User Commits (Total)	User Commits / Second
2023 API Mix	**Node 1**	407,680,383	74,649.17	104,442,865,036	19,124,229.01	2,676,780	490.14
	Node 2	441,403,950	80,824.13	105,031,304,736	19,231,962.41	2,223,250	407.09
2024 API Mix	**Node 1**	297,553,734	83,627.31	192,898,252,536	54,213,943.41	2,485,468	698.54
	Node 2	414,029,215	116,362.71	121,491,104,928	34,145,015.06	1,581,594	444.51
Day End	**Node 1**	1,669,585,639	103,044.22	1,314,800,074,180	81,147,409.63	7,733,365	477.29
	Node 2	1,580,135,358	97,523.47	1,044,353,000,608	64,455,826.18	5,865,498	362.01
Month End	**Node 1**	1,332,081,487	122,828.41	1,270,493,548,552	117,149,517.71	5,516,982	508.71
	Node 2	1,378,633,487	127,248.81	1,009,112,198,940	93,141,745.32	4,578,211	422.57

7.2.5. Database Waits – Online

In the context of the Modern Banking Platform (MBP) performance benchmarking, database waits are categorized and measured to understand their impact on overall system performance, for instance, wait events such as "db cpu," "user i/o" and "commit". These events are ordered by wait time and are critical for identifying performance bottlenecks and optimizing the database.

Understanding and analyzing wait events is essential for database administrators to ensure that the database performs efficiently and meets the required service level objectives. Addressing the causes of significant wait events, allows improvements in the overall performance and responsiveness of the database system.

In this table, we focus on the wait events that occurred during the Peak Online Load Test.

Test Type/Instance		Wait Class	Waits	Total Wait Time (Seconds)	Average Wait Time (µS)	DB Time Percentage	Average Active Sessions
2023 Transaction Mix	**Node 1**	DB CPU		88,831		73.5	16.3
	Node 1	User I/O	247,102,736	30,416	123.09	25.2	5.6
	Node 1	Cluster	118,425,431	7,513	63.44	6.2	1.4
	Node 1	Commit	2,283,436	667	292.21	0.6	0.1
	Node 2	DB CPU		74,440		71.8	13.6
	Node 2	User I/O	237,631,300	30,009	126.28	28.9	5.5
	Node 2	Cluster	104,765,173	6,066	57.9	5.9	1.1
	Node 2	Commit	1,704,557	479	281.04	0.5	0.1
2024 Transaction Mix	**Node 1**	DB CPU		82,749		65.1	23.3
	Node 1	User I/O	178,572,927	43,128	241.51	33.9	12.1
	Node 1	Cluster	161,520,396	10,891	67.43	8.6	3.1
	Node 1	Commit	1,665,103	748	448.98	0.6	0.2
	Node 2	DB CPU		72,868		65.2	20.5
	Node 2	User I/O	176,144,189	41,019	232.87	36.7	11.5
	Node 2	Cluster	146,867,997	8,415	57.29	7.5	2.4
	Node 2	Commit	781,083	319	408.93	0.3	0.1

7.2.6. Database Waits – Periodic Processing

In this table, we focus on the wait events that occurred during the End of Month Period Processing test.

Test Type/Instance		Wait Class	Waits	Total Wait Time (Seconds)	Average Wait Time (μS)	DB Time Percentage	Average Active Sessions
Day End	**Node 1**	DB CPU		458,821		45.4	28.3
	Node 1	User I/O	1,037,276,899	243,902	235.14	24.1	15.1
	Node 1	Cluster	481,073,870	240,722	500.39	23.8	14.9
	Node 1	Commit	6,400,588	4,052	633.08	0.4	0.3
	Node 2	DB CPU		365,419		41.7	22.6
	Node 2	User I/O	841,300,602	202,937	241.22	23.1	12.5
	Node 2	Cluster	453,580,746	246,478	543.4	28.1	15.2
	Node 2	Commit	4,421,283	2,370	536.06	0.3	0.1
Month End	**Node 1**	DB CPU		414,305		36	38.2
	Node 1	User I/O	916,891,454	221,311	241.4	19.2	20.4
	Node 1	Cluster	534,848,029	387,472	724.5	33.7	35.7
	Node 1	Commit	4,371,140	4,399	1,006.50	0.4	0.4
	Node 2	DB CPU	551,841,246	412,259	747.1	37.6	38.1
	Node 2	User I/O	741,209,636	190,596	257.1	17.4	17.6
	Node 2	Cluster		360,982		32.9	33.3
	Node 2	Commit	3,311,125	2,748	830	0.3	0.3

7.3. Events and Kafka

Kafka Platform: Cloudera Data Platform 7.1 (Kafka)

7.3.1. Events – Online Event Throughput Peak and Average

Approximately 10 million events are published during the peak online traffic hour, represented by the steady-state 2024 Traffic Mix running at 4,700 Requests Per Second (RPS). This event volume equates to 2,778 Events Per Second (EPS).

According to the Model Bank Processing Profile (see Section 3.5), Periodic Processing runs for around 5.5 hours. During the remaining 18.5 hours, the average API throughput is around 80% of the peak throughput described above. As a result, the average event volume is 2,222 EPS, or 8 million events per hour for the 18.5 hours outside the Batch window.

7.3.2. Events – Periodic Processing Event Throughput

- During the Charge Process Batch job, Event throughput peaked at approximately 18,000 Events Per Second, including events generated from the concurrently running 30% online load
- Approximately 44 million events were processed during the 5hr 35min periodic processing window
- Event volume includes core events produced and n-1 version backward compatible events.

7.3.3. Total Daily Event Volume

Processing Phase	Duration	Total Events Processed
Periodic Processing	5.5 hours	44 million
Online Processing	18.5 hours	18.5*8 million = 148 million
Total Events	24 Hours	192 million

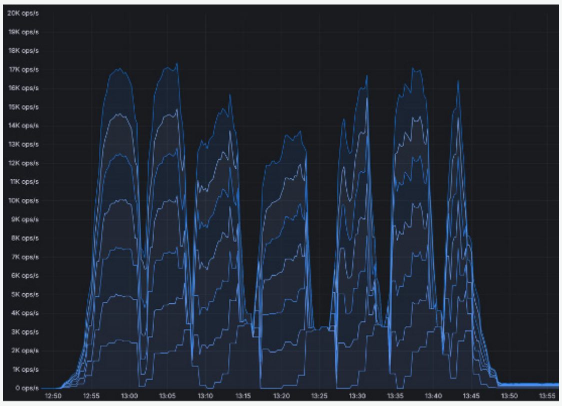
7.3.4. Event Volume Storage and Transmission Metrics

The average event size is approximately 8 kilobytes uncompressed, so the total daily uncompressed event volume of 192M events represents 1,536,000,000 kilobytes or 1,536 Gigabytes of data.

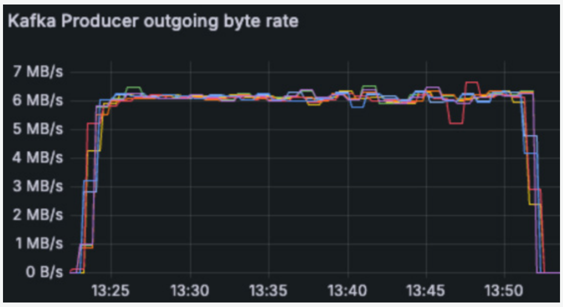
However, event payloads are stored and transmitted in compressed format:

- For event compression in the Oracle DB CDC event tables, the gzip algorithm is used, achieving a compression ratio of between 80% and 85%, depending on event type. A typical day's worth of events would require around 275 GB of daily storage in the CDC Outbox Archive table, and corresponding size in the CDC Inbox table of the consumers
- For event compression prior to transmission to Kafka and storage on the Kafka side, the lz4 algorithm is used, achieving a compression ratio of around 75%. Daily event volume using this compression results in a daily event load size of 320 GB
- Assuming a 3-day retention window in Kafka with some extra padding, 1,200 GB of storage is required

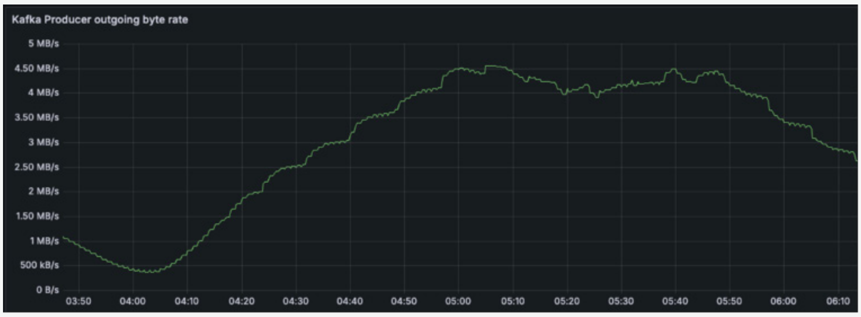
Peak outgoing rate at 18,000 EPS observed including backward compatible transformed events. Events are configured to process in 5-minute partition intervals. The partition changeover is observed with a slight delay during the transition, which will be improved in future releases.



Peak outgoing rate at 18,000 EPS observed in a separate test with an event size of 10KB and 7 CDC engines results in a network load to Kafka of 7x6 MB/s = 42,000 KB/s.



For a time-window in which we processed 2,500 EPS at 8 KB per event, the smoothed Kafka producer Outgoing Byte Rate peaked at around 4.5 KB/s, vs an expected 5 KB/s based on the calculation above (2,500 EPS x 8KB x 0.25 volume post-compression).

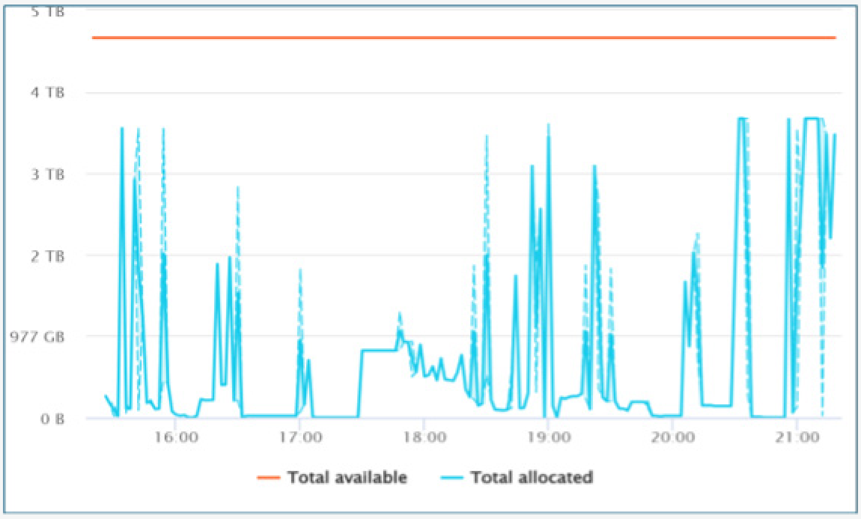


7.4. Hadoop - Data Platform

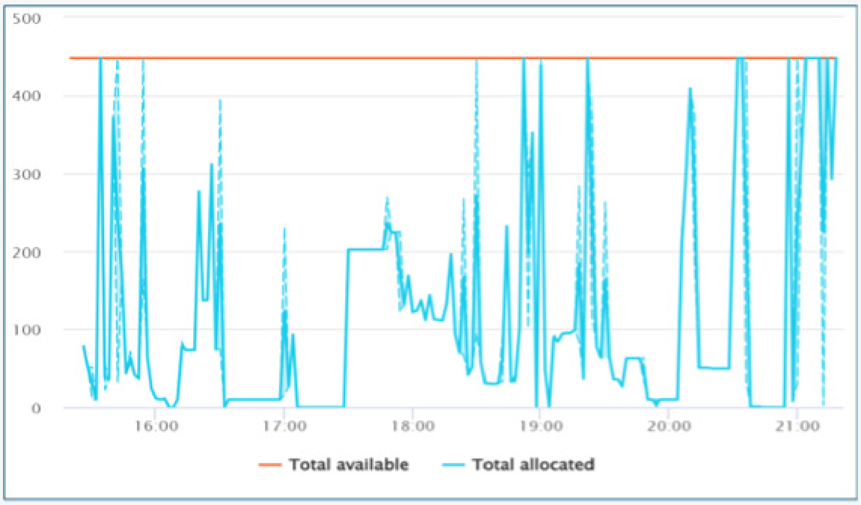
Data Platform: Cloudera Data Platform 7.1 (Hadoop / Spark)

7.4.1. Hadoop – Peak Usage During Month End Test

The figure below shows memory usage for Month End test. There are several spikes in memory usage to approximately 3.5 TB.



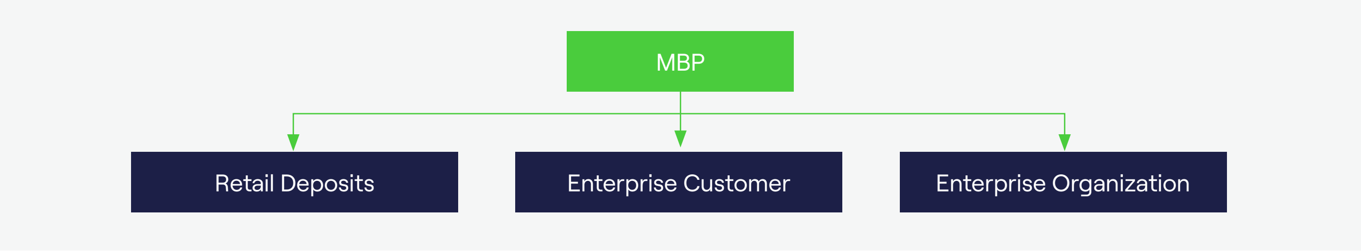
The figure below shows CPU usage for Month End test. There are several spikes using all 448 allocated vCores:



8. Key Architectural Components

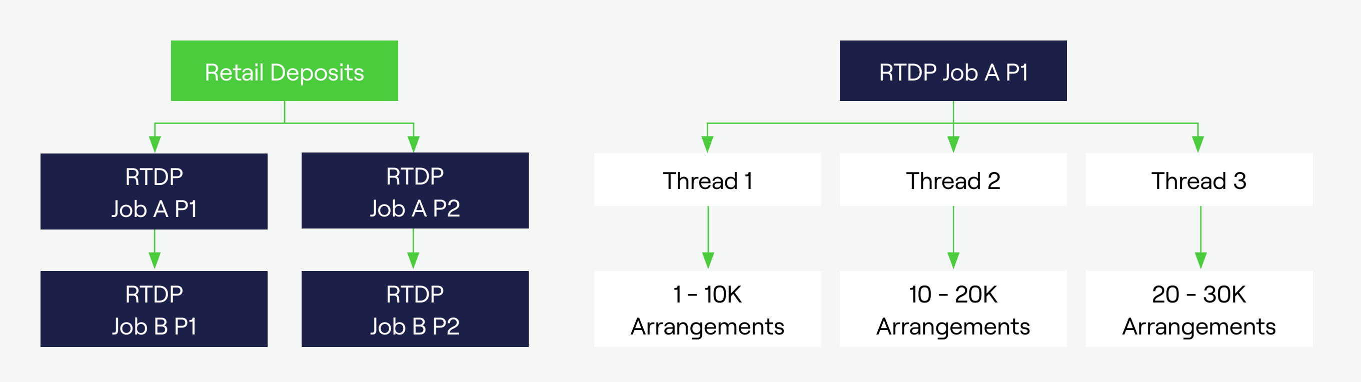
8.1. MBP 3.11 Periodic Processing – Parallel Processing at Three Levels

Parallelization was implemented across components, within components, and within job partitions. Each aspect is described in more detail below. Batch flows across different components can execute in parallel, meaning each component can run one job at a time, but different components can run different jobs simultaneously. Additionally, the CAPE RAS platform has the capability to run multiple different jobs concurrently.



Parallelization within a job was achieved through configuration-driven Job Partitioning. A large job workload is partitioned across multiple pods, each representing a Job Partition.

A specific Job Partition can be pinned to a distinct set of Database Partitions (1:n or n:1) to reduce database overhead associated with Exadata cluster synchronization ('cluster waits'). For more details, see the section on application node pinning below.



Some of the lower-demand global jobs do not implement partitioning, so run on a single Partition 0. Those jobs tend to be fast-executing, and the lack of partitioning is not a performance concern.

A single Job Partition can in turn parallelize its workload across a configurable number of concurrently executing threads; this is specified by the job's 'grid size'. The grid size enables additional multi-threading which in turn typically demands proportionally more CPU. Given our intent to right-size each container, the grid size is closely managed.

Note that if network latency to the database is high, each thread spends proportionally more time waiting for results of database calls, and more threads are needed to make optimal use of pod CPU resources.

8.2. Periodic Processing – Intraday

Apache Airflow is a platform used in the MBP solution benchmark to programmatically author, schedule, and monitor workflows. It uses Directed Acyclic Graphs (DAGs) to define a collection of tasks and their dependencies, otherwise referred to as a workflow.

Intraday Periodic Processing was executed prior to the Benchmark Day End and Month End Periodic Processing cycles via a separate DAG/job flow sequence. This DAG included core Retail Deposit and RAS jobs that do not require all postings for the day to be completed, and can be executed ahead of the main Periodic Processing cycle. The value proposition for Intraday Periodic Processing is to reduce the total processing load and execution time for the Day End/Month End Processing flow, as well as to reduce the overall infrastructure footprint required for Periodic Processing.

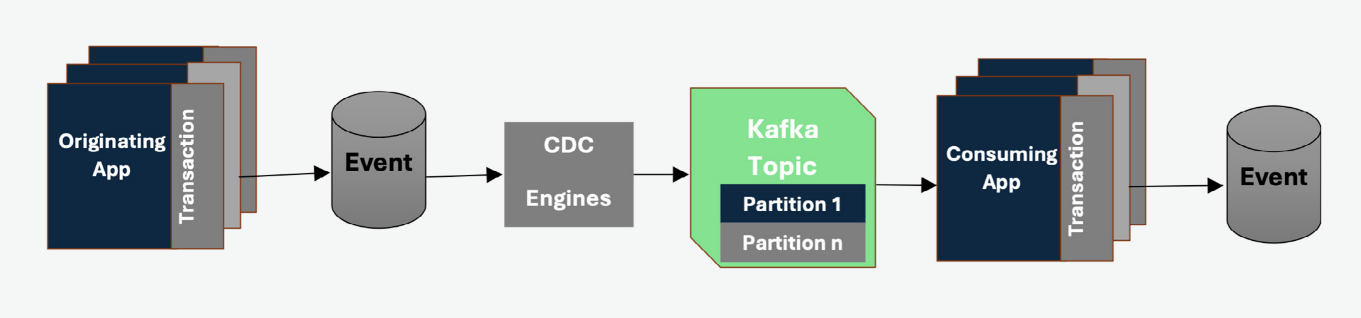
The types of jobs in this Intraday Processing included:

- Transaction Exception Review Process
- Charge Pre-Process and Charge Related Jobs
- Process Pending Balances
- Posting to Archive
- Transaction Data Load (TDL) jobs

8.3. Event Processing Architecture

This Benchmark introduced a streamlined event processing mechanism for the Modern Banking Platform (MBP), building on the previously established Change Data Capture (CDC) architecture. As part of transactional processing, components create event payloads which are inserted into the CDC Outbox tables as part of a single commit of the MBP transaction. From there, a horizontally scalable CDC Engine reads the records and publishes them directly to Kafka in a store and forward pattern. Events are distributed evenly across Kafka topic partitions using a partition key header computed from the identifiers of specific customers or arrangements, ensuring that events related to the same entity stay in the correct publishing order within the same topic partition ("Event Sequencing"). When applicable, the engine also facilitates the creation of backward compatibility versions of the events via JSON, transforming in a way that maintains order of delivery into each Kafka partition.

Event consumer applications hook into partitioned Kafka topics to consume events directly and commit them to their own component's CDC Inbox table, with concurrency mapping to the number of topic partitions. These event consumer applications can scale horizontally to ensure event consumption remains current even at high publish rates. The Outbox and Inbox can be reconciled via a separate Reconciliation Engine.



Benchmark reached **peak event volumes of near 18,000 events per second (EPS)**, resulting from highly scaled Batch processing and from concurrently executing online transaction load.

Note: Backward compatible version support enables the core banking ecosystem to more easily transition from release to release. However, **the number of backwards versions should be strictly limited to avoid unnecessary event volume** and the associated cost of processing that can result if left unmanaged. See MBP version support document for details.

8.4. Within-Namespace Load Balancing for Online/Microgateway

Incoming traffic from outside the namespace is generally load balanced properly at the Route/Ingress layer. However, if HTTP traffic is not evenly distributed across application instances/pods, it often indicates that upstream client applications in the same namespace are caching connections to specific service pods. In such situations, throughput will be soft-capped at the level where the most-utilized service pods reach their capacity limit, causing response times to deteriorate and potentially leading to pod failure or auto-restart. Any additional pods spun up will generally receive even less traffic than the least-utilized pods.

In MBP 3.11, the load balancing architecture has been enhanced to implement Client-Side Load Balancing in all applications that make calls to other services within the same namespace. This pattern was implemented for all MBPAPI pods as well as the Microgateway to ensure that calls from these components are evenly distributed across called services. When Kubernetes decommissions or destroys pods downstream, the client pods may have old connections to the destroyed pods. This challenge was addressed in the benchmark with a new architecture to recognize and refresh these connections to the remaining pool in a timely manner.

8.5. Turbo Customer API

To enhance Customer API performance, the Get Customer MBP API was re-architected using a new implementation referred to as a Turbo API pattern. The Turbo Customer API utilizes the Turbo SQL design technique, which combines Common Table Expressions (CTEs) and an optimized database interaction model.

Below are some of the benefits realized by the Turbo Customer API design:

- Reduced stress (in turn, CPU usage) on database
- Reducing network overhead
- Less thread blocking
- Improves application server resource and throughput
- Process more transactions per POD
- Fewer PODs required for same output of transactions
- Improved concurrency in app
- Better client experience
- Significant cost savings
- Reduced database connections
- More readable queries
- Result set better matches domain model
- Save in IO

8.6. Key Architectural Components – Special Customizations for Benchmark

The following techniques were special customizations for the Benchmark and were instrumental to system performance. The descriptions below provide details on specific parameters and techniques used by the Benchmark team to achieve the reported results.

8.6.1. Batch App Layer – Scaling and Key Parameters

Day End workloads depend on the total number of customers and accounts to be processed, as well as the volume of transactions during the respective period (day or month). Since Periodic Processing is a highly database-centric application, the total number of database CPUs available is a critical factor, along with the IOPS supported by the database. Some database-intensive batch jobs spend over 60% of their time executing database operations, requiring a substantial portion of a database CPU for each app-side worker thread. Other batch jobs require more processing power on the application side, with less relative work done in the database. The portion of time spent executing database operations is proportional to the Job Constant. The precise value of the constant also depends on the specific latency between the application tier and the database. Higher latency will somewhat reduce the Job Constant, as database CPUs are not executing queries while the data is being transmitted. However, other database resources will still be busy.

The other scaling parameters are:

- **Batch Partitioning** – horizontal scaling of pods can help with the workload across pods at the app tier
- **Grid Size** – the number of parallel worker threads in each Batch pod. Grid size can be set per job.

Scaling Heuristic used in Benchmark

Roughly, if the product of (**Batch Partitions x Grid Size x Job Constant**) exceeds the number of database CPUs, increased database concurrency will counteract parallelism and impose a lower limit on batch runtime. Optimally, each CPU can handle only a single database connection with minimal context switching. Each parallel thread may require its own connection.

Note: Grid Size will drive up container CPU utilization, and increased grid size will only be effective until CPU throttling occurs at the pod level. Once that occurs, it is more effective to increase the number of partitions, provided that incremental DB CPU resources are still available.

Grid Size Parameters used for 25M Month End Run:

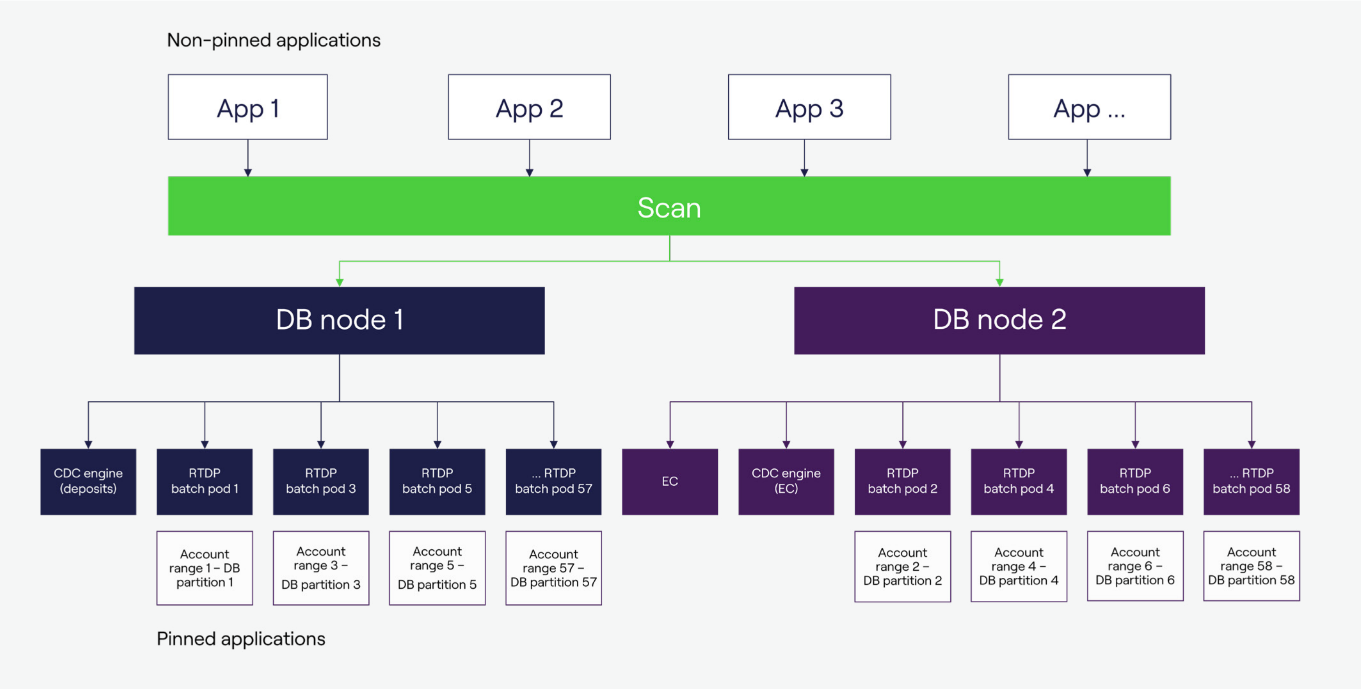
Default Grid Size	RTDP_CHRGPRCS001 Grid Size	RTDP_BALPERIODSTAT001 Grid Size
10	12	16

8.6.2. Application Node Pinning

Application node pinning was implemented to reduce cluster wait events, where the data blocks needed for a transaction are being updated by both database nodes. By "pinning" applications to specific database nodes, the database no longer must maintain these data blocks on both nodes, greatly improving performance. This was implemented in the following components:

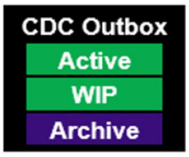
- CDC Engine: Deposits and Enterprise Customer were placed on opposite database nodes
- Event Listener Apps: For Deposits and Enterprise Customer, these were pinned to the same node as the other component's CDC engine, opposite their own CDC engine
- Enterprise Customer Service App: Node pinning was used only for online Enterprise Customer pods. Node pinning wasn't implemented for Enterprise Customer Batch pods, as there was no expected performance improvement
- RD Batch App: Each of the 58 Retail Deposit Batch pods was given an account range that mapped to a physical Oracle partition and was pinned to a specific database node. This allows all the blocks associated with a specific tablespace segment to remain on one database node, thus eliminating cluster wait events with blocks being updated on both nodes.

All other components were processed through the database Scan Listener, which distributes connections and requests to each of the available nodes.



8.6.3. Event Processing – CDC Outbox Tuning for High Throughput

The CDC Outbox design allows for high-performance transaction processing, ensuring related event capture without the overhead of a two-phase commit (XA) protocol. It serves as a store-and-forward mechanism, facilitating asynchronous message processing while also acting as an audit table for various points of reconciliation. CDC Outbox processing is managed using three table types: Active CDC Outbox, Work-in-Progress, and Archive.



Note: CDC Engine configurations should be adjusted to match the size of the environment and the expected event throughput as well as the configured backward compatibility transformations. This benchmark implemented the latest CDC 3.0 version, allowing CDC Engines to be scaled horizontally.

Benchmark Configurations:

CDC-outbox

- Auto-partition time slice: 5 minutes
- Partition-retention: 24 hours
- Archive-retention: 1 month

CDC-engine Configurations

```
cdc:
  classic:
    archive:
      batch-limit: 5,000
      worker:
        task-executor:
          queue-capacity: 100,000
    escrow:
      batch-limit: 5,000
    event-partition-count: 7
    migrate:
      batch-limit: 5,000
    publish:
      worker:
        partition-count: 20
  cluster:
    minimum-size: 3
  memory:
    average-record-size: 10,000
```

9. Infrastructure Recommendations – Bill of Materials

The Bill of Materials shown below is a recommended set of resource requirements to adequately service all MBP system processing needs for a Model Bank with 25 million accounts.

NOTE: HA capacity needs are not estimated in this section, only the capacity needed for a single non-HA environment.

Bill of Materials (BOM):

Component Type	Provider	Platform	CPU (Cores)	RAM (GB)	Storage (TB)	Estimated Storage Growth/Day	Transaction / Data Volume
RDBMS	Oracle	Oracle 19c	230	675	36.0	318 GB/Day	875,000 Requests/ Second
Kubernetes	Openshift	Red Hat OpenShift 4.12	1,035	2,170	8.28		
Hadoop	Cloudera	Cloudera Data Platform 7.1	448	4,400	540	1,480 GB/Day	
Kafka	FIS	Cloudera Data Platform 7.1 (Kafka)	48	192	6	320 GB/Day	18,000 (peak) Events/ Second
Batch Client	Airflow	Apache Airflow 2.6.3	94	268			
API Gateway	FIS	Code Connect					4,800 API Requests/ Second
Tooling	Cisco	Splunk 9.2.1					11,000 GB/day

Capacity estimates resulting from the column Estimated Storage Growth/Day need to be applied in the context of specific retention policies. For example, for the RDBMS, the assumption is that events in the CDC tables are retained for 30 days only, while event retention on the Kafka side is assumed to be three days.

9.1. BOM – RDBMS

BOM values for CPU and RAM were determined from maximum usage for each RAC node during Benchmark testing. A safety factor of 25% was applied to this initial value, then this value was multiplied by two to account for two database RAC nodes. Refer to section 7.2 for full database metrics from Benchmark testing.

Max usage (PGA + SGA) during testing was 275 GB; BOM values for RAM are then 675 GB (275 plus 25% safety factor * 2 nodes).

Database Storage Breakdown:

* Physical Tables = 27 TB
* UNDO = 2 TB
* TEMP = 2 TB
* REDO LOG = 5 TB (this is for a 1 log in a group, 163 groups at 32 GB / log)

Maximum transaction volume for both nodes was approximately 700,000 requests/second, occurring during the Month End test. A safety factor of 25% applied to this value is 875,000 requests/second.

To support one year of operations at constant volumes, we estimate that the DB storage would have to grow by around 108 TB.

9.2. BOM - Kubernetes

Total CPU and Memory Limits for the BOM are derived from set limits for each of the two main load profiles. While CPU/Memory usage may be below set limits during testing, it cannot be assumed that CPU/Memory can be lowered further below the set limits without adverse consequences.

Additionally, the load profile for Periodic Processing must account for 60% of ‘Online Transaction Load’ during Intraday processing; hence recommended limits for the following Container Categories are calculated by: (0.6*Limit of Online Transaction Load): Customer Service App, MBP API (including Microgateway), and Retail Deposit Service App.

For some critical applications under load, Benchmark set CPU requests equal to limits to minimize run-by-run variability and throttling. Production environments might accept different trade-offs and set requests lower at the cost of increased variability.

After calculating the total CPU and Memory Limits for each of the test profiles, these values were compared, and the larger of each was selected for the BOM. Finally, a 15% safety factor was added to create final values for the BOM.

Test Type	Sum of CPU Limits	Sum of CPU Limits (+15%)	Sum of Memory Limits	Sum of Memory Limits (+15%)
Online Transaction Load	899	1,034	1,697	1,951
Day End/Month End (Adjusted for 60% online load during Intraday)	897	1,031	2,019	2,168

9.3. BOM – Hadoop

BOM values for CPU and RAM were determined from peak values observed during the Month End Benchmark test (see section 7.4.1). During that test, CPU peaked at the maximum allocated 448 CPUs (equivalent to seven worker nodes with 64 CPUs each). BOM value was determined to require 4,400 GB.

Node	Count	CPUs each	RAM (GB) each
Worker Node	7	64	754

BOM values for Hadoop storage are estimated from the approximate daily event load described in section 7.3.3, with comparison to production storage growth for current Hadoop clients to estimate the corresponding growth in Hadoop storage for a bank with 25 million accounts:

- Projected monthly storage growth = 45 TB/month
- Projected yearly storage growth = 540 TB/year

9.4. BOM – Kafka

Our Model Bank produced nearly 200 million projected events for a typical processing day. Uncompressed event payload volume amounted to 1,536 Gigabytes of data for such a day. Compression is in general use for storage and transmission of events, resulting in data volume reduction of between 75% - 85%, depending on use case.

Please see Section '7.3 Events and Kafka' for detailed assumptions and underlying metrics.

We assume that events are retained in Kafka for three days, and in DB CDC tables for up to a month. Reducing event retention periods can have large-scale benefits around required storage capacity.

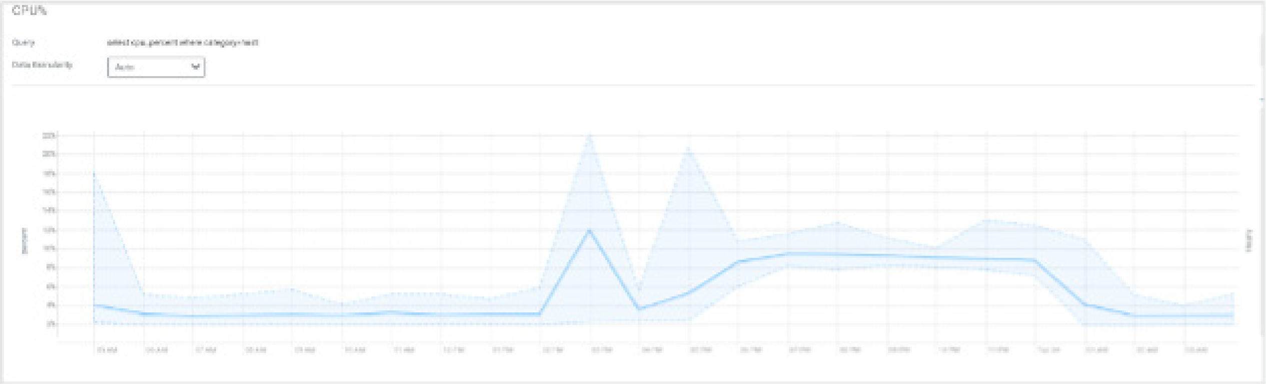
All Kafka topics were configured to 16 partitions and replicated to 3 nodes each. There were four different Kafka consumer groups: one for Retail Deposit, one for Enterprise Customer, one for CAPE RAS, and one for the WebHook engine, used for external event delivery via HTTP POSTs.

Resources shown below are sufficient for Kafka processing of the tested 25M workload:

Node	Node Count	FIS Build Type	CPUs	RAM (GiB)	Storage Type	Broker Storage
Broker	3	GEN3	16	64	Premium SSD	2 TB

A base Kafka install is assumed to be present; the resources above represent incremental infrastructure needed to support a client with 25M accounts. This determination was based on observed CPU usage during peak event loads of the Benchmark tests. CPU usages of these Broker Nodes was well within maximum capacity, never exceeding 25%.

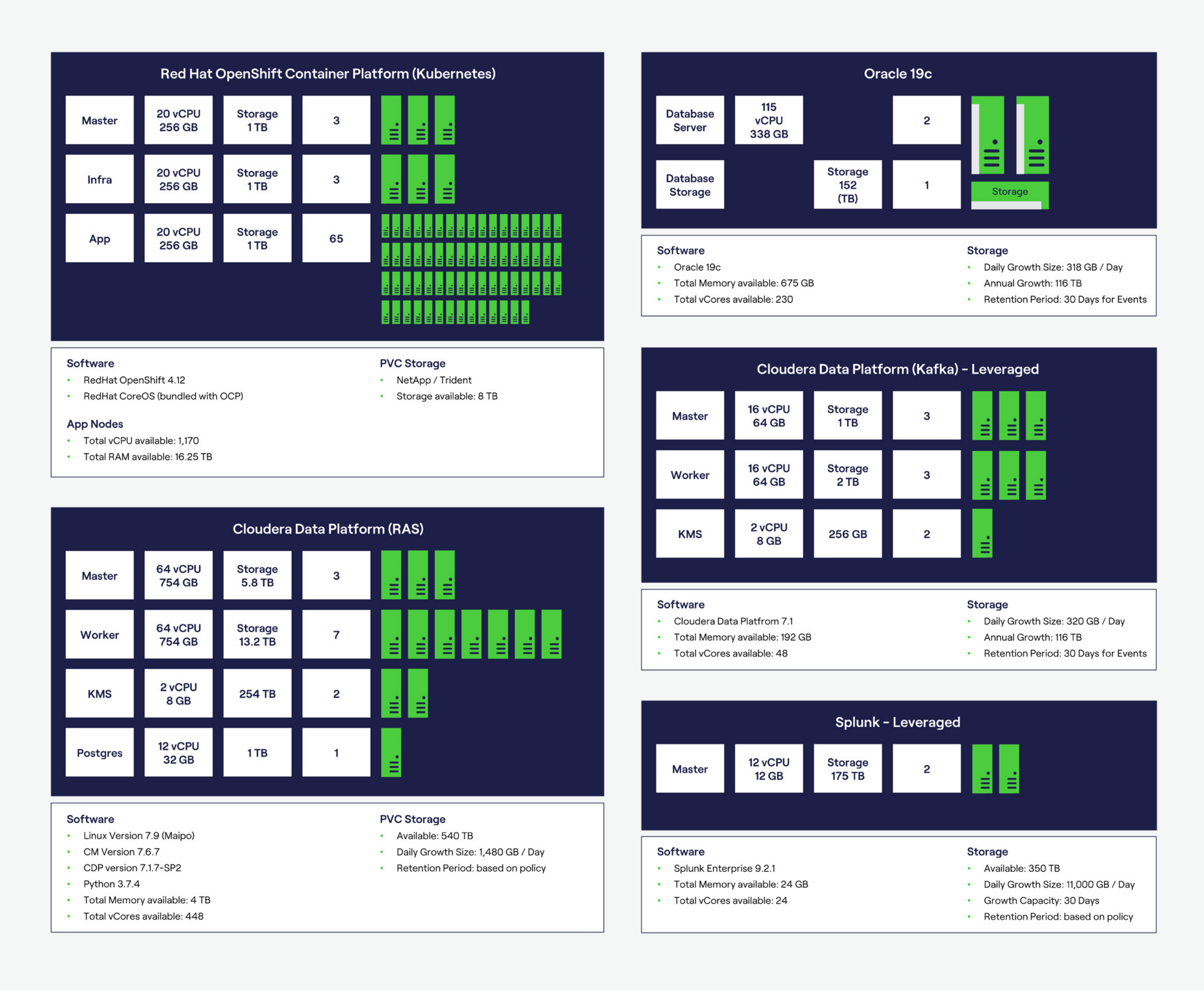
Kafka Broker % CPU Usage during Month End Test:


CPU%

Query select cpu_percent where category=test

Data Granularity Auto

9.5. Infrastructure Diagram



Red Hat OpenShift Container Platform (Kubernetes)

Master	20 vCPU 256 GB	Storage 1 TB	3
Infra	20 vCPU 256 GB	Storage 1 TB	3
App	20 vCPU 256 GB	Storage 1 TB	65

Software
- RedHat OpenShift 4.12
- RedHat CoreOS (bundled with OCP)

App Nodes
- Total vCPU available: 1,170
- Total RAM available: 16.25 TB

PVC Storage
- NetApp / Trident
- Storage available: 8 TB

Cloudera Data Platform (RAS)

Master	64 vCPU 754 GB	Storage 5.8 TB	3
Worker	64 vCPU 754 GB	Storage 13.2 TB	7
KMS	2 vCPU 8 GB	254 TB	2
Postgres	12 vCPU 32 GB	1 TB	1

Software
- Linux Version 7.9 (Maipo)
- CM Version 7.6.7
- CDP version 7.1.7-SP2
- Python 3.7.4
- Total Memory available: 4 TB
- Total vCores available: 448

PVC Storage
- Available: 540 TB
- Daily Growth Size: 1,480 GB / Day
- Retention Period: based on policy

Oracle 19c

Database Server	115 vCPU 338 GB		2
Database Storage		Storage 152 (TB)	1

Software
- Oracle 19c
- Total Memory available: 675 GB
- Total vCores available: 230

Storage
- Daily Growth Size: 318 GB / Day
- Annual Growth: 116 TB
- Retention Period: 30 Days for Events

Cloudera Data Platform (Kafka) - Leveraged

Master	16 vCPU 64 GB	Storage 1 TB	3
Worker	16 vCPU 64 GB	Storage 2 TB	3
KMS	2 vCPU 8 GB	256 GB	2

Software
- Cloudera Data Platfrom 7.1
- Total Memory available: 192 GB
- Total vCores available: 48

Storage
- Daily Growth Size: 320 GB / Day
- Annual Growth: 116 TB
- Retention Period: 30 Days for Events

Splunk - Leveraged

Master	12 vCPU 12 GB	Storage 175 TB	2

Software
- Splunk Enterprise 9.2.1
- Total Memory available: 24 GB
- Total vCores available: 24

Storage
- Available: 350 TB
- Daily Growth Size: 11,000 GB / Day
- Growth Capacity: 30 Days
- Retention Period: based on policy

10. Conclusion – Summary of Findings

The Performance Benchmark demonstrated the fundamental scalability of the MBP Solution up to a volume of 25 million accounts, achieving or exceeding all targeted Service Level Objectives (SLO). The traffic mix used to achieve these objectives reflects the most representative processing profile associated with a bank of this size. This mix was executed at a proportional volume beyond the scale of any past or current production client, which includes transactions and queries known to be processing-intensive. Specific results will vary based on traffic mix and institutional client profile. Proactively managing the use of customizations and optional performance-impacting features, such as excessive backward compatibility, can improve both cost efficiency and performance of the solution. This includes maintaining currency with the latest MBP releases and upgrading to the latest API and Event versions.

Major accomplishments of the 25M Solution Benchmark include:

- Achievement of processing over 4,700 Requests Per Second (RPS) within the target Service Level Objective (SLO)
- During the Online Transaction Load, Soak Test, and Periodic Processing tests, the following SLOs were achieved: the Enterprise Customer Transaction Mix reached 1,630 RPS with an average response time of 66 milliseconds; the Retail Deposits Transaction Mix achieved 3,136 RPS with an average response time of 98 milliseconds; and MBP Core and RAS Monthly Day End Processing completed in 5 hours and 35 minutes
- Demonstration of the fundamental scalability of the MBP solution, with near similar results when tested on 5 million versus 25 million accounts
- Notable improvement in total request throughput with increased account and customer volume compared to previous 10M Benchmark. Retail Deposits throughput increased from 1,400 RPS to 3,136 RPS, while average response time decreased from 468 milliseconds to 98 milliseconds. Enterprise Customer throughput increased from 200 RPS to 1,630 RPS, while average response time decreased from 171 milliseconds to 66 milliseconds
- Development and testing of numerous technologies and architecture techniques, including but not limited to Intraday Processing, Application Node Pinning, Within-Namespace Load Balancing, updated Event Processing Architecture, and Turbo Customer API
- Detailed Bill of Materials for a full recommended solution size for institutions of 25 million accounts.

During the 25M Performance Benchmark, the Modern Banking Platform has demonstrated exceptional performance, scalability, and reliability under high-load conditions. These results demonstrate the ability to meet all service level objectives and affirms the platform's readiness to scale up to 25 million accounts for a representative large bank.

11. Appendices

11.1. Appendix – Infrastructure Kit Provisioned for the Benchmark

11.1.1. Kubernetes Resources

Kafka Platform: Red Hat OpenShift 4.12

Node Type	Number of Nodes	CPUs	CPUs Available	RAM (GB)	RAM (GB) Available	Storage/ Node (TB)
App	65	1,300	1,040	16,380	3,900	1
Infra	3	60	48	756	603	1
Master	3	60	48	756	603	1

11.1.2. Oracle Resources

RDBMS Platform: Oracle Exadata X10; Oracle Database 19c

Application	Hardware
Version	Oracle 19c
RAC	Yes
Archive	Yes
Server Type	Exadata X10
vCPU	384
Memory	2,304 GB
SGA	240
PGA	200

11.1.3. Hadoop Resources

Data Platform: Cloudera Data Platform 7.1

Includes Hive, Spark, HDFS, YARN, MapReduce

- Linux Version - 7.9 (Maipo)
- CM Version – 7.6.7
- CDP version – 7.1.7-SP2
- Python – 3.7.4

Node	Count	CPUs	RAM (GB)	Storage (GB)
Master Node	3	64	754	5,800
Worker Node	7	64	754	13,200
kms Node	2	2	8	254
PostgresDB	1	12	32	1,000

11.1.4. Kafka Resources

Kafka Platform: Cloudera Data Platform 7.1 (Kafka)

Node	Nodes Count	FIS Build Type	CPUs	RAM (GiB)	Storage Type	Broker Storage
Broker	4	GEN2	12	32	Premium SSD	1 TB
Broker	4	GEN3	16	64	Premium SSD	1 TB
Utility Nodes	4	GEN2	8	32	Premium SSD	N/A
Utility Nodes	4	GEN3	16	64	Premium SSD	N/A

11.2. Appendix - Observability

Application	Usage
Dynatrace	Data visualization, triage
O11y Toolchain	Data visualization, triage
Splunk	Triage

11.2.1. Dynatrace

All OpenShift nodes are instrumented with Dynatrace APM. Dynatrace monitors code execution at the thread level, and supports tracing calls across multiple components to discover performance bottlenecks. It also supports live memory analysis. In addition, Dynatrace monitors utilization and error metrics associated with workloads at the Kubernetes level, and it exposes aggregate metrics for DB queries executed in the context of monitored service calls.

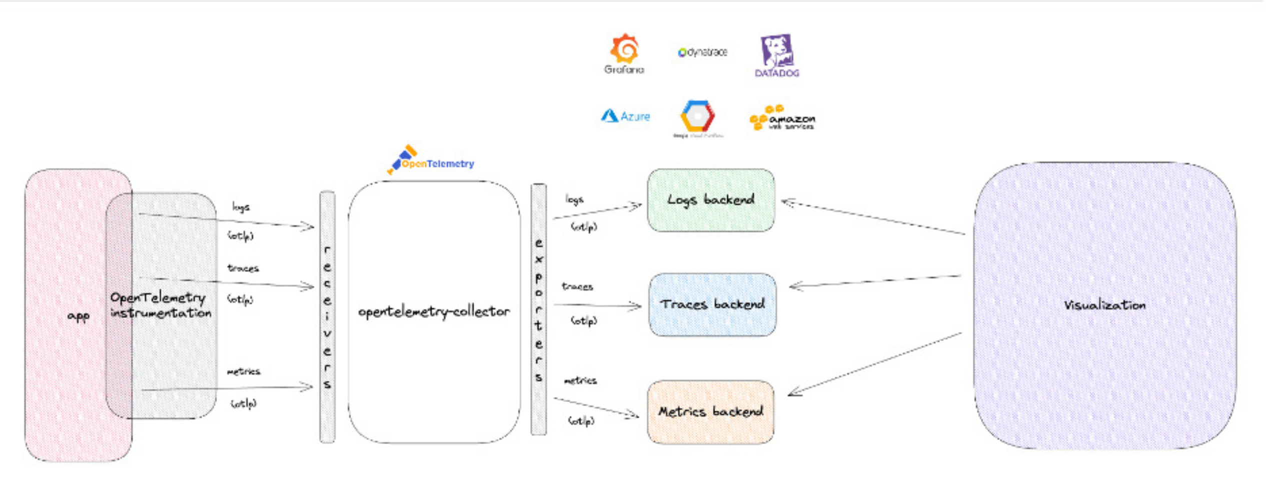
11.2.2. Splunk

Splunk is a powerful software platform used primarily for searching, monitoring, and analyzing machine-generated data in real-time. It is widely used for load management, security, and operational intelligence. Key features are Data Indexing, Search & Query, Real-Time Monitoring, Data Visualization, Alerting, Machine Learning & Integration.

Splunk capabilities include Application Management, Security & Compliance, IP Operations & Monitoring and Business & Web Analytics. Splunk relies on its indexes to store data and does not require any database. Splunk gathers all relevant information into one central index, making it easy to access, analyze, and visualize machine-generated data from multiple sources for optimizing machine performance.

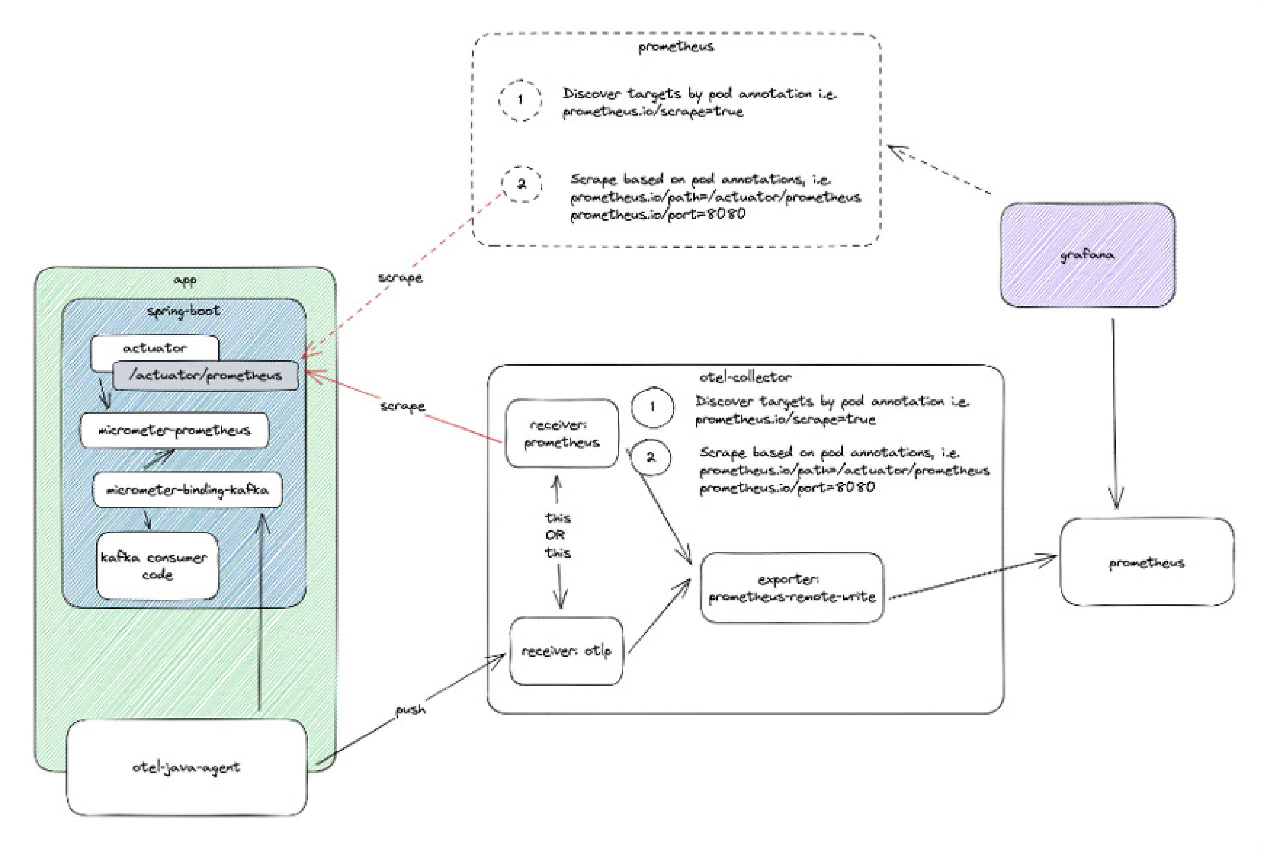
11.2.3. OTEL/Prometheus/Grafana (O11y Toolchain)

Open Telemetry is integrated into select applications to collect and export key metrics.



Prometheus is a metrics collection, aggregation, storage and query tool. Metrics are provided by services at a specific location (or can be sent directly to a Prometheus server in certain cases) and are collected by various Prometheus instances. By default, Prometheus uses local storage and trims metrics after a fixed age or aggregate size, but compacting backends are available (cortex, thanos, mimir) which reduce the resolution of older metrics and/or ship them to cold storage locations for historic metric availability.

Integration of Open Telemetry, Prometheus, and Grafana is described below:


prometheus
1
Discover targets by pod annotation i.e. prometheus.io/scrape=true
2
Scrape based on pod annotations, i.e. prometheus.io/path=/actuator/prometheus prometheus.io/port=8080
grafana
scrape
app
spring-boot
actuator
/actuator/prometheus
micrometer-prometheus
micrometer-binding-kafka
kafka consumer code
scrape
otel-collector
receiver: prometheus
1
Discover targets by pod annotation i.e. prometheus.io/scrape=true
2
Scrape based on pod annotations, i.e. prometheus.io/path=/actuator/prometheus prometheus.io/port=8080
this
OR
this
exporter: prometheus-remote-write
prometheus
receiver: otlp
push
otel-java-agent

11.2.4. Database Monitoring

Database monitoring was achieved with Oracle Enterprise Manager (OEM), which is the Production Oracle DBA's tool to the FIS Private Cloud environments.

Through integration with the Oracle product stack, OEM automates the management of Oracle Applications and Oracle Database instances, along with other hardware and software assets, including middleware and engineered systems. Enterprise Manager was selected because it delivers:

- Increased visibility and intelligent analytics
- Comprehensive lifecycle automation and control
- An enterprise-grade management platform that is secure, accessible, and extensible.

11.3. Appendix - Testing Summary and Procedures

Test Type	Description
MBP & CAPE RAS Periodic Processing Test – Day End/Month End	Periodic Processing Tests were performed using the standard job flow published as part of the base solution. These tests also included executing a proportional volume of transactions via APIs to simulate daily activity for a bank and Ad hoc (Intra-Day) batch, prior to executing the Day End batch. During batch execution, a concurrent online load was executed, to simulate conditions existing in a Production Environment. Month End job flow is the same as Day End but executed with End of Month processing date.
Online Transaction Load Test	These tests included execution of a representative subset of MBP APIs (REST and SOAP) based on the observation of the high-volume online traffic of existing MBP production clients hosted in the FIS Datacenter. These tests used the 2023 and 2024 transaction mix profiles.
Soak Test	Running the Steady-state test cycle load with lower baseline volume, for longer duration (8 hours) as an attempt to identify the impact on system components due to long running production-like workload timeframes. This test used the 2023 transaction mix profile, running for 8 hours with 70% of max steady-state throughput of that mix, at combined 2,840 RPS.

11.3.1. Performance Toolchain - Online Transaction Load Test

The Performance Toolchain provided a scalable test harness that was used for load testing online applications using Apache JMeter test scripts. It was deployed in Kubernetes (OCP) as worker pods reporting to a central coordinator pod, and was used for intensive load tests, simulating hundreds or thousands of simultaneous users.

The Performance Toolchain included Apache JMeter, Jenkins, and Taurus. It integrated with an Observability Toolchain comprised primarily of Prometheus and Grafana. Performance Toolchain metrics were sent to Prometheus using the built-in backend listener available in JMeter. Grafana was used to render this data and app as well as infrastructure metrics in an easily consumable set of dashboards. Each test execution generated an AWR report and links into Grafana for the test timeframe.

Primary online reference results were documented via JMeter reports, capturing throughput, various latency measures (mean, median, 90th percentile, 99th percentile, standard deviation), and error rate.

11.3.2. Test Procedure – Online Transaction Load

1. Restore the database and/or host data to known initial state
2. Redeploy/restart the pods
3. Run a full workload test to warm up the database and the servers
4. Run the test scenario with final identified user load distributed across key business process for at least 1 hour at steady state load
5. Once the test is completed, generate the test report as well as AWR report and perform the analysis
6. If required by analysis, log defect tickets, apply and capture tunings or settings for improving performance, and/or modify load, and restart at the first step.

11.3.3. Test Procedure – Soak Test

1. Restore the database and/or host data to known initial state
2. Redeploy/restart the pods
3. Run a full workload test to warm up the database and the servers
4. Run the test scenario with final identified user load distributed across key business process for at least 8 hours at steady state load
5. Once the test is completed, generate the test report as well as AWR report and perform the analysis
6. If required by analysis, log defect tickets, apply and capture tunings or settings for improving performance, and/or modify load, and restart at the first step.

11.3.4. Airflow for Periodic Processing Tests

Airflow batch broker application is used to orchestrate the MBP Batch Processing workflows. The required Static DAG was generated from YAML configuration files. There are separate DAGs for the Intra-Day and Day End Batch. CAPE RAS is part of both the DAG's.

11.3.5. Test Procedure – Periodic Processing

One-time Preparation

1. Run the Batch Transaction Load via API to simulate a day's worth of transactions for ~20% of all Checking and Savings accounts
2. Run Intraday Batch with Concurrent Online Load profile (2,440 RPS)
3. Create a "Pre-Day End" restore points of database state for all SOR (Retail Deposits, Enterprise Customer, CAPE RAS).

Test Execution

1. Reset the SORs to known initial "Pre-Day End" state
2. Redeploy / restart the pods
3. Start Concurrent Online Load profile (1,220 RPS); allow load to reach steady state
4. Trigger Day End Batch/Month End Batch via Airflow
5. Once the test is completed, generate the test report as well as AWR report and perform the analysis
6. If necessary, log defect tickets, apply and capture tunings or settings for improving performance. Decide if another run is needed.

About FIS

FIS is a financial technology company providing solutions to financial institutions, businesses and developers. We unlock financial technology that underpins the world's financial system. Our people are dedicated to advancing the way the world pays, banks and invests, by helping our clients confidently run, grow and protect their businesses. Our expertise comes from decades of experience helping financial institutions and businesses adapt to meet the needs of their customers by harnessing the power that comes when reliability meets innovation in financial technology. Headquartered in Jacksonville, Florida, FIS is a member of the Fortune 500® and the Standard & Poor's 500® Index. To learn more, visit FISglobal.com.

🖥 fisglobal.com/contact-us

in linkedin.com/company/fis

𝕏 x.com/fisglobal

FIS | Advancing the way the world pays, banks and invests™