fiserv.

White Paper

Signature® International Modernization on IBM Hybrid Cloud

January 2024

the speed of life

Objective

In line with continuous innovation for our clients, Fiserv is working with IBM to provide a flexible roadmap to include Signature cloud-native workloads on Red Hat® OpenShift®, to run natively on IBM's latest Power10 processor-based servers that host the international version of the Signature core banking platform.

By consolidating existing Fiserv workloads and adding new cloud-native microservices on the IBM Power® platform, clients can accelerate application modernization to deliver agile and ground-breaking Fiserv services to their customers on private or hybrid cloud-service deployments.

By taking advantage of incremental, in-place modernization, customers are able to add new Red Hat OpenShift-based services provisioned as containers on the same IBM Power environment, which also runs traditional applications on IBM i virtual machines.

This provides an accelerated path to cloud modernization by leveraging existing unused or unassigned resources on the IBM Power platform for Microservices for Signature from Fiserv. By running a mix of virtual machines container microservices on the same IBM Power platform, the productive utilization of the servers is maximized.

Clients can also expect superior scaling performance, and can repurpose the integrated hardware and system software security profiles that they are already using in production to reduce the effort and cost of modernization, versus having to rip and replace systems to achieve new cloud capabilities.

Introduction to Microservices for Signature

Microservices for Signature enables your organization to meet the ever-increasing demands of consumers by providing a modern, highly scalable REST API. The microservice-based architecture design enables you to maximize your investment in the systems, and offers the flexibility to incorporate exciting new functionalities with agility and certainty.

Microservices for Signature provides a service layer in the Signature core banking platform, which enables the integration to Signature business applications and mobile and web channels, as well as system-to-system integration. With proven scalability, Microservices for Signature delivers information logically and consistently to any channel, at any time, on any device – 24/7/365.

Executive Summary

IBM and Fiserv conducted an IBM Garage Laboratory test to evaluate the benefits of running Microservices for Signature and UUI using Red Hat OpenShift Container Platform, all running on the Power10 IBM i platform.

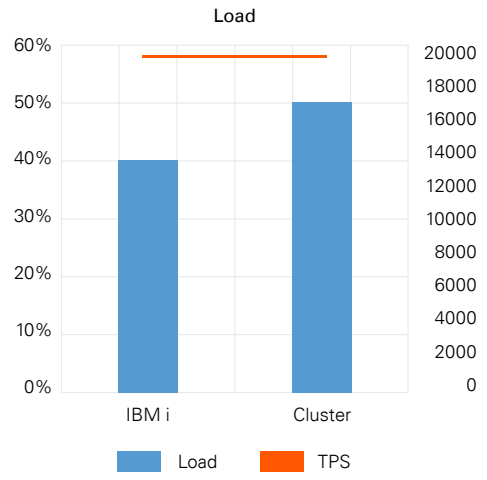
The objectives were:

- Show how all elements worked together seamlessly on the same platform
- Determine the possible throughput on IBM Power
- Explore how to utilize spare resources on the IBM i for running Microservices workloads

The performance achieved was approximately 19,000 transactions per second (TPS), with a mixed set of microservices without audit logs. See the list below of microservices used in the performance test.

This load was achieved:

- With the Red Hat OpenShift Cluster load on the Worker nodes at 50% CPU and memory – 30%
- The IBM i load for the Core was 50% CPU



Modernizing With Red Hat OpenShift on IBM Power

Our clients are modernizing their applications with cloud-native capabilities leveraging Red Hat OpenShift, enabling workloads for a hybrid cloud environment. The strength and leading isolation capabilities of IBM Power's hypervisor technology, PowerVM, enables clients to securely run AIX® or IBM i applications side by side with cloud native applications on Red Hat OpenShift. This maximizes efficiency and utilization, and provides greater flexibility to run across multiple clouds. Red Hat OpenShift extends Kubernetes with built-in tools to enhance application lifecycle development, operations, and security. With Red Hat OpenShift, clients can consistently deploy workloads across multiple public or private clouds with ease.

Running Red Hat OpenShift on Power10 enables customers to take advantage of the superior scale, RAS and security advantages that IBM Power is well known for. Processes running in container microservices can leverage up to eight simultaneous threads per Power10 core (versus two threads on an x86 core). Power10 has built-in hardware security, including transparent memory encryption, as well as full homomorphic and quantum-safe cryptography capabilities to anticipate current and future threats.

IBM Power development and R&D teams collaborate with Red Hat to make sure Red Hat tools such as Advance Cluster Management and Advance Cluster Security support IBM Power and run efficiently on the IBM Power technology. For more on Red Hat on IBM Power, visit community.ibm.com/community/user/powerdeveloper/home.

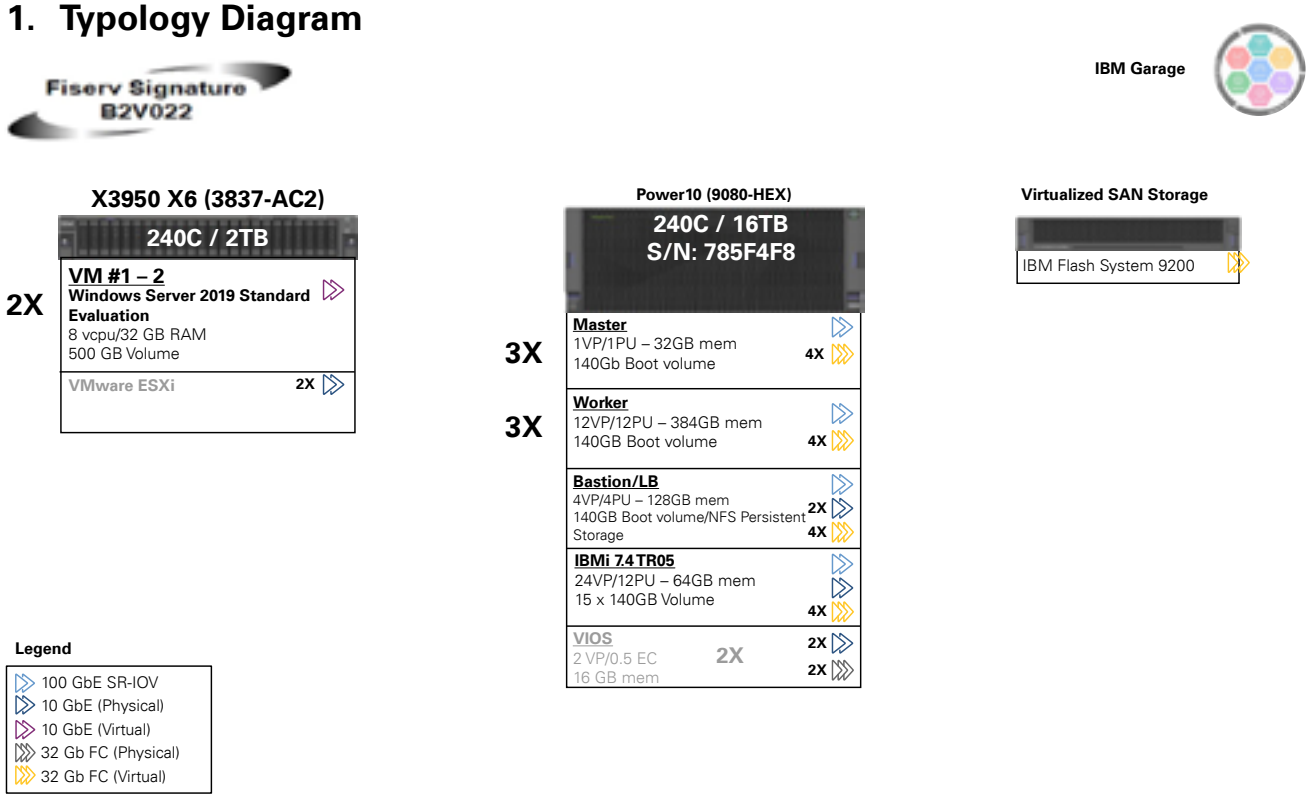
Environment Overview

Topology

One Power10 Model 9080 Hex was used for all the LPARs for the Master and Worker Nodes which ran Microservices for Signature and the IBM i used for Signature.

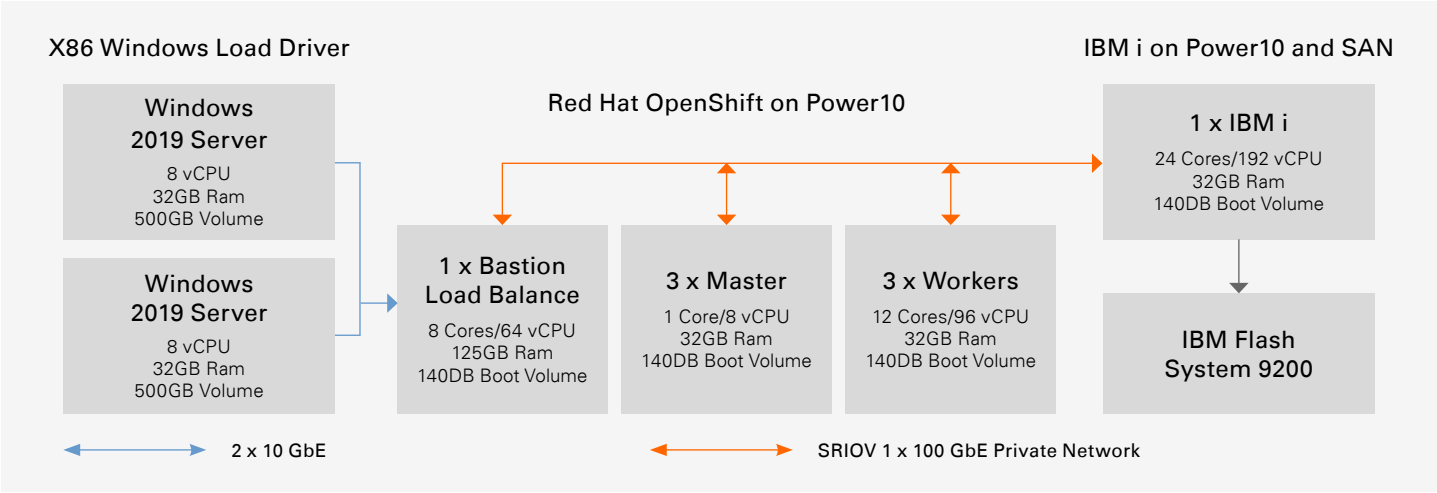
Additionally, a Bastion Host/Load Balancer was provided by an additional LPAR running on the same Power10.

1. Typology Diagram



Network Topology

The Red Hat OpenShift Cluster and IBM i shared a 100GbE network. Load was generated over a 10GbE network.



Cluster Details

Red Hat OpenShift (on IBM Power)	
Version	4.11.39
No. Worker Nodes	3
Processor Architecture	ppc64le
Worker Node – Cores	12 Cores/96 vCPU
Worker Memory	384GB
OS	Red Hat Core OS 4.11
Disk	140 GB
Proxy	HA Proxy

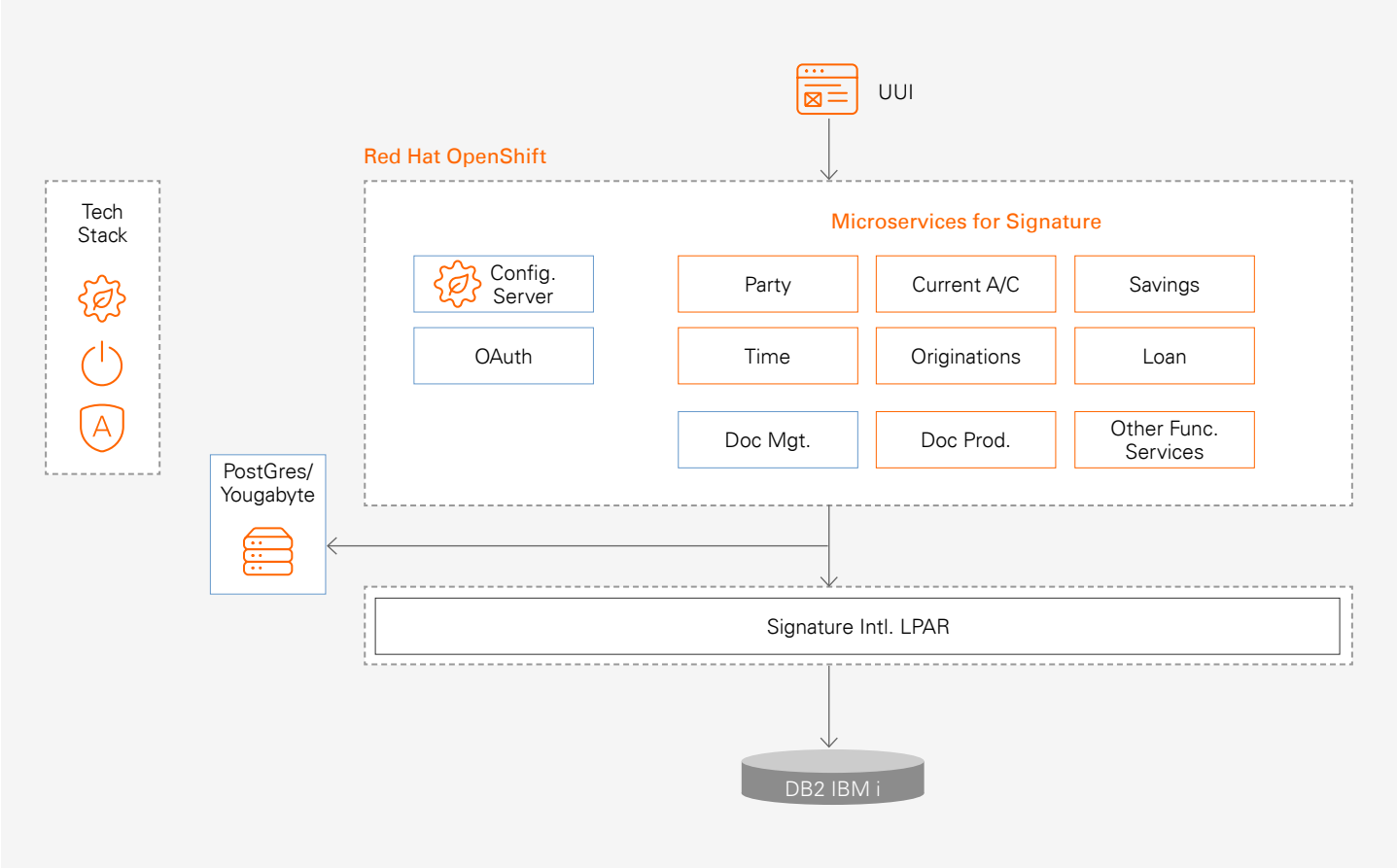
Containers Details	
Base Image	UBI and Operating System version as Red Hat Enterprise Linux release 8.1 (Ootpa) – 64bit
JDK	OpenJDK Runtime Environment (build 1.8.0_242-b08)

Power10 9080 Hex/240 Cores/16TB Ram			
Node	Cores	Memory	OS
Master 1	1	32GB	Core OS 4.11
Master 2	1	32GB	Core OS 4.11
Master 3	1	32GB	Core OS 4.11
Worker 1	12	384GB	Core OS 4.11
Worker 2	12	384GB	Core OS 4.11
Worker 3	12	384GB	Core OS 4.11
Bastion/Load Balancer	4	128GB	RHEL 8.6
IBM i LPAR	12	64GB	IBM I 7.4 + TL05
Total	55 (23% of available cores)	1.3TB (12% available RAM)	

IBM i

IBM i 9080 HEX	
Model	9080
Processor CPW	264K
Memory	64GB
OS	V7R4
Disk	2TB
Cores	12 Cores / 96 vCPU

Architecture of Signature and Microservices for Signature



For performance testing we tested Microservices for Signature and Signature. Load was generated from virtual machines using JMeter and sent to the REST APIs. A select set of the REST APIs was used, providing a mix of service requests in order to generate the load.

Test Configuration

Test Strategy

Services

The testing strategy was to determine the maximum throughput we could support. Ideally keeping the CPU utilization at around 50%–60%, thus providing room for spikes in load.

Threads

We use threads to maximize the load. The number of threads chosen is based on the highest throughput we can achieve.

Think Time

There is no think time built into the requests, so messages are continuously sent on each thread.

Test Message Set

For each test, the same transaction set was run. The transaction set used was:

Transaction Mix
App Taking Inquiry
Application Inquiry
Application Loan Payment Calc
Credit Line Summary
Current Account Summary
Current Statement History
Customer Account List
Customer Address List
Customer Basic Info Modification
Customer Basic Information
Customer Search
Employment Information
Loan Account Summary
Loan History
Time Account Summary
Time History

Number of Services (Pods/Containers) Copies Running

This table details the number of copies of the services used for the performance test:

Service	Copies
Deposit	6
Current	6
Customer Position	6
Loan	9
Party Data Management	9
Auth	1
Data	3
Host	1
Config Server	1

Test Results

The following sections will present the performance test results for each of the scenarios. In each section the following information will be used:

- Statistics table showing statistics for the message set run
- Load – CPU utilisation across the cluster and on IBM i
- Throughput for each message in seconds
- Total throughput, expressed as the total number of transactions per second
- Response times
- Threads
- Latency

Results Using Three Workers Nodes

Statistics

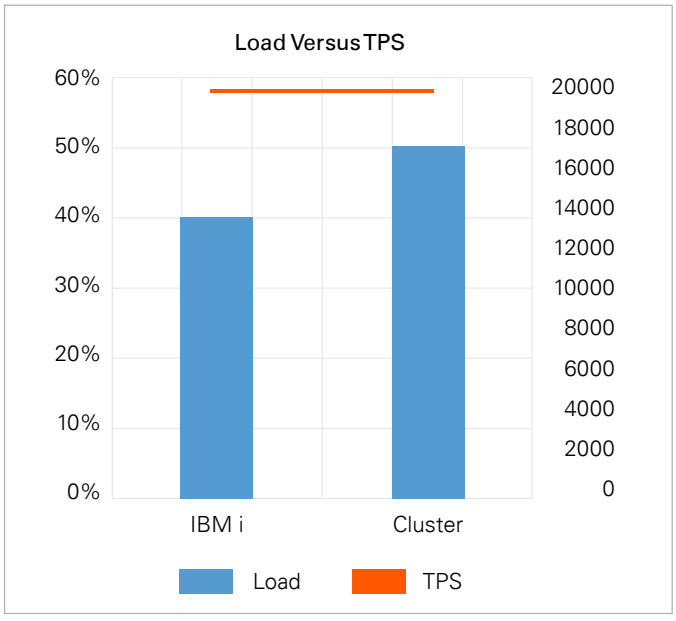
As the statistics show, we were able to reach a TPS of 19,000 overall. Each of the services TPS is also depicted and, on average, we recorded 2500 TPS for most services, and some were 750 TPS.

Statistics

Request	Executions			Response Times (ms)							Throughput	Network (KB/Sec)	
Label	#Samples	FAIL	Error %	Average	Min	Max	Median	90th pct	95th pct	99th pct	Transactions/s	Received	Sent
Total	**5727276**	**0**	**0.00%**	**39.62**	**2**	**2042**	**19.00**	**76.00**	**95.00**	**169.00**	**19067.21**	**127948.81**	**24974.87**
Credit line Summary	767295	0	0.00%	11.88	2	916	12.00	21.00	26.00	42.00	2562.92	7248.11	3373.85
Current Account Summary	767789	0	0.00%	105.49	3	2042	85.00	147.00	177.00	418.99	2556.70	5757.99	3246.57
Current Statement History	767435	0	0.00%	25.90	2	1921	16.00	65.00	77.00	692.99	2560.82	11923.68	3446.87
Customer Account List	224536	0	0.00%	75.41	5	1913	69.00	101.00	114.00	364.97	749.51	3985.75	950.07
Customer Address List	224490	0	0.00%	63.93	4	1562	59.00	80.00	90.00	308.97	749.52	1450.86	1032.05
Customer basic information	224584	0	0.00%	67.00	5	1649	61.00	82.00	92.00	358.99	749.53	3402.52	973.38
Customer Search	224642	0	0.00%	67.17	4	1911	60.00	82.00	92.00	344.99	748.09	3250.41	985.30
Employment Information	224435	0	0.00%	65.21	4	1641	60.00	81.00	90.00	313.99	749.41	1373.28	981.41
Loan History	767327	0	0.00%	21.46	2	964	23.00	37.00	42.00	58.00	2562.38	57149.34	3389.64
Time Account Summary	767383	0	0.00%	12.90	2	1341	12.00	23.00	32.00	70.00	2561.30	7698.21	3241.87
Time History	767360	0	0.00%	18.92	2	1234	18.00	30.00	38.00	74.00	2561.83	25052.23	3412.67

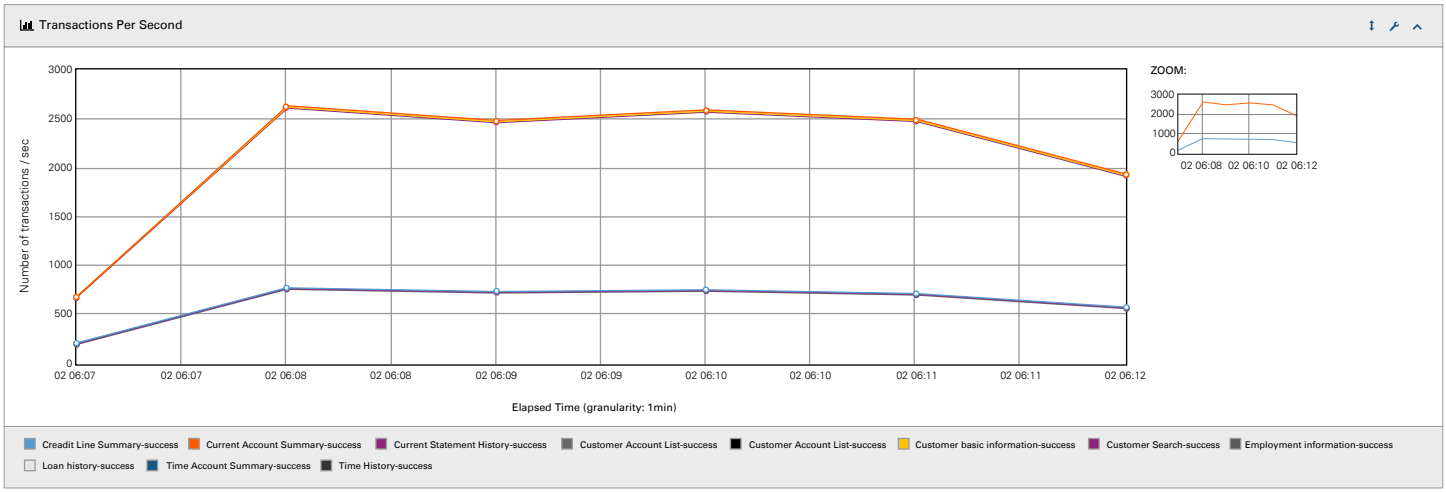
Load

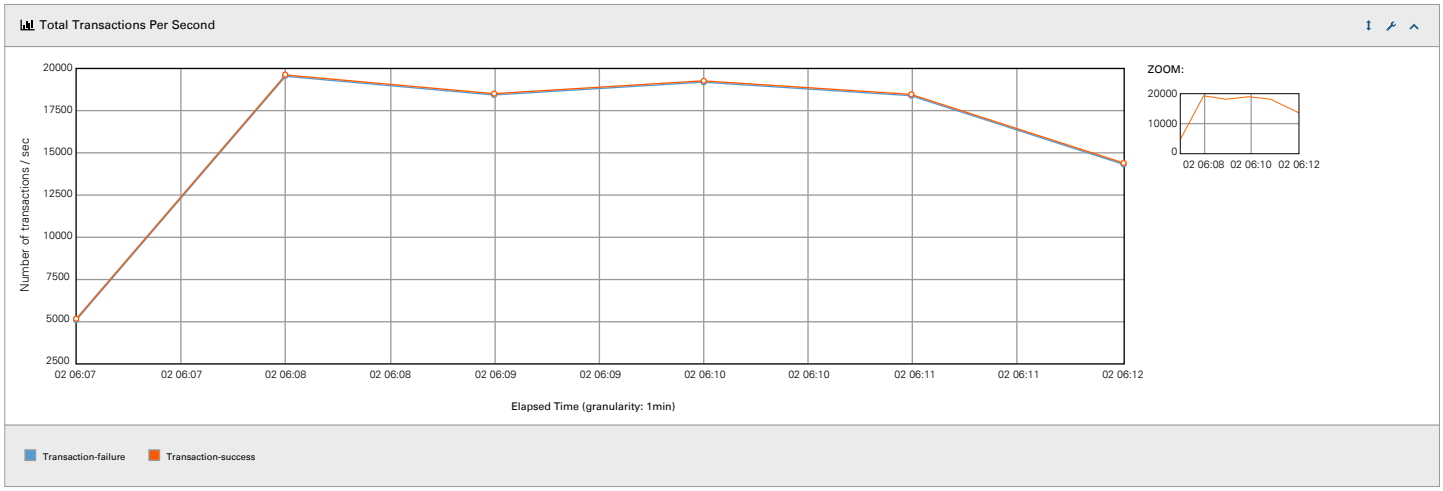
Target load was expected to be between 50 and 60%. However, we achieved the TPS with a load of 40 to 50%.



TPS per Service and Overall Total TPS

The next two graphs show the individual TPS per service and the overall TPS.

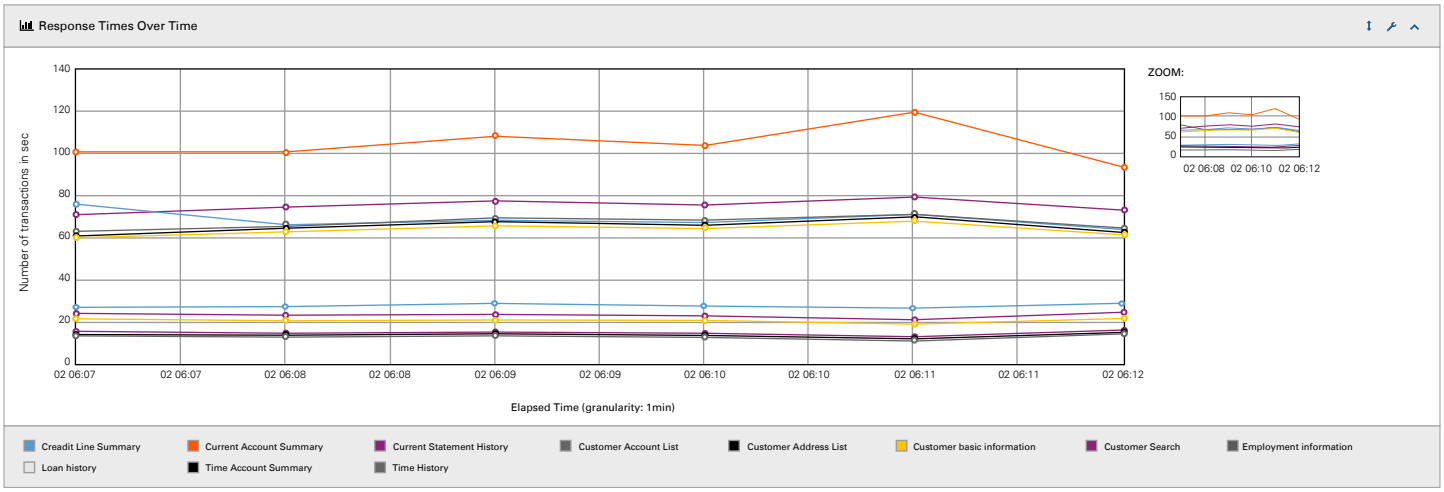




Response Times

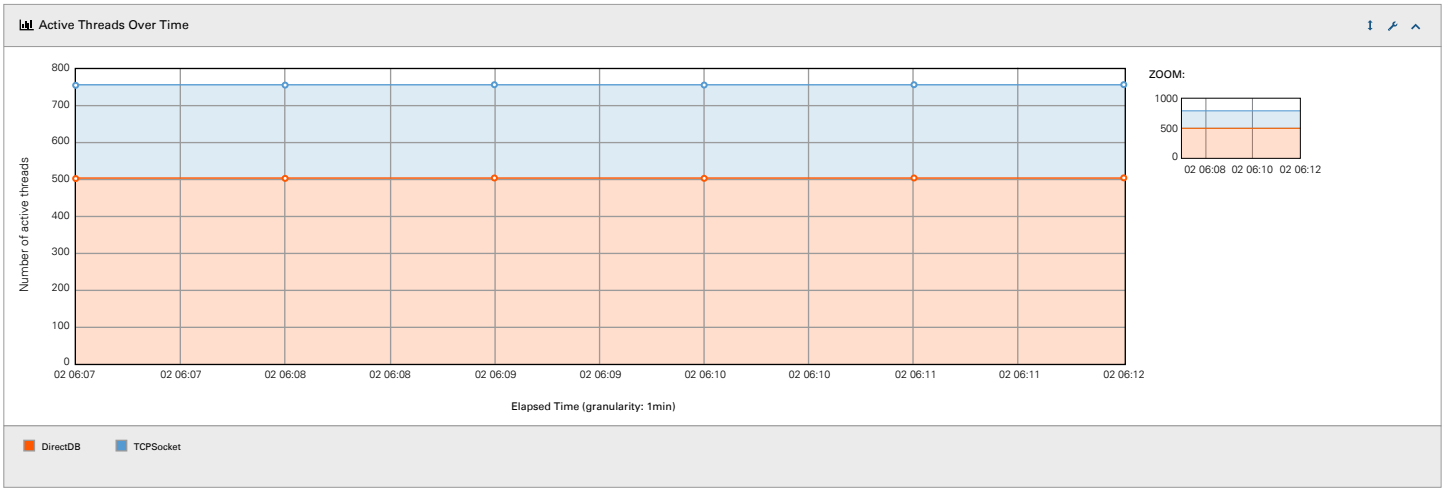
Response times were very good with the complex service, account summary, running at 100ms.

The average response time across the services is between 60 and 80ms.



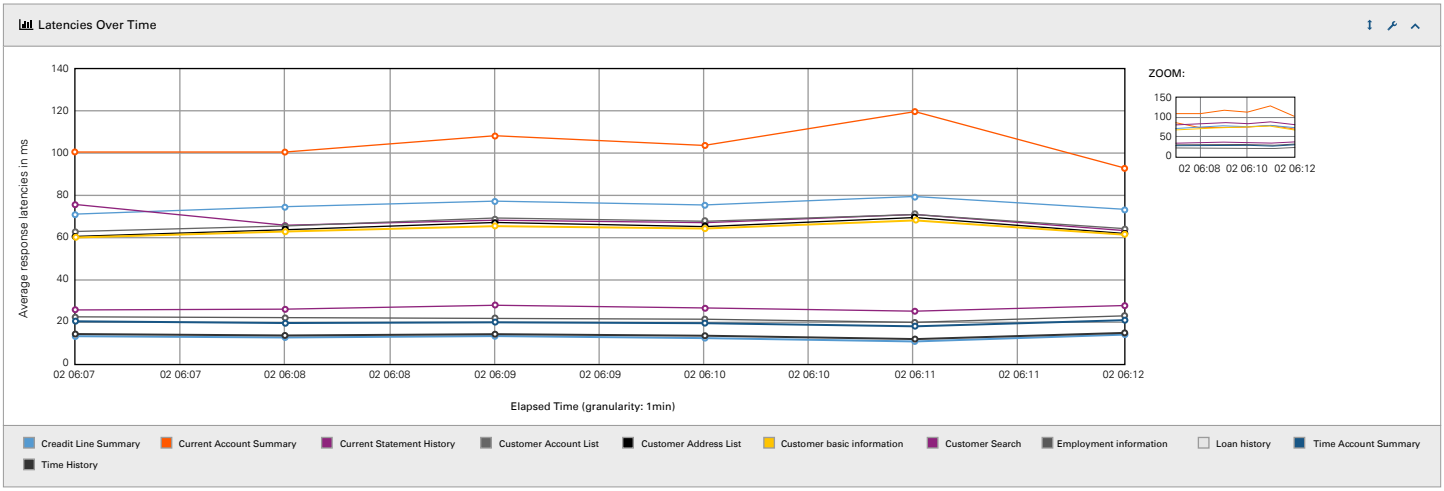
Threads

There are two types of Microservices for Signature – those that are process server based and those that are database based using JDBC. We ran 500 database threads and 256 process server threads to achieve the load.



Latency

Latency wasn't really a concern as we can see that response times (see response graph below) and latency are pretty much in line. This was to be expected, given that we ran all of the testing within the same lab.



The IBM Power of Collaboration

Fiserv and IBM continue to collaborate on performance and optimization of its latest technologies in order to deliver superior solutions to our commercial and retail-focused banking clients. Because of this ongoing collaboration, your organization will be able to introduce new services, deliver value and meet the evolving demand for the hyper-personalization of customer experiences in banking, today and in the future.

Connect With Us

For more information about Signature:

800-872-7882

getsolutions@fiserv.com

fiserv.com

Fiserv is driving innovation in Payments, Processing Services, Risk & Compliance, Customer & Channel Management and Insights & Optimization. Our solutions help clients deliver financial services at the speed of life to enhance the way people live and work today.

Visit **fiserv.com** to learn more.

Fiserv, Inc.
600 Vel R. Phillips Avenue
Milwaukee, WI 53203
fiserv.com

fiserv.