

SAP Loans Management on IBM System z and DB2 for z/OS

Performance and Scalability



Contents

1	Preface.....	6
1.1	Document Scope	6
1.2	Main authors of the document.....	6
1.3	With gratitude and acknowledgment to our sponsors.....	6
1.4	Project Team	7
2	Executive Summary	7
3	Introduction	9
3.1	SAP Solutions for Banking.....	9
3.2	SAP for Banking on System z Reference Architecture	10
4	Motivation and Test Objectives	11
5	Tested Solution.....	12
5.1	Solution Overview.....	12
5.2	Test scope	13
5.3	Process Step Description	14
5.4	IBM Infrastructure Configuration	16
	Database Server.....	16
	Application Servers	17
	Storage	17
	Network	17
	Infrastructure Architecture	17
6	Results.....	18
6.1	Performance test results.....	18
6.2	DB Server Utilization.....	23
6.3	Application Server Utilization	23
7	Testing Methodology	25
7.1	Test approach	25
7.2	Data creation.....	26
7.3	Running the test.....	30
7.4	Measurement methodology & tools	31

8	Test set-up.....	34
8.1	Application configuration	34
8.2	Application server set-up.....	35
8.3	Database Setup	37
8.4	End-of-day processing design.....	37
8.5	Application workload details	38
8.6	Database—Large Table Analysis	42
9	Customer-specific end-of-day runtime projection	43
10	Summary and Outlook	45
11	Appendix	46
11.1	Result details per testing stage.....	46
11.2	Elapsed processing time per process step	48
11.3	Application Server on IBM Power System - Result details.....	51
11.4	MIS4SAP—Monitors used.....	52
11.5	Software levels	53
11.6	SAP Notes implemented	54
11.7	Bufferpool Sizes	56
11.8	Database Bufferpool Isolation.....	58
11.9	BaS Profile Parameters	65
11.9.1	Default Profile	65
11.9.2	Central Instance Profile.....	67
11.9.3	Enqueue Profile	69
11.9.4	Dialogue Profile	70
11.10	FI-CAX Profile Parameters	70
11.10.1	DEFAULT Profile	70
11.10.2	Central Instance Profile.....	73
11.10.3	Enqueue Profile	74
11.10.4	Dialogue Profile	76
12	Terms and definitions.....	76
13	References to related material	77
13.1	Latest SAP on System z Banking Customer References	77
13.2	Documents	78
13.3	Additional links	78

List of Figures

Figure 1. Loans processing flow	14
Figure 2. Technical view of process steps	16
Figure 3. Test landscape installed at IBM Poughkeepsie, NY, USA.....	18
Figure 4. Total elapsed processing time	19
Figure 5. Million of loans/hour processed rate.....	20
Figure 6. Allocation of runtime on process steps.....	21
Figure 7. Database Server CPU % Utilization for 8 million loan.....	23
Figure 8. Application Server CPU % Utilization for 8 million loans.....	24
Figure 9. Scaling out by parallelization	25
Figure 10. Scaling out by increasing data volume	25
Figure 11. Aging of the loans	27
Figure 12. Data volumes and aging of the loans for the four test stages	28
Figure 13. Main performance measurements when running the tests.....	31
Figure 14. MIS4SAP monitoring flow	32
Figure 15. MIS4SAP automated monitoring start/stop and time profile	33
Figure 16. MIS4SAP communication with AIX and z/OS	34
Figure 17. SAP Loans Management solution landscape	35
Figure 18. Billing Elapsed Times	48
Figure 19. Invoicing Elapsed Times.....	48
Figure 20. Payment Distribution Elapsed Times	49
Figure 21. Outbound Interface Elapsed Times	49
Figure 22. Accruals Elapsed Times.....	50
Figure 23. Settlement Elapsed Times	50
Figure 24. Capital Calculations Elapsed Times.....	50

List of Tables

Table 1. Throughput of loans per hour	22
Table 2. Runtime in hours per process step.....	22
Table 3. Housekeeping jobs used.....	30
Table 4. Product features and attributes	35
Table 5. Options for EOD runtime optimization	37
Table 6. Billing runs with different numbers of parallel processes	38
Table 7. Optimal number of parallel processes for the different steps.....	39
Table 8. Determination of the optimal package size for the billing run.....	40
Table 9. Optimal package size for the different steps	40
Table 10. Throughput per product for different steps	41
Table 11. Largest banking tables after 8 million created and aged loans	42
Table 12. Projecting performance results for different process steps.....	44
Table 13. Result details for 8 million loan accounts	46
Table 14. Result details for 4 million loan accounts	46
Table 15. Result details for 2 million loan accounts	47
Table 16. Result details for 0.5 million loan accounts	47
Table 17. Application Server Hardware Consumption	51
Table 18. Monitoring tools used.....	52
Table 19. Software components for the Banking Services system	53
Table 20. Components for FI-CAX system.....	54
Table 21. Database components.....	54
Table 22. SAP Notes implemented for the Banking Services system	55
Table 23. SAP Notes implemented for the FI-CAX system	56
Table 24. Bufferpool sizes for FI-CAX and BaS system	56
Table 25. Database bufferpool isolation for FI-CAX.....	58
Table 26. Database bufferpool isolation for BaS system	60
Table 27. Largest FI-CAX tables after 8 million created and aged loans	63
Table 28. Terms and definitions	76

1 Preface

1.1 Document Scope

This joint IBM® and SAP® whitepaper is a technical report of the results of the IBM and SAP co-innovation project optimizing the SAP Loans Management for Banking Services solution with help of an IBM enterprise computing infrastructure. All software, middleware and hardware parts of the tested solution are commercially available. The whitepaper documents how we, the IBM and SAP project team, tested, analyzed and refined the solution for an optimized behavior. This project and document delivers proof points how to reach specific levels of performance under realistic conditions.

Aligning customer requirements in SAP Loans Management workload scaling with the performance behaviors of IBM enterprise systems for a right-sized set-up is a major driving force behind both companies' investment. This document is intended for architects and implementations teams to help them design and implement an SAP Banking solution on the SAP reference architecture for IBM System z®.

1.2 Main authors of the document

- Rose Manz, IBM Senior Programmer on SAP Performance Test
- Rolf Müller, IBM Senior IT Specialist
- Carsten Siegler, IBM, Project Lead
- Marc-Stephan Tauchert, IBM Performance and Benchmark Specialist
- Dr. Christian M. Müller, SAP AG, Project Lead
- Sonja Bahn, SAP AG Application Performance
- Marcus Ronellenfitsch, SAP AG Application Expert
- Oliver Wiebracht, SAP AG Application Expert

1.3 With gratitude and acknowledgment to our sponsors

We thank those who have given the support and vision that allow us to provide this whitepaper in support of our joint customer needs.

- Dr. Paul Lekkas, IBM Distinguished Engineer
- Klaus Scholten, IBM Program Manager for SAP on IBM System
- Diemo Wilhelm, SAP AG, Global Head of FSI Active Global Support
- Jürgen Hofmann, SAP AG Solution Manager
- Antonio Palacin, Director IBM SAP International Competence Center

1.4 Project Team

This project was run by a joint IBM and SAP team, combining the SAP application skills with the IBM platform and solution skills.

Role/Area Responsible	Person Responsible	Company
Project Lead IBM	Carsten Siegler	IBM Germany
IBM Distinguished Engineer	Paul Lekkas	IBM Germany
Project Steering and Management	Klaus Scholten	IBM Germany
Technology – Database DB2 and z/OS	Rose Manz	IBM Poughkeepsie
Technology – Database DB2	Seewah Chan	IBM Poughkeepsie
Technology – AIX	Paul Delessio	IBM Poughkeepsie
Technology – IT Specialist	Rolf Müller	IBM Germany
Performance and Benchmark Tools	Marc-Stephan Tauchert	IBM Germany
Project Lead SAP	Dr. Christian M. Müller	SAP AG
Performance Expert – Application	Sonja Bahn	SAP AG
Application Expert	Marcus Ronellenfitsch	SAP AG
Application Expert	Oliver Wiebracht	SAP AG
Solution Manager	Jürgen Hofmann	SAP AG

2 Executive Summary

The banking industry is in an ongoing transformation process to differentiate themselves by tailoring their products and services, and optimizing the efficiency of their operations. SAP Loans Management for Banking Services is a core banking software for retail and commercial banking. The software delivers flexibility and fast time-to-market for new lending products and services. Another request of these banks in order to attract customers and keep their loyalty is the stable availability of their services. The high acceptance of mobile and internet delivery channels drive the industry towards a 24/7 continuous operation.

IBM and SAP collaborated on a major performance and scalability test to deliver an optimized reference architecture for SAP Loans Management. The tests were performed in an IBM enterprise systems environment with IBM System z and DB2® for z/OS® as SAP database server. IBM Power 7 servers were used for the SAP application workload and IBM DS8000® storage for SAP data at rest and backup copies. IBM System z and DB2 for z/OS technologies offer advanced high-availability components built in as standard.

The following customer statement explains well our clear motivation for this IBM and SAP investment, particularly in the SAP Loans Management solution on System z: “A major IT challenge that I face every day is to ensure that a million loan accounts are processed within a given time frame.” A further objective of our project was the optimization and sustainability of customer investments of the required infrastructure. We could answer these questions through extensive test series to validate end-to-end workload performance for high volumes.

The results exceeded our expectations for the business relevant end-of-day (EOD) processing time, comprising all the performance impacting business process steps.

- The tests prove the performance and scalability of the solution. SAP Loans Management 8.0 can process up to 8 million loan accounts in less than 6.5 hours, or at a rate of 49 minutes per 1 million accounts.
- The total SAP data in these tests increased from 0.7 TB to 4.4 TB.
- We experienced a linear scaling of the setup when the workload increased from 0.5 to 8.0 million loans. With a peak utilization of 88% of the deployed application servers the database system never went into saturation. The infrastructure under test was always stable, thanks to the efficient resource management of the highly virtualized IBM enterprise technologies.
- Over the course of the project the SAP Loans Management application was continuously refined, so that we achieved a total improvement of 18% for the workload throughput with the optimized software.

The individual steps show how the database server and application server load depend on the type of application, underlining the importance of conducting performance testing.

All this was successfully accomplished by a globally dispersed, small team from both IBM and SAP that performed the tests and optimized the software. This joint project continues the successful collaboration of the two companies in the area of core banking transformation. Our joint efforts drive towards end-to-end offerings. Our attempts align SAP banking solutions with integrated architectural components, optimized infrastructure and collaborative consultation.

This project demonstrates that SAP Loans Management and IBM enterprise systems can help banks, improve efficiency back-office operations, increase their ability to grow, and become more resilient.

3 Introduction

This whitepaper documents the results for the performance tests and improvements of SAP Loans Management for Banking Services with DB2 for z/OS on the IBM System z platform. The main goal of this test was to demonstrate the scalability of SAP Loans Management in an IBM enterprise system computing environment and to optimize the workload performance of the overall solution.

3.1 SAP Solutions for Banking

SAP offers a rich and growing application portfolio for retail and commercial banking. Global economy developments force banks to further transform their business to become more client focused and service oriented. To attract everyday customers, banks need fast and real-time processes that are also cost efficient. Rather than redeveloping and modernizing their home grown IT solutions, SAP Solutions for Banking help banks to speed up the time to market in this transformation.

SAP has banking specific solutions for

- Retail Banking Operations—including Retail Lending services
- Commercial Banking Operations—including Commercial Lending services
- Risk and Compliance management
- Sales and Service
- Multichannel delivery

The SAP Loans Management application in combination with other SAP applications makes up the SAP solutions for Retail and Commercial Lending. In addition, SAP offers many SAP Business Suite solutions customized to manage a bank.

SAP Loans Management

SAP Loans Management for Banking Services is part of the SAP Core Banking applications family. The product implements administration services for a variety of personal and business lending products. The SAP software manages and monitors the entire lifecycle of a loan contract, including origination activities, all subsequent transactions, agreements, statements, and more.

SAP Loans Management delivers

- **Loans Servicing**—Administration of loan accounts across their entire lifecycle.
- **Bundled Product Services**—Facilitate the definition of bundled banking products for enabling pricing and shared features across different types of products. Examples are global credit lines or relationship based pricing, e.g. a “one account approach”.
- **Due Item Management**—Automatic creation and clearing of due items for a related loan account.
- **Funding Management**—Enables fund managers to control revenues, expenditures, and funding transactions across the organization.
- **Collateral Management**—Supports the full process for secured loan origination, collateral monitoring and administration, and liquidation.

Loans management comprises online transactional and batch operation modes. During the bank working hours online transactions are executed by the end-user, either banking staff or agents, or via the delivery channels. At predetermined times, banks need to close their books for the day, or issue reports, and perform an end-of-day summary calculation and reporting. This so called EOD processing needs to happen in a defined time window, often determined by legal constraints. For Loans Management this is a batch type process and its completion within the given timeframe is crucial for bank operations. Customers and consultants need to understand the behavior of these batch processes in order to design the required IT infrastructure accordingly.

This whitepaper documents and discusses the tested SAP Loans Management 8.0 on IBM infrastructure solution, describing the results, testing methodology, test set-up, and performance optimization details for end-of-day loans processing in SAP Loans Management.

3.2 SAP for Banking on System z Reference Architecture

SAP for Banking solutions are usually the core of a bank’s business processes. Therefore, these solutions pose very demanding requirements on the underlying infrastructure, which is based on the interplay of servers, operating system, database, and storage.

Due to the nature of banking itself, two major requirements of a bank's IT infrastructure are high availability and data recovery. For example, when a core banking application is not operational, the whole bank is affected, and a bank never can afford to lose or compromise customer or account data.

The SAP for Banking on System z reference architecture focuses on the system infrastructure for SAP for Banking. It is a guideline how to best implement and run SAP for Banking solutions on IBM Enterprise systems, in particular IBM System z, to meet our customers' stringent infrastructure needs. Core infrastructure products and technologies that are part of the reference architecture—and also were used in this project—are:

- IBM System z
- DB2 for z/OS
- IBM Power Systems™
- IBM Disk Storage DS8000® with fast replication services like Metro Global Mirror and FlashCopy®

Additionally recommended for a productive and highly available system are, among others:

- IBM Parallel Sysplex® and Geographically Dispersed Parallel Sysplex™ (GDPS®)
- SAP Central Services with Parallel Sysplex support
- IBM Tivoli® System Automation for z/OS

This whitepaper also documents how SAP Loans Management was implemented, tested, and optimized in a System z Reference Architecture environment. Please find a link to the SAP reference architecture documentation under 13.2 "Documents".

4 Motivation and Test Objectives

SAP Loans Management for Banking Services is used by mid-size retail banks as well as by large banks. In most implementations the system performance and workload throughput is a crucial factor for the success. Therefore banks strongly demand from software vendors and hardware suppliers to provide information that helps make the right decisions how to design and build an

efficient implementation. In particular these customers want help how to size their systems, about tuning options, and how to set up the environment for EOD processing. We would like to provide some of this information based on our test results. At the same time we proved and documented the performance capabilities of SAP Loans Management running on an IBM enterprise system infrastructure.

The objectives in detail were:

- Demonstrate scalability of the SAP Loans Management solution on an IBM platform
- Information how to optimize SAP Loans Management performance on the given IBM infrastructure
- Demonstrate that large volumes of loans can be processed in a reasonable time window during peak end-of-day processing, meeting industry requirements
- Provide information about optimal configuration settings, infrastructure settings, and coding improvements
- Obtain input for sizing rules and practices.

5 Tested Solution

5.1 Solution Overview

SAP Loans Management for Banking Services is a transactional platform to manage different kinds of loan products. It covers for example simple consumer loans, collateral based mortgage loans, and—in combination with SAP Master Contract Management—also product bundles to offer customers financial services that span different loan types.

The solution processes the typical life-cycle of a loan, starting with account creation, disbursement, installment processing, and account closure. Configurable business rules allow a high degree of process automation to avoid manual intervention. Additionally 13 Business Operations are available to manage unexpected events such as

- Payment Plan Change
- Renewal
- Payoff

The service orientated architecture (SOA) provides interfaces to standardize the integration of Loans Management into existing system landscapes. Complementary solutions such as SAP CRM, SAP NetWeaver Business Warehouse (BW), and SAP Bank Analyzer complete the end-to-end view for the processes account origination, reporting, and analytics.

5.2 Test scope

The test scope was the regular life-cycle of a loan account. The life-cycle comprises of mass processes that typically run at end-of-day. The batch type process steps are described in the following chapter. We focused on the execution of straight forward processing of high volumes. Loans were fully disbursed and installments were paid on time. Special processes such as late payments, reversals or back-dated changes were not tested. They are not expected to have a major performance impact as the processed volumes are small compared with the total loans volume.

Loans products and process are highly configurable. To consider different sets of system settings, we used different products with different settings in the test. For details see chapter 8.1 “Application configuration”.

Another characteristic of the test scope is the concentration on peak volumes. Usually loans in a customer portfolio have different due dates with periodical peaks at month-end or year end. In the tested portfolio all loan installments are due on the same day. This way a portfolio of 8 million loans creates an EOD processing peak of 8 million objects to be processed.

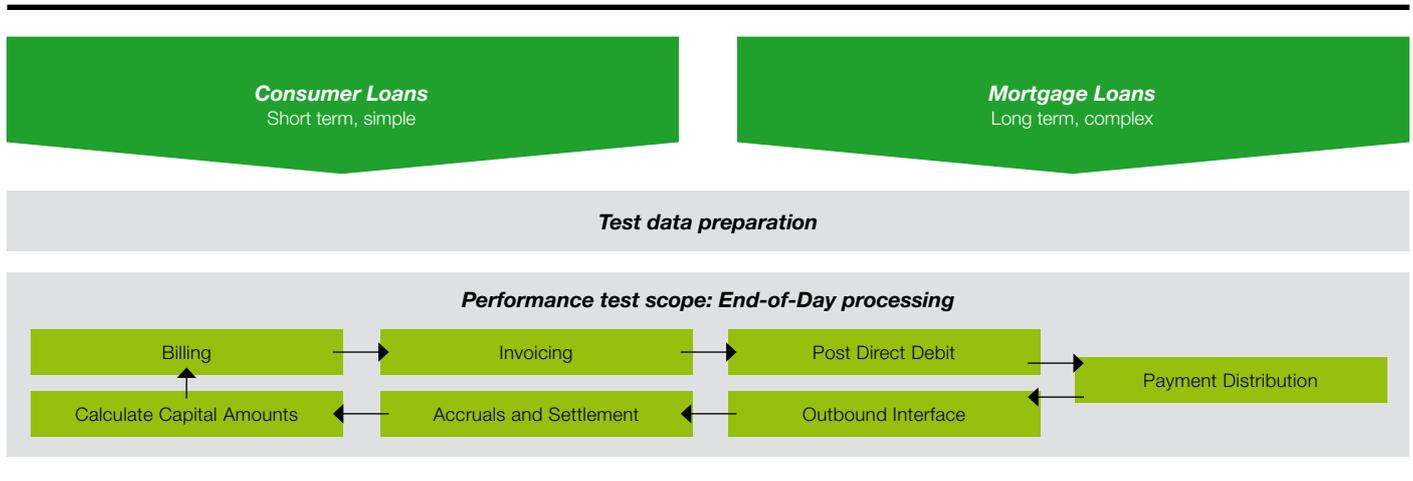
Out of scope:

Online processes that are usually issued individually during working hours, such as account creation, were not tested.

5.3 Process Step Description

In this section we briefly explain the meaning of each of the process steps from a business perspective. SAP Loans Management comprises two technical systems: Banking Services (BaS) and contract accounting (FI-CAX). Some processes are executed on Banking Services while others are executed on FI-CAX. Banking Services and FI-CAX are tightly integrated by default. These two systems were represented each by a unique SAP System ID (SID).

Figure 1. Loans processing flow



The process steps in detail are:

Billing

Billing generates information about due installments. The information is retrieved from the loan's cash flow and replicated to FI-CAX for further processing.

Invoicing

The billing information is used to create open items in FI-CAX. Optionally the customer receives a notification of the due receivables. The creation of the notification was not in scope for the performance test.

Post payment orders for direct debit

A payment can be either bank initiated (such as direct debit) or account holder initiated (money transfer, cash payment, etc.). Both result in a posting of an incoming payment in Banking Services. We used a custom report to generate incoming payments automatically. Therefore this report is not in scope of the performance test.

Payment distribution

The payment distribution process determines the usage of each payment for a specific loan. Possible usages are e.g. clearing open items, prepayments or extra repayments. The payment distribution process can be set to synchronous or asynchronous mode. We used the “synchronous” mode as it performed better in our system setup than the “asynchronous” mode.

Execute outbound interfaces (FPOITR)

The outbound interface transfers information about clearing of open items from FI-CAX to Banking Services.

Accruals

This process step calculates accrued interest in order to provide information to the General Ledger.

Settlement

The settlement process calculates and posts all loan related prices such as interest and charges.

Calculate capital amounts

In this step seven capital amounts are calculated for each loan. In a customer environment, the capitals would be extracted and loaded into SAP NetWeaver BW for reporting:

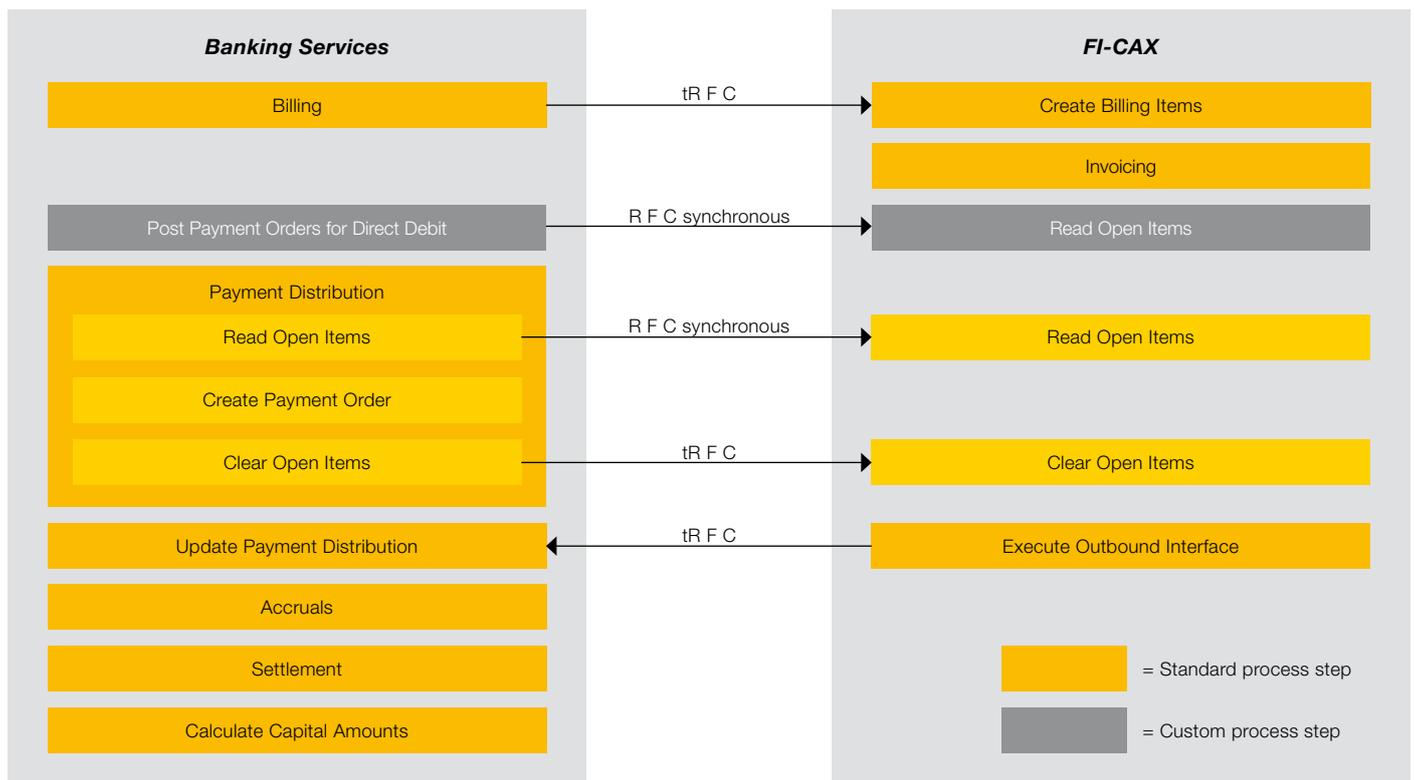
1. Outstanding interest
2. Effective capital
3. Outstanding charges
4. Repayment balance still due
5. Interest per day
6. Interest paid
7. Interest accrued but not settled

Between these steps, the posting dates are adjusted accordingly to simulate the passage of time.

The technical interaction between the Banking Services and the FI-CAX system is displayed in Figure 2. Note that there is communication between the two systems, namely in the billing, payment, payment distribution, and the execute outbound interface processes. The intersystem communication uses the

SAP Remote Function Call (RFC) method and protocols. Most interactions are of asynchronous nature (transactional RFC or tRFC), just a few are synchronous, where the sender waits for a response from the receiver. Sequence and type of interaction determine the critical path of the workflow.

Figure 2. Technical view of process steps



5.4 IBM Infrastructure Configuration

Database Server:

The tests were performed on an IBM zEnterprise® EC12 (zEC12) Class 2827-HA1 server with 101 5.5 GHz processors with a total of 1.6 TB memory installed. The test cycles utilized one dedicated LPAR (Logical partition) for DB2 for z/OS with 20 processors and 150GB memory. The LPAR connects to the DS8000 storage via high speed FICON channels. With two SIDs this LPAR hosted two DB2 for z/OS V10 instances, one for each SID. DB2 for z/OS uses data and index compression to cache as much as possible SAP data in memory. These two database server instances share the 20 zEC12 processors and the memory of the LPAR, competing for the resources. However, we restricted memory over-commitment of the LPAR to avoid memory swapping, as it would kill SAP performance.

Application Servers:

Two physical IBM Power enterprise servers were used in the test for SAP application serving; one was used for the SAP Central Services including the enqueue server. The other ran four LPARs hosting the SAP dialog instances, commonly called SAP application servers (SAP AS). The enqueue server ran on an IBM Power 740 with 16 3.5 GHz processor cores utilizing SMT and 256 GB memory. The application servers were run on one IBM Power 795 system with 256 4.0 GHz processors utilizing SMT and 4 TB memory. Each LPAR had eight SAP AS instances per SAP System ID (SID), providing a total of 48 SAP AS instances on this enterprise server.

Storage:

One dual frame IBM System Storage DS8800 2424-951/95E with 56 TB storage, 384 GB cache and 12 GB NVS (non volatile storage) were used for the two databases (1 per SID) and their associated backup copies. The database for FI-CAX used 20 3390 model 54 (virtual) volumes and was approximately 0.7 TB in size. The database for BaS used 96 3390 model 54 volumes and was approximately 4.4 TB in size, providing a total of 48 SAP AS instances on this enterprise server.

The active database logs were placed on an IBM System Storage DS8700 2421-941 with 26 TB storage, 112 GB cache and 4GB NVS. Each active log was 4 GB in size and striped across four volumes on four different ranks.

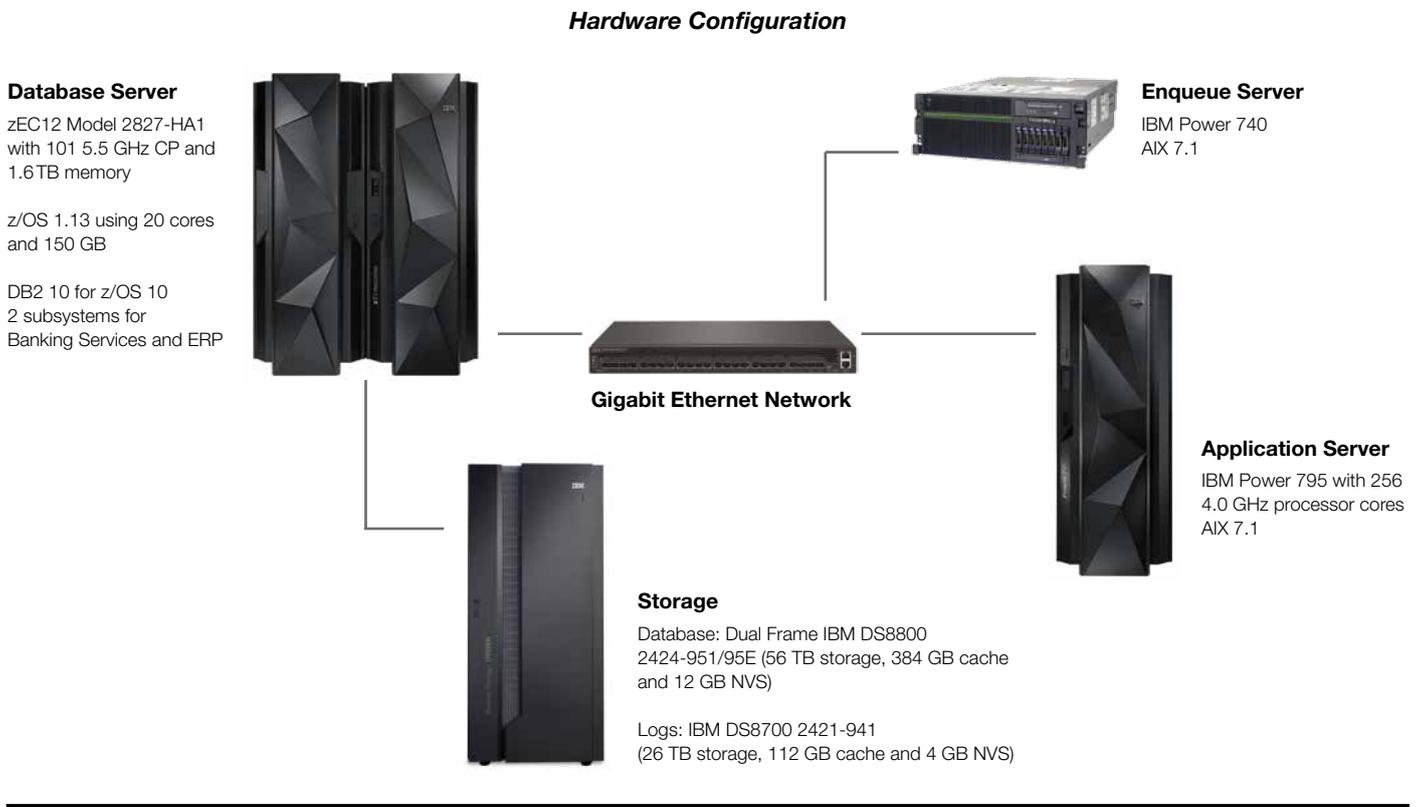
Network

A dedicated 10 GB Ethernet network was used for all connections. Each of the Power servers was connected via four 10 GB Ethernet adapters through a 10 GB Ethernet switch to the database server on System z, utilizing four OSA-Express4S adapters on the zEC12.

Infrastructure Architecture

Below is a conceptual view of the configuration that was used in the test environment.

Figure 3. Test landscape installed at IBM Poughkeepsie, NY, USA



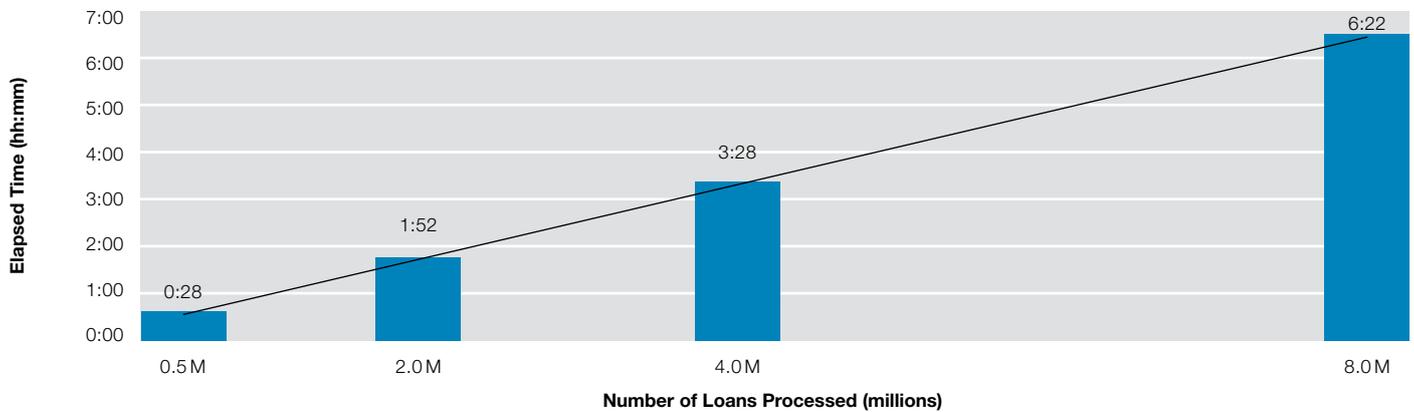
6 Results

During the course of this testing effort many test cycles were performed. Some tests were to familiarize the team with the environment and workload, others for application improvements and debugging. It is beyond the scope of this paper to discuss and show them all. The cycles selected for this paper represent the most performance optimized setups for the various loans database sizes and test stages.

6.1 Performance test results

The result of the performance test demonstrated that the SAP Loans Management for Banking Services on IBM zEnterprise System scales linearly as the number of loans increased from 500,000 to 8 million. In the following we show the summarized results, while each step in the cycle and how it scales is documented in the Appendix, on page 46.

Figure 4. Total elapsed processing time per cycle



Processing high volumes

Customers can manage portfolios of multi-million loan accounts on Loans Management running on IBM hardware. We could prove that a peak with **8 million loan accounts** is manageable in the EOD processing window.

Total processing time

Customers can run the EOD process with 500,000 loans in as little as **0.5 hours**. Typically the allowed time-window for EOD processing is significantly longer so there is room for resource requirement optimization for the daily processing. Peaks show up periodically (end-of-month, end-of-year, etc.) and therefore are predictable, permitting resource reallocations and temporary operations adjustments.

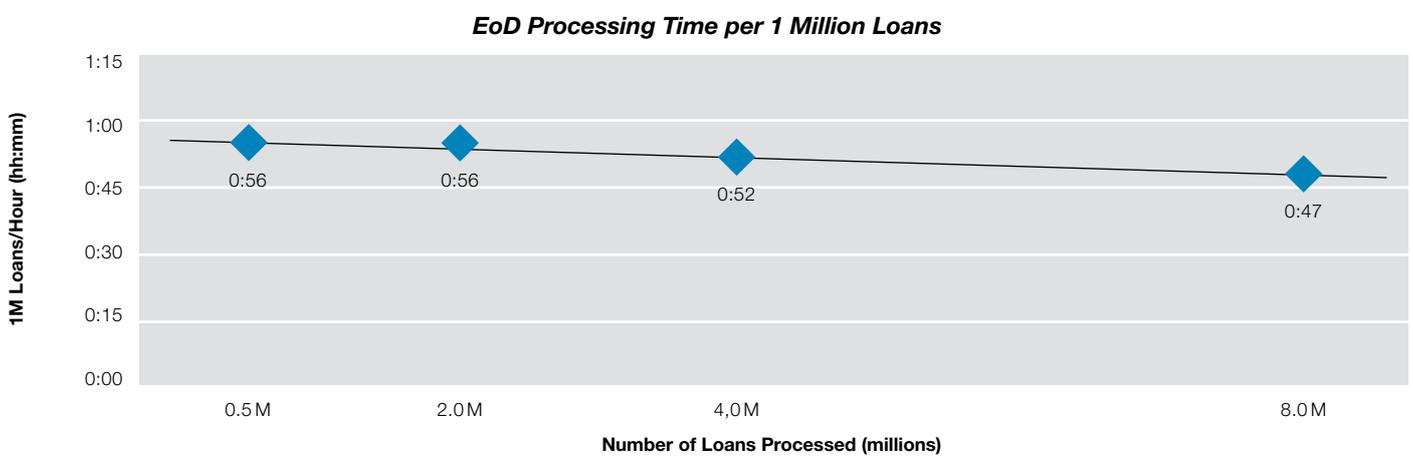
The total processing time for 8 million loan accounts was **6.5 hours** which equals a throughput of **1.25 million loan accounts per hour**.

Total processing time represents all EOD processing steps that must follow a sequential order. This we call the critical path of the EOD process. We are aware that EOD processing in a real customer scenario would need to invoke more processing steps than what has been in scope for the performance tests. However, not all processing steps are necessarily on the critical path and they can potentially run in parallel, given that the computing capacity is available. In chapter 9 we also show that the assumed EOD processing has significant potential to be faster when adjusted to a custom-specific design.

Scalability

The performance test has proven the overall EOD processing chain to be scalable. Over the course of the project and test cycles, we achieved an 18% better scaling of the throughput, from 58 minutes per one million loan accounts (based on a 500 k accounts cycle) to 49 minutes per one million loan accounts (based on the 8 M accounts cycle).

Figure 5. Million of loans/hour processed rate



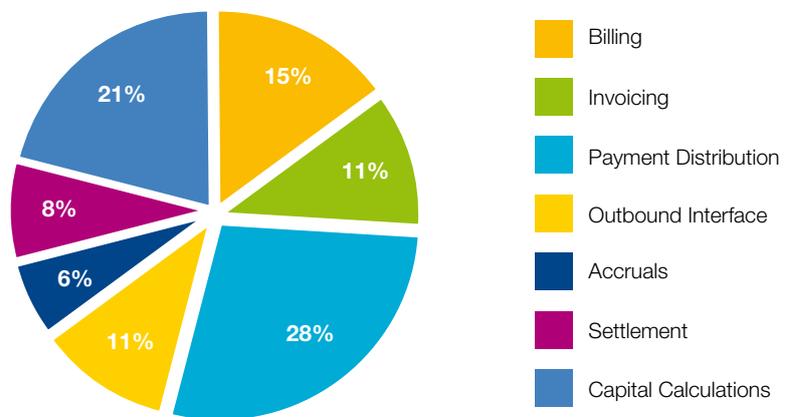
This unexpected reduction of the processing time with a growing number of loan accounts in a testing stage can be explained by performance improvements that we applied between cycles of the different stages. We started with 500,000 loan accounts, and then we immediately used the information collected for tuning and continuous improvement of the EOD processing for the following test stages with 2, 4 and 8 million loan accounts accordingly. These solution adaptations are available for customers of SAP Loans Management 8.0.

Allocation of runtime across different process steps

The runtimes of the individual process steps execute in an uneven spread. Process steps that generate a lot of transactional data such as Payment Distribution or that require cash-flow calculations such as Capital Calculation and Billing consume more time than other processes.

Figure 6. Allocation of runtime on process steps

Runtime allocation per process steps for 8 million loans measurement



A more detailed breakdown provides the throughput of loans per hour and the runtime per process step. This information can be used to apply the results to a specific customer scenario. For more details per testing stage see 11 Appendix, on page 46.

Table 1. Throughput of loans per hour

Throughput of loans per hour	Volume processed			
	500,000	2,000,000	4,000,000	8,000,000
Billing	6,185,567	6,360,424	6,929,740	8,172,531
Invoicing	12,857,143	12,996,390	15,141,956	11,538,462
Payment distribution	4,128,440	3,973,510	3,708,473	4,525,456
Outbound execution	8,955,224	6,255,430	9,479,921	10,946,408
Accrual	13,043,478	20,869,565	21,460,507	21,951,220
Settlement	11,042,945	14,035,088	14,472,362	16,116,396
Capital calculation	5,504,587	5,916,187	5,970,149	5,928,366

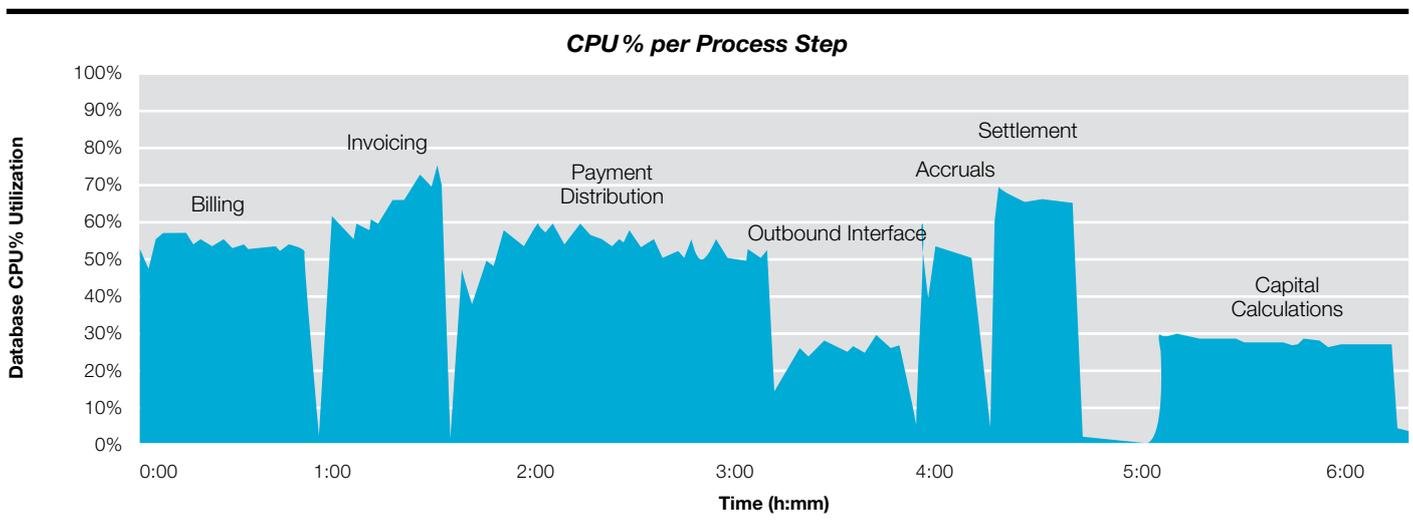
Table 2. Runtime in hours per process step

Process step	Volume processed			
	500,000	2,000,000	4,000,000	8,000,000
Billing	0:04	0:18	0:34	0:58
Invoicing	0:02	0:09	0:15	0:41
Payment distribution	0:07	0:30	1:04	1:46
Outbound execution	0:03	0:19	0:25	0:43
Accrual	0:02	0:05	0:11	0:21
Settlement	0:02	0:08	0:16	0:29
Capital calculation	0:05	0:20	0:40	1:20
Total runtime	0:28	1:52	3:28	6:22

6.2 DB Server Utilization

The database utilization remained fairly constant across the individual processor steps, as it would be expected for batch jobs. Between the steps, database utilization varies, caused by the different program and processing logic that each step represents. In Figure 7 the y-axis shows the DB server CPU utilization, and the x-axis is a function of elapsed time, visualizing the varied duration of the steps under test as well as the total processing time.

Figure 7. Database Server CPU% Utilization for 8 million loans



6.3 Application Server Utilization

With every test cycle, we collected a detailed set of application server monitoring data. We used the monitoring data output for error searching, statistics, tuning and for this documentation. To run data collection, we embedded an automation for the monitoring process into each job (find more information about the automated data collection in chapter 7.4 'Measurement methodology & tools'). To analyze the application server hardware resource consumption, we used the IBM AIX Tool NMON. We had NMON collect data in intervals between 20 and 60 seconds, depending on the total elapsed time that we anticipated, in order to keep the amount of measurement data at a manageable level

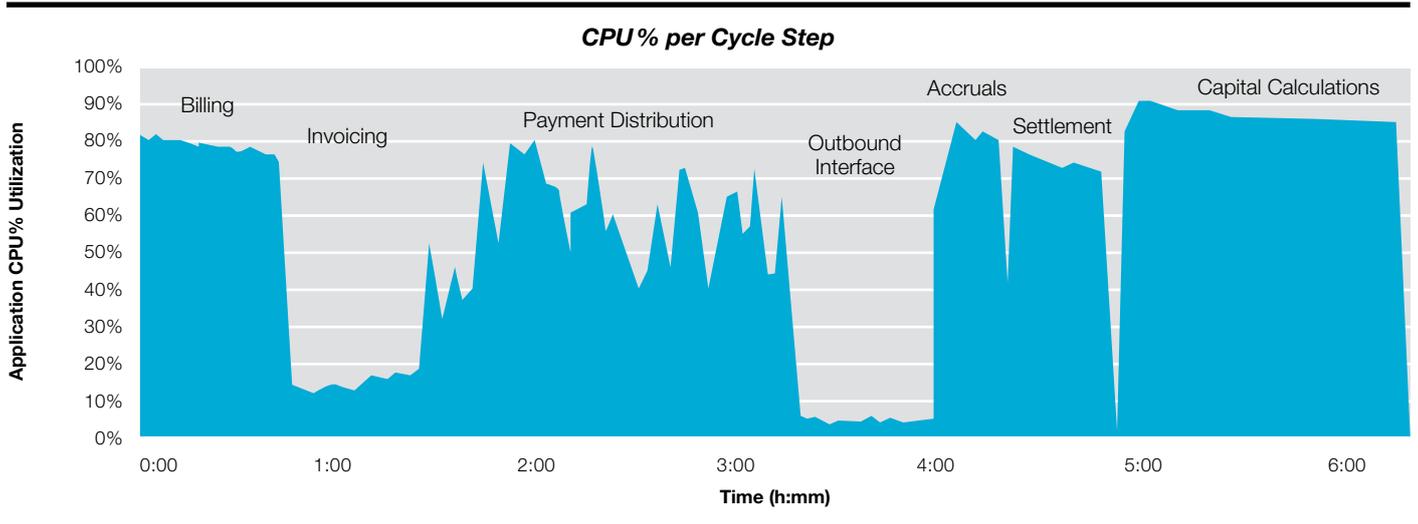
In this chapter we focus on CPU, memory and network utilization. To be flexible in migrating our test SAP environment to different server hardware, we run the SAP NetWeaver Central Services on a separate, dedicated AIX server, an IBM Power 740.

For Batch and Dialog the workload of the application servers was distributed across several LPARs and instances, running in parallel on an IBM Power 795. The utilization profile of the LPARs was similar. The result graph summarizes the consumption of the LPARs under test.

You find additional information about the 500K, 2 million, 4 million, and 8 million cycles in the appendix.

The following figure shows the CPU consumption of each of the banking applications.

Figure 8. Application Servers CPU % utilization for 8 million loans



By splitting the Power 795 server into four logical partitions, setting up 48 Dialog instances and optimizing up to a total of 1632 dialog and work processes we could balance the workload for an optimal resource allocation and throughput.

7 Testing Methodology

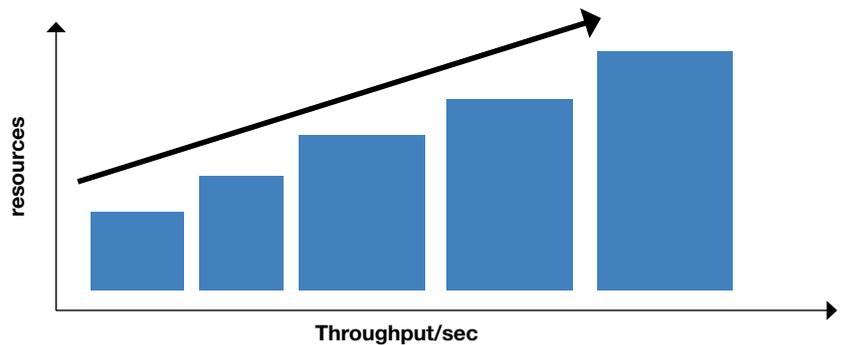
7.1 Test approach

In order to provide a reference architecture and best practices guide for implementation and customer assurance, we evaluated all potential bottlenecks in process steps that are on the critical path of a typical loans end-of-day process chain. To do this effectively, we separated the steps into repeatable component steps which allowed us a focused tuning and evaluation.

We performed several tests for each step in isolation by the following approach:

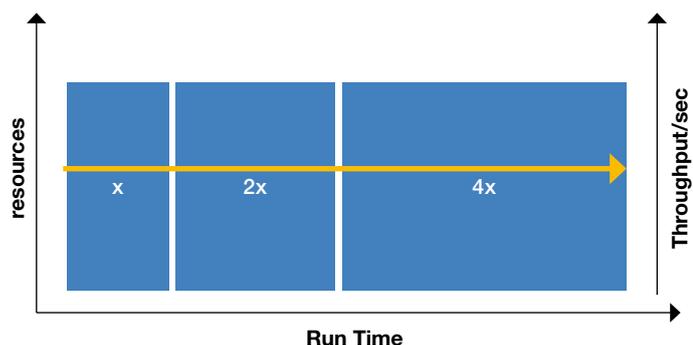
- Run the step with increasing parallelization—check scaling by increasing the number of jobs (=scalability) while keeping the total volume to be processed constant

Figure 9. Scaling out by parallelization



- Run the step with increasing volume—check linear scaling by increasing the volume (= reliability) while keeping the number of parallel jobs constant

Figure 10. Scaling out by increasing data volume



After each cycle we analyzed the metrics and made adjustments to the hardware or software setup in order to optimize each step in the process separately. A stress testing was performed to find out the maximum possible load on the system for each step if scalability was sufficient. To reach optimal throughput an application server CPU load of 80 % was targeted by adapting the degree or parallelism of each step.

After all the process steps were optimized we ran each step of the complete cycle to obtain measurements of throughput and time required for the total volume with the optimized settings.

7.2 Data creation

Data Generation

Master data generation was done using the *Mass Data Generator (MDG)* tool from SAP Active Global Support. MDG is a tool used for fast and easy creation of master data and transactional data. The tool can create business partners, all kinds of different accounts (deposits, loans, savings, etc.), master contracts, in- and outgoing payments as well as the payment orders, forward orders and standing orders. Data creation is done using parallel processing and is therefore optimized for creating large volumes of data.

With MDG we created the following data objects:

- Business Partners^a (BP) in same volume as the number of loan accounts, so that one BP is the account holder of exactly one loan account.
- Loan accounts including the loan disbursement. Loan disbursement was done as cash disbursement.

With MDG data was created on the Banking Services system and then transferred to FI-CAX system using the SAP standard tools.

- Business Partners were automatically sent to FI-CAX system using the IDocs as they were being created on the BaS system.
- Accounts were transferred to the FI-CAX system using the SAP standard report for Account Master Data distribution.

This means that for 100 loan accounts in Banking Services we had in total 100 business partners in both Banking Services and FI-CAX and 100 contract accounts in FI-CAX.

^a In SAP Banking terminology, customers or account owners are often called "Business Partners". This might be in contrast to other Business Partner terms.

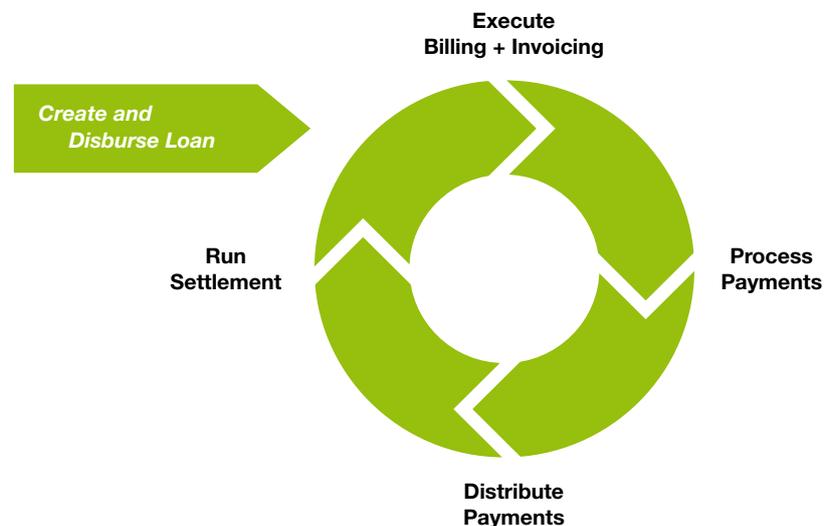
Data Aging

Once the newly created loans were created on both systems we did data aging. This means we fast forward the system to create transactional data of long time periods such as 3 months or 1 year on a single day or few days. By this technique we are able to run the performance test on a production-alike data basis that has “aged” for months or years.

Data aging is achieved by processing loan accounts through the loans lifecycle through a few periods. To process loans through the lifecycle there are a few steps which need to be executed:

1. Billing + Invoicing
2. Loan periodic repayment
3. Payment Distribution + Outbound Interface
4. Account Settlement

Figure 11. Aging of the loans



Each of the steps needs to be executed one after another and the whole cycle should be repeated three times in order to age the loans through three periods. We chose to do aging through three periods because we did this for each of the four data loads, and we wanted to have a system which is at least six months to one year productive.

Data aging tool

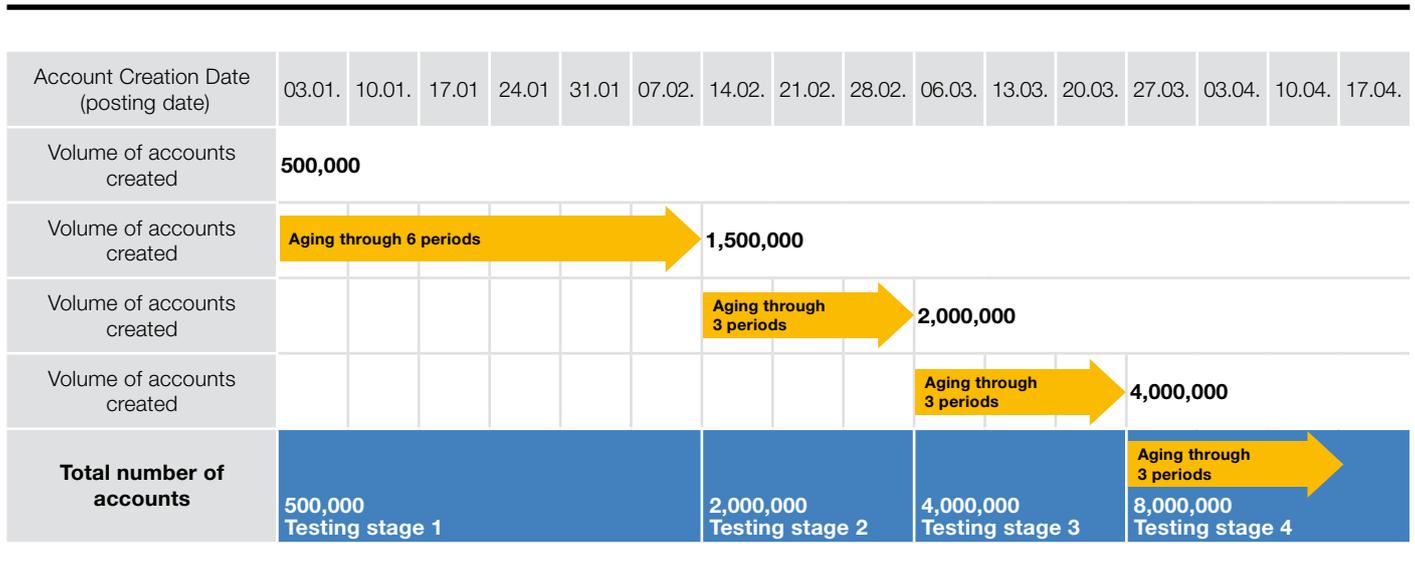
Data aging can either be executed manually or with the *Contract Lifecycle Tool* from SAP AGS. That way the aging can be done in an automated way. We ran the steps mostly manually using the standard SAP jobs, so we can have a better control of when we execute a step. There is no standard report to create mass incoming payments automatically, so we did this step with the *Contract Lifecycle Tool*.

Data aging in our tests

For performance testing we processed the data through three periods, except for the first 500.000 loans, which we processed through six periods.

In four different stages of performance testing our data volume and the aging of the data was as following:

Figure 12. Data volumes and aging of the loans for the four test stages



Stage 1: 500,000 accounts

In the first testing stage we created 500,000 accounts and did the account aging through six periods. This was our account base for the first testing stage.

Stage 2: 2 million accounts

In the second testing stage we additionally created 1.5 million accounts on top of the existing 500,000. Therefore the total data volume for the second testing stage was 2 million accounts. We did the aging for three more periods, having in the end

1.5 million accounts processed through three periods and 500,000 accounts processed through nine periods. This was our account base for the second testing stage.

Stage 3: 4 million accounts

In the third testing stage we created another 2 million accounts, having in total 4 million accounts. We did the aging for three more periods, having in the end 2 million accounts processed through 3 periods, 1.5 million accounts processed through six periods and 500,000 account processed through 12 periods. This was our account base for the third testing stage.

Stage 4: 8 million accounts

In the last testing stage we created another 4 million accounts, having in total 8 million accounts. We did the aging for three more periods, having in the end 4 million accounts processed through 3 periods, 2 million accounts processed through six periods, 1.5 million account processed through nine periods and 500,000 account processed through 15 periods. This was our account base for the fourth testing stage.

Database backups during the data creation

During the data creation and aging process we took several backups between the steps, so that we could restore the database to the last consistent state in case of any unexpected events. We did a backup after the following data load stages:

- After creating BPs and accounts with the MDG tool we did a consistency check to make sure all accounts and BPs are created in the BaS system and also all BPs are transferred to the FI-CAX system. When all the data we expected had been created we did a backup.
- After transferring the accounts to FI-CAX system we did a consistency check to make sure all accounts are transferred to FI-CAX system. If all the data we expected was created we did another backup.
- After each aging cycle we did a backup when each of the aging steps had successfully completed.

In case of any errors within the data creation, data transfer or data aging, we could restore the last backup. Once the data creation and aging was complete and all data was consistent we took a final backup, called the baseline backup which we used for running the tests.

Housekeeping jobs

After each data load, a general housekeeping was performed to clean up run data and tables. We ran certain jobs to clean application logs, PPF logs, job logs, spool files as well as ABAP dumps. The following housekeeping jobs were run:

Table 3. Housekeeping jobs used

Housekeeping job	Report name
Delete ABAP dumps	RSSNAPDL
Delete old jobs	RSBTCDEL2
Delete orphaned job logs	RSTS0024
Delete old spool files	RSPO0041
Delete PPF data	RBANK_DELETE_RUNS_ALL
Delete application logs	SBAL_DELETE
Delete change documents	RSCDOK99

7.3 Running the test

The starting point for each test cycle was the last baseline backup. We executed the defined process steps separately. For each of the process steps we collected relevant measurements for the application server side as well as the database. When necessary, we took SQL and ABAP traces and analyzed them to find any tuning potential.

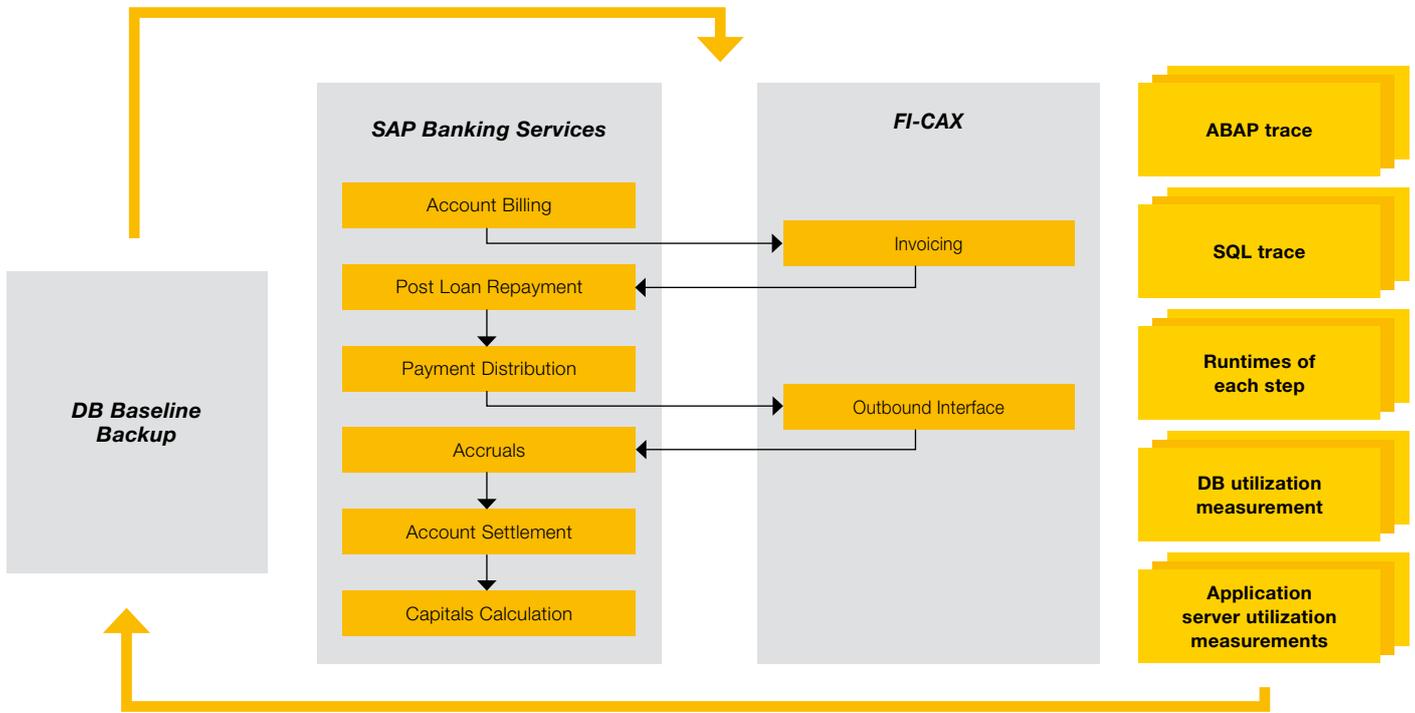
After analyzing each full cycle we restored the database to the baseline backup. Before restoring we noted and saved the relevant run information, such as runtime and the times of execution. We also saved any traces which we wanted to keep locally on the computer, since with the database restore all the data we made during the test cycle would no longer exist.

We then repeated the test cycle again from the baseline backup. This made all our test cycles within one testing stage one-to-one comparable, as they were all done with the identical data base.

Implementing performance improvements

When we implemented any performance improvement, in forms of an SAP Note or a parameter change of any kind, we compared the new results with the older ones and we measured any improvement the changes brought. If we were satisfied with the improvements and decided to permanently save it in our baseline backup, we

Figure 13. Main performance measurements when running the tests



restored the database again (as we did not want to save the data from the current test cycle), implemented the improvements and made a new baseline backup. We kept a log of all the changes we did in the system (parameter change, SAP Note implementation, etc. ...), so we could match the system performance at each point in time with the actual state of the system.

Once we finished tuning all the process steps with the certain volume of accounts, we restored to the last saved baseline backup and moved to the next testing stage, where we loaded more data into the system and continued testing and tuning.

7.4 Measurement methodology & tools

Collecting the result and monitoring measurements

The IBM Monitor Integration Solution for SAP (MIS4SAP)[®] was used to collect results and monitoring measurements. We integrated the tool into each of the steps of the cycle and we took a separate measurement for each of them. The measurements collected are resource consumption on the database side, resource consumption on the application server side such as CPU, memory consumption, network, IOPS and SAP Statistical Records.

Focus metric

The focus metric we used to measure the success of each of the steps was the throughput, or the number of accounts which can be processed in one hour. For collecting the throughput relevant data we used the SAP Mass Run Analysis Tool (MassMan).

The formula we used to calculate the throughput was the following:

$$\text{Throughput} = \langle \text{Account_Volume} \rangle / \langle \text{Runtime} \rangle \times 3600$$

where the Runtime is equal to the longest running parallel work process. With each of the changes we did either in the infrastructure or in the application we observed the throughput in order to determine how much improvement each change gave us.

Monitor Integration Solution for SAP

Monitor Integration Solution for SAP (MIS4SAP)^b is an IBM developed tool for system architects that handles lots of performance measurements in an SAP landscape. The tool allows incorporating many of the technical monitoring tools the operating systems and third party software product vendors provide. Using MIS4SAP we automated the following activities:

^b The MIS4SAP tool is an IBM service offering by the SOLAR team in Germany and other countries. Please contact ISSIC@de.ibm.com if you are interested to determine if it is available in your country.

- Coordinating the start/stop of all involved monitoring tools.
- Managing the output of all involved monitoring tools.
- Providing the output protocols with timestamp and saving in a centralized folder.

Figure 14. MIS4SAP monitoring flow

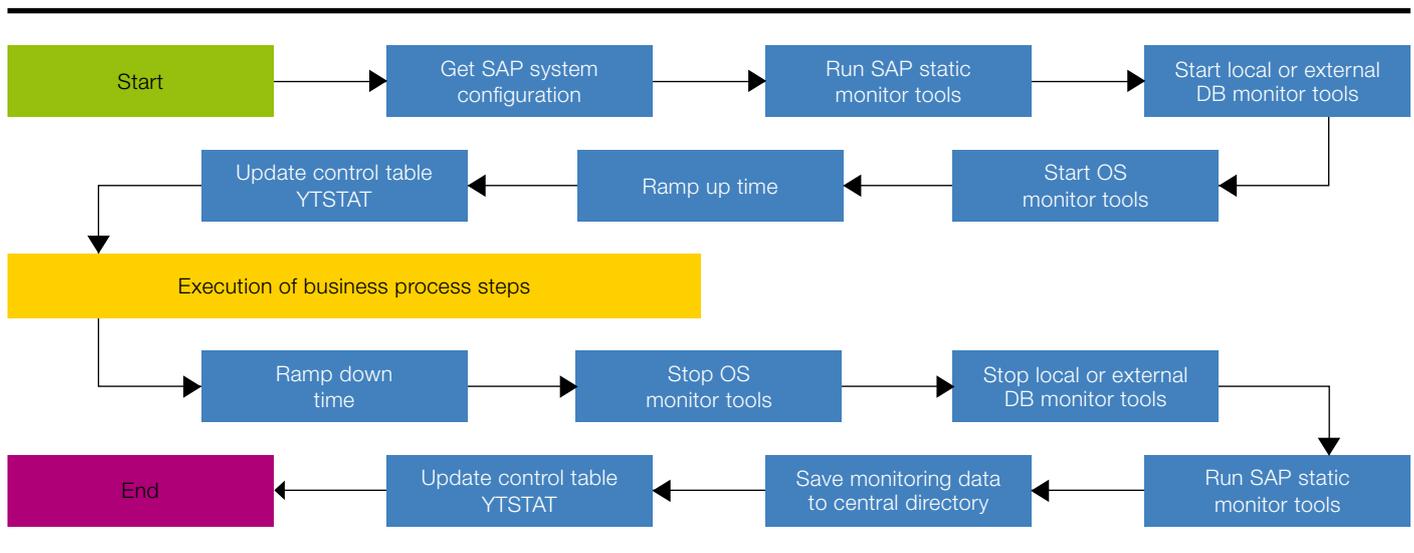


Figure 14 shows the basic workflow of MIS4SAP. This diagram shows the correct sequence of execution of the Figure 14. MIS4SAP monitoring flow steps.

Monitoring implementation and time profile

For all banking services and SAP Loans Management business processes in a test cycle we implemented an SAP batch job, based on the following:

Step 1: Program Y_MIS_START / User MIS4SAP

Step 2: Program of BS or ERP business process / User SAP_PERF

Step 3: Program Y_MIS_STOP / User MIS4SAP

Figure 15. MIS4SAP automated monitoring start/stop and time profile

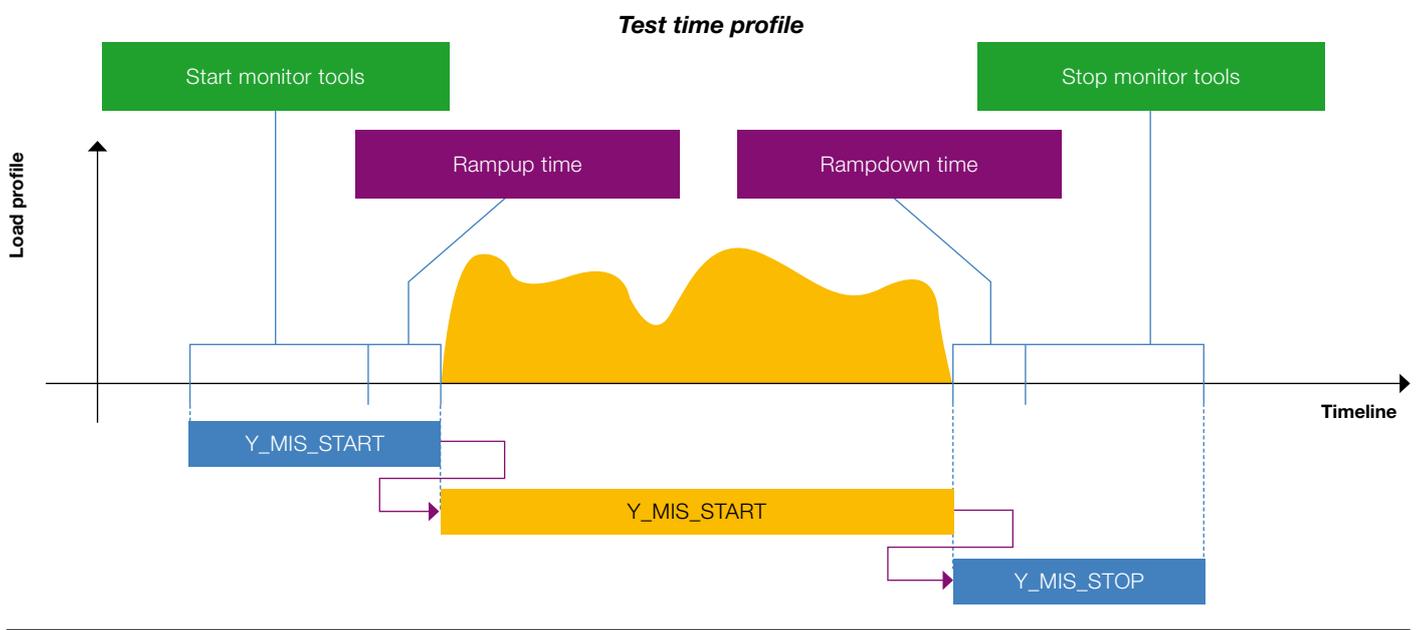


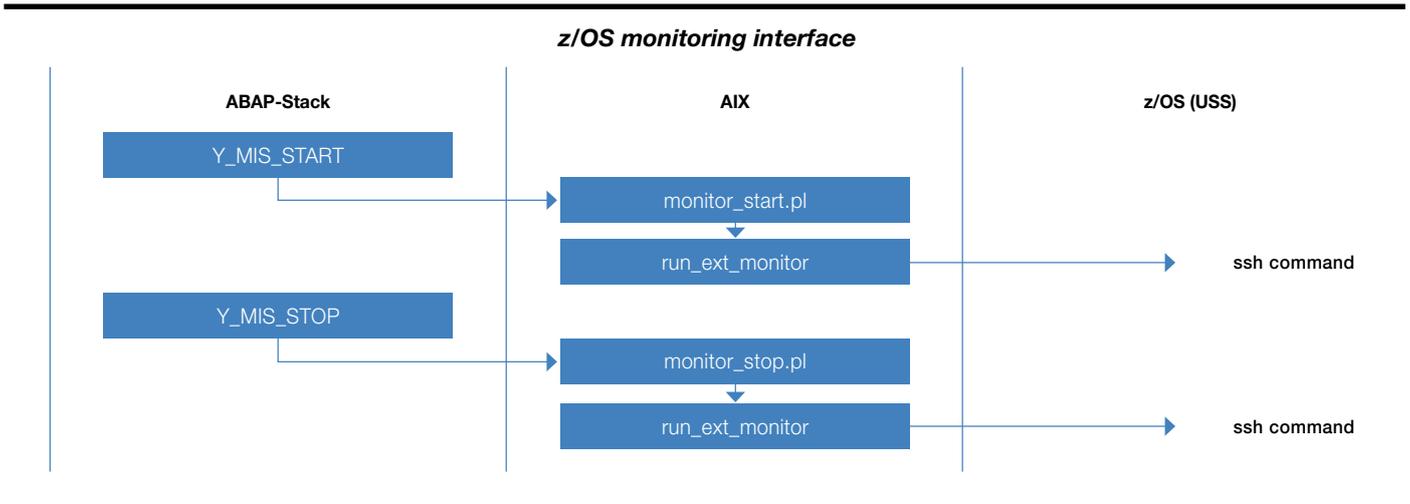
Figure 15 displays the timing between the monitoring tools and the workload, which could be observed. After the start procedure for all SAP, DB or OS monitoring tools, a ramp up time could be defined. This ramp up time guaranteed that all interval-based monitors did their first measurement snapshots, before the real workload started. The ramp down time guaranteed the same after the workload ended. This mechanism helped us monitor exactly the ramp up and ramp down behavior of the workload.

Setup of MIS4SAP and specials for the SAP Loans Management project

On the SAP Banking Services System and the connected SAP FI-CAX system we created a dedicated user to run all monitoring tools for the SAP workload. A second user was created to run the business processes. We separated this workload, caused by the monitoring user, to isolate it from the workload caused by the business processes user. That allowed us to identify the monitoring overhead.

MIS4SAP was installed on the Application Server on AIX. To interact with the Database Server, running on z/OS, we implemented a ssh connection to the z/OS Unix System Services. Thus we automated and synchronized the control of RMF report reporting of z/OS with the application server measurements.

Figure 16. MIS4SAP communication with AIX and z/OS



This diagram explains the real communication between MIS4SAP and the z/OS system.

8 Test set-up

8.1 Application configuration

The key to accomplish meaningful performance results is to execute the performing tests on a customer-alike data basis. Therefore we defined a loans portfolio matching different attributes of the typical retail loan products consumer loan and mortgage loan. Both products were evenly (50/50) represented in the loans portfolio. To implement these products and the relevant loans life-cycle processes quickly and with minimal testing effort, we used the “express path” deploying the “SAP Loans Management rapid-deployment solution (RDS)”. An RDS is a preconfigured content that helps customers to quickly and affordably deploy SAP solutions.

For further details regarding the RDS see service.sap.com/rds-loans.

The performance test has considered the following attributes:

Table 4. Product features and attributes

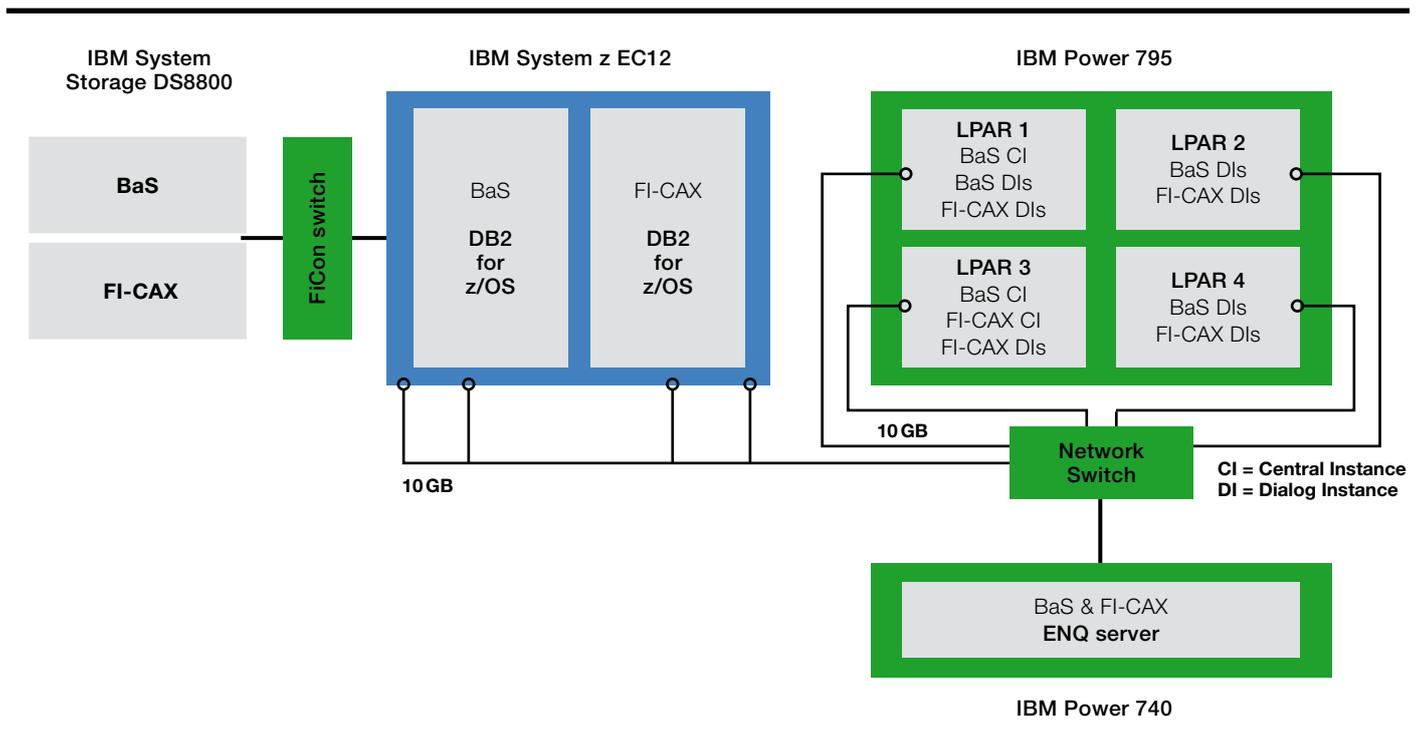
Loan product feature	Attributes
Short and long term loans	A consumer loan with 16 monthly installments. A mortgage loan with 96 monthly installments (an 8 years term).
Variable and fixed interest calculation	The interest is either a fixed rate or calculated on a floating reference rate by a customer specific formula.
Repayment option	Annuity and installment payments
Pricing options	Nominal interest + periodic charges
2 Interest calculation methods	Interest calculation according to incoming payments (Incoming payment principle) Interest calculation according to a payment plan (Payment plan principle)

A goal of using different attributes was also to analyze in detail their impact on runtimes and to derive optimization measures such as code changes or implementation advices. See chapter 8.5 Application workload details for details.

8.2 Application server set-up

The test system was split into four logical partitions (LPARs), with both Banking Services and FI-CAX systems in each LPAR running on AIX.

Figure 17. SAP Loans Management solution landscape



Banking Services system had 33 instances, one being the Central Instance and the rest 32 were the Dialog Instances (DI) split among the four LPARs, each LPAR having 8 DIs. FI-CAX system had 17 instances, one being the Central Instance and the rest 16 were the Dialog Instances split among the four LPARs, each LPAR having 4 DIs. A standalone enqueue (ENQ) server was used, on the separate box.

Each central instance had the same configuration:

- 10 dialog work processes
- 1 update work process
- 5 background work processes
- 1 spool work process
- 1 update 2 work processes

Application server instances were configured in the following way:

On BaS System:

- 20 dialog work processes
- 16 background work processes

On FI-CAX System:

- 18 dialog work processes
- 12 background work processes

In total on the application server instances on BaS system we had 512 background work processes and 640 dialog work processes. On FI-CAX system we had 192 background work processes and 288 dialog work processes.

Load Balancing

We wanted the workload to run only on the application server instances, not on the central instance. That is why we configured one server group, where we included 32 application server instances on the BaS server. On FI-CAX system we used the automatic workload distribution, so the workload was distributed among all instances.

For the communication between the BaS and FI-CAX we enabled the load balancing mechanism for the RFC connections. On each of the systems we also configured one logon group, where we included only the application server

instances on that logon group, not the central instance. Additionally the group was enabled for External RFC processing and we used the Weighted Round Robin as the preferred load balancing option. That way all the incoming RFCs would run only on the application server instances and the load would be equally distributed among application server instances.

8.3 Database Setup

The databases were built using DB2 10 for z/OS dated January 2013 and partitioned as per SAP Notes 496904 and 1295963 and each table using 128 partitions. Bufferpool isolation was done on both DB2 subsystems. See the appendix for isolation specific details as well as bufferpool sizes.

8.4 End-of-day processing design

Goal of the performance test was to create a worst-case scenario for the EOD processing peak. At the same time it means the result figures can be over-achieved by some simple (non-technical) adjustments to the EOD processing design based on customer requirements. The following table shows some options to optimize the EOD processing runtime.

Table 5. Options for EOD runtime optimization

Process	Optimization by	Impact on EOD processing
Billing	Usually customers want to generate open items some days before they become due. This can be achieved by using the parameter "billing lead days". To distribute billing for peak due dates to more than one previous EOD processing a customer could even set different lead days for different products.	If billing lead days are > 0, the process steps for billing and invoicing are not part of the peak EOD processing any longer. Total peak EOD processing runtime is reduced.
Payment distribution	Incoming payments may arrive during the day. If a customer wants to update the loan account balance intraday, too, payment distribution needs to run intraday.	Less payments to be processed in peak EOD processing. Total peak EOD processing runtime is reduced.
Payment distribution	For some payment methods an incoming payment is posted assuming that the corresponding debit to the customers transaction account will be executed successfully. Only when the debit fails then the credit (incoming payment) is reversed. These payments can be scheduled more flexibly, e.g. in the early morning of the peak day.	Less payments to be processed in peak EOD processing. Total peak EOD processing runtime is reduced.
Calculate Capital amounts	Not all key figures for reporting are required to be updated daily. Verify if some key figures can be deselected for the EOD processing peak day in the select options of BCA_CAPITAL_CALC.	Less key-figures to be calculated in peak EOD processing. Total peak EOD processing runtime is reduced.
Calculate Capital amounts	Depending on the detailed implementation design it is possible to re-calculate some key figures only based on certain event. Check SAP note 1844363 for details.	Less key-figures to be calculated in EOD processing. Total peak EOD processing runtime is reduced.

8.5 Application workload details

To achieve optimal application performance and hardware consumption of each of the process steps in the loans lifecycle we focused on two main parameters, number of parallel processes and the package size. Number of parallel processes influences the load we put on the system and the package size determines the number of objects to process per work package in parallel processing.

Number of parallel processes

We used different number of parallel processes for each process step because of the different load they produce and the different resource consumption they cause. With the number of parallel processes used we wanted to achieve around 80% CPU consumption on the application servers and also not exceed the 80% utilization of the database.

To find the optimal number of parallel processes for a specific process step we started testing with 240 parallel processes (a number close to the number of available CPU cores on the application servers) and observed the utilization on the application servers and the database. If they were below the target utilization of 80% we would increase the number of parallel processes and observe the resource consumption again. At the same time we checked if the throughput of the process steps was increasing with the number of parallel processes.

Table 6. Billing runs with different numbers of parallel processes

Process name	Number of parallel processes	Throughput (loan accounts/hour)	App. server CPU utilization	DB server CPU utilization
Billing	240	5,263,158	50 %	39 %
Billing	320	5,950,413	64 %	45 %
Billing	416	6,360,424	75 %	47 %
Billing	480	8,172,531 ^c	81 %	54 %

With 480 processes we achieved the 81% CPU utilization on application servers. The throughput was also the best we measured, so the optimal number of parallel processes for billing on our test system was 480.

^c The large increase in throughput can partly be attributed to performance enhancements in the application

We did the same exercise for the rest of the process steps and determined the optimal number of parallel processes for each of the process steps in the runs with 8 million loan accounts:

Table 7. Optimal number of parallel processes for the different steps

Process name	Number of parallel processes
Billing	480
Invoicing	160
Payment Distribution	240
Outbound Interface	190
Accruals	480
Settlement	480
Capitals Calculation	512

Package Size

The number of accounts to be processed in one process step will be split into smaller bundles, called work packages. Package size determines how many loan accounts will be included into one work package. Technically one work package will be processed and committed to the database as one unit. The optimal package size will bring the optimal application performance.

To find the optimal package size for a specific process step we started testing with the package size 50, increased it gradually and observed the throughput.

Table 8. Determination of the optimal package size for the billing run

Process name	Number of parallel processes	Package size	Throughput (loan accounts/hour)
Billing	416	50	5,960,265
Billing	416	100	6,315,789
Billing	416	150	6,360,424
Billing	416	200	6,228,374

The best throughput we achieved with the package size of 150, therefore this was the optimal package size for billing.

We did the same exercise for the rest of the process steps and determined the optimal package sizes for each of the process steps:

Table 9. Optimal package size for the different steps

Process name	Number of parallel processes	Package size
Billing	416	150
Invoicing	160	100
Payment Distribution	240	100
Outbound Interface	190	100
Accruals	480	100
Settlement	480	100
Capitals Calculation	512	100

Differences between consumer and mortgage loans in runtime and throughput

In the SAP Loans Management performance tests we used two types of products, consumer and mortgage loans. The main differences between the products are the term length, interest calculation methods and the usage of the variable and fixed interest rate. We had 50 % of consumer loans and 50 % of mortgage loans in the system. The details of the two different products you can find in chapter 8.1 Application configuration. We wanted to measure the influence of the different product features on performance, so we did one test where we separated the consumer and mortgage loans. Here are the results:

Table 10. Throughput per product for different steps

Process name	Number of loan accounts	Throughput (loan accounts/hour)
Billing Consumer loans	4 million	5,990,017
Billing Mortgage loans	4 million	9,125,475
Invoicing Consumer loans	4 million	12,339,332
Invoicing Mortgage loans	4 million	13,198,900
Payment Distribution Consumer loans	4 million	4,628,737
Payment Distribution Mortgage loans	4 million	5,265,082
Outbound Interface Consumer loans	4 million	10,249,110
Outbound Interface Mortgage loans	4 million	9,213,052
Accruals Consumer loans	4 million	22,748,815
Accruals Mortgage loans	4 million	21,460,507
Settlement Consumer loans	4 million	16,705,336
Settlement Mortgage loans	4 million	15,618,221
Capitals Calculation Consumer loans	4 million	6,357,616
Capitals Calculation Mortgage loans	4 million	5,755,396

Our results show that the runtime and the throughput are almost the same in most of the process steps of the loans lifecycle, except for billing where there is a big difference between processing consumer loans and processing mortgage loans. Consumer loans using the variable interest rate and mortgage loans using the fixed interest rate cause this difference. The variable interest rate involves a more complex installment calculation and is therefore more expensive. Combined with a recalculation of the installment amount is more complex than a fixed interest rate with a fixed installment amount. That is why it takes longer to bill 4 million consumer loans than the 4 million mortgage loans. Other product differences, such as term length do not have a major influence on performance of the process steps in the loans lifecycle.

8.6 Database—Large Table Analysis

The database was partitioned as per SAP Notes 496904 and 1295963 and each table used 128 partitions. The largest banking tables are listed in the table below. This data was collected after the 8 million loans were generated and aged.

Table 11. Largest banking tables after 8 million created and aged loans

BaS tables	Number of rows
/FSPD/RESULT ^d	3,400,000,000
BCA_TRANSFIG	511,619,041
BCA_PAYMITEM	309,078,676
BKK92_SUMS	271,083,225
BCA92_RESTART	240,572,183
BCA_CN_LINK	181,668,459
BCA_PO_IT	174,052,189
BCA96	112,533,989
BCA_BL_ITEM ^d	112,000,000
BKK92_POSTINGS	90,018,465
BCA_ACCTBAL	88,522,871
BCA92	53,515,860
BCA_PO_HD	53,265,225
/FSPD/ITEM ^d	45,000,000
BCA_CN_EVENT	42,264,541
BCA_CONTRACT	16,755,768
BCA_BCAS_EVBST	8,502,788
BCA_BANO_DUE	8,252,974
BCA_BCAS_DUE	8,252,974
BCA_CNBP_ACCT	8,252,974
BCA_CN_PER_ACBAL	8,252,911
BCA_ALIAS	8,251,970

^dEstimate, data not recorded

9 Customer-specific end-of-day runtime projection

Every customer will have different business models, different implementations and therefore different performance requirements. That means the results of the performance test as presented in chapter 6 need to be put in the specific customer context. The following two examples show how to do this.

Determination of the customer-specific end-of-day processing peak:

Considering that we focused on an extreme peak, the actual customer portfolio size might be different. Assume that a specific bank has a loan portfolio with 40% of the loan accounts having the same installment due date, then the total amount of loans would be 20 million loan accounts generating an equivalent EOD processing peak as in our performance test of 8 million loan accounts (20 million x 40% = 8 million).

Determination of a customer-specific EOD process chain:

Assuming the following customer scenario:

1. The loans portfolio comprises 20 million loans
2. Determination of the EOD processing peak: 40% of all loans (8 million accounts) have an identical due date
3. Billing and Invoicing runs 2 days before the installment date
4. 30% of incoming payments of due installments occur during the day and are processed intraday, leaving 70% of 8 million loans to be processed during EOD.
5. Accrual runs for the entire portfolio while settlement runs only for the loans that are due

Assuming that the same hardware and the same technical settings are used (see chapter 8, Test set-up), the projection of the performance test figures can be calculated as shown in the following table:

Table 12. Projecting performance results for different process steps

Process step	Runtime per step in hours (h:mm) for 8 million loans	Customer-specific adjustment factor	Projection results
Billing	0:58	0 %	
Invoicing	0:41	0 %	
Payment Distribution	1:46	70 %	1:14
Outbound Execution	0:43	70 %	0:30
Accrual	0:21	250 %	0:53
Settlement	0:29	100 %	0:29
Capital Calculation	1:20	100 %	1:20
Sum	6:22		4:26

As result, the projected processing time for an EOD processing of a portfolio of 20 million is approximately 4.5 hours.

Unquestionably both calculations are only indications; to determine profound figures a more detailed, customer-specific analysis is required.

10 Summary and Outlook

This project is another example of IBM and SAP working together to show successful collaboration in SAP Loans Management on IBM System z in the area of core banking.

Major outcomes of the joint performance test were:

- We proved scalability of the application and the technical infrastructure
- Performance results were achieved while running the application and the hardware under standard conditions and without any modification
- Functionally stable environment
- Successful performance test for SAP Loans Management 8.0 on IBM System z
- Very good and fruitful teaming of SAP and IBM
- We are able to provide valuable information to customers for implementation and sizing.

For the future of SAP applications running on the IBM System z platform, following thoughts should be considered:

- Only 20% of the available database server capacity of the EC12 system was used. Therefore we assume that further vertical scaling is possible; however, we cannot predict to what extent scaling will be possible.
- We found no bottleneck that would stop us from going to even higher peak volumes—we stopped at 8 million due to the project time restrictions; higher loads should be possible.
- On the IBM System z database server, we did not activate the IBM System z Integrated Information Processor (zIIP). A significant cost benefit is possible by activating zIIPs.

IBM and SAP will continue their successful collaboration in co-innovation projects in Banking.

11 Appendix

11.1 Result details per testing stage

The following tables show the result details per testing stage of 500K, 2 million, 4 million, and 8 million loan accounts.

Table 13. Result details for 8 million loan accounts

Process step	8 million				
	Runtime in sec	Runtime hours	ms per loan	Parallel jobs	Throughput per hour
Billing	3,524	00:58:44	0.441	480	8.172.531
Invoicing	2,496	00:41:36	0.312	160	11.538.462
PaymentDistr	6,364	01:46:04	0.796	240	4.525.456
Fpoitr	2,631	00:43:51	0.329	190	10.946.408
Accrual	1,312	00:21:52	0.164	480	21.951.220
Settlement	1,787	00:29:47	0.223	480	16.116.396
Capitals Calc	4,858	01:20:58	0.607	512	5.928.366
Total Elapsed Time	22,972	06:22:52	2.872		

Table 14. Result details for 4 million loan accounts

Process step	4 million				
	Runtime in sec	Runtime hours	ms per loan	Parallel jobs	Throughput per hour
Billing	2,078	00:34:38	0.520	416	6.929.740
Invoicing	951	00:15:51	0.238	160	15.141.956
PaymentDistr	3,883	01:04:43	0.971	240	3.708.473
Fpoitr	1,519	00:25:19	0.380	190	9.479.921
Accrual	671	00:11:11	0.168	480	21.460.507
Settlement	995	00:16:35	0.249	480	14.472.362
Capitals Calc	2,412	00:40:12	0.603	512	5.970.149
Total Elapsed Time	12,509	03:28:29	3.127		

Table 15. Result details for 2 million loan accounts

Process step	2 million				
	Runtime in sec	Runtime hours	ms per loan	Parallel jobs	Throughput per hour
Billing	1,132	00:18:52	0.566	416	6.360.424
Invoicing	554	00:09:14	0.277	160	12.996.390
PaymentDistr	1,812	00:30:12	0.906	240	3.973.510
Fpoitr	1,151	00:19:11	0.576	190	6.255.430
Accrual	345	00:05:45	0.173	480	20.869.565
Settlement	513	00:08:33	0.257	480	14.035.088
Capitals Calc	1,217	00:20:17	0.609	512	5.916.187
Total Elapsed Time	6,724	01:52:04	3.362		

Table 16. Result details for 500,000 loan accounts

Process step	500,000				
	Runtime in sec	Runtime hours	ms per loan	Parallel jobs	Throughput per hour
Billing	291	00:04:51	0.582	320	6.185.567
Invoicing	140	00:02:20	0.280	120	12.857.143
PaymentDistr	436	00:07:16	0.872	240	4.128.440
Fpoitr	201	00:03:21	0.402	190	8.955.224
Accrual	138	00:02:18	0.276	320	13.043.478
Settlement	163	00:02:43	0.326	320	11.042.945
Capitals Calc	327	00:05:27	0.654	512	5.504.587
Total Elapsed Time	1,696	00:28:16	3.392		

11.2 Elapsed processing time per process step

Figure 18. Billing Elapsed Times

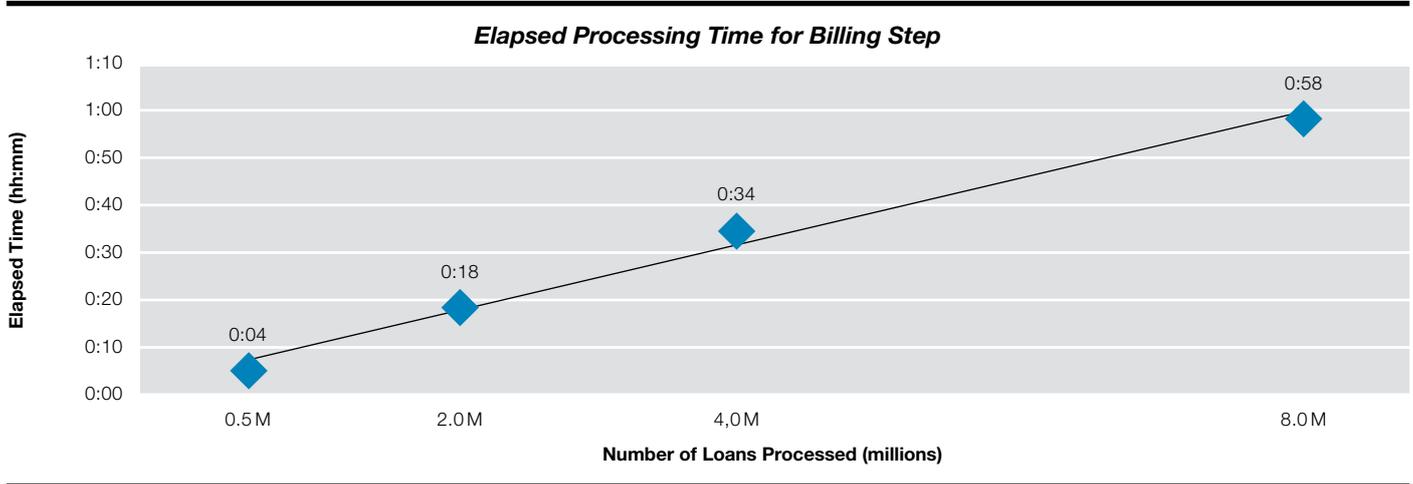


Figure 19. Invoicing Elapsed Times

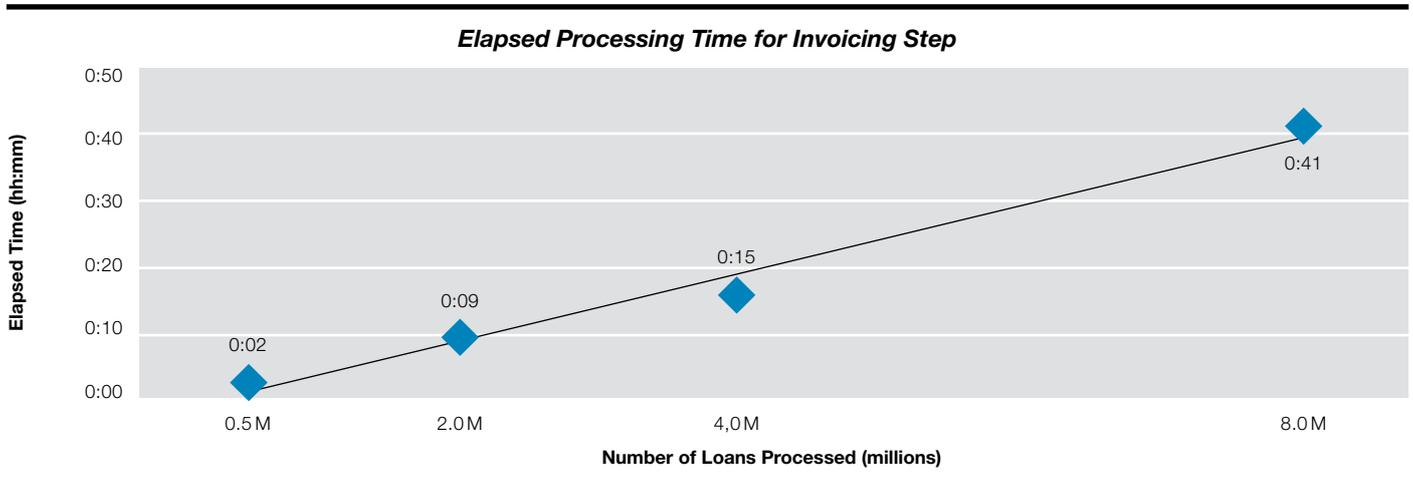


Figure 20. Payment Distribution Elapsed Times

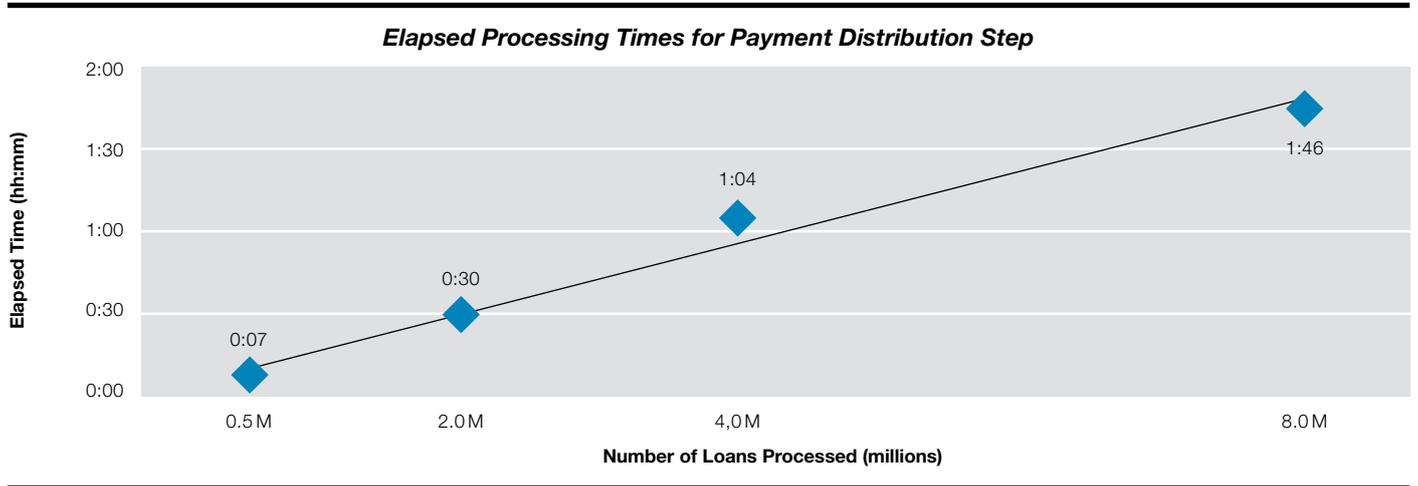


Figure 21. Outbound Interface Elapsed Times

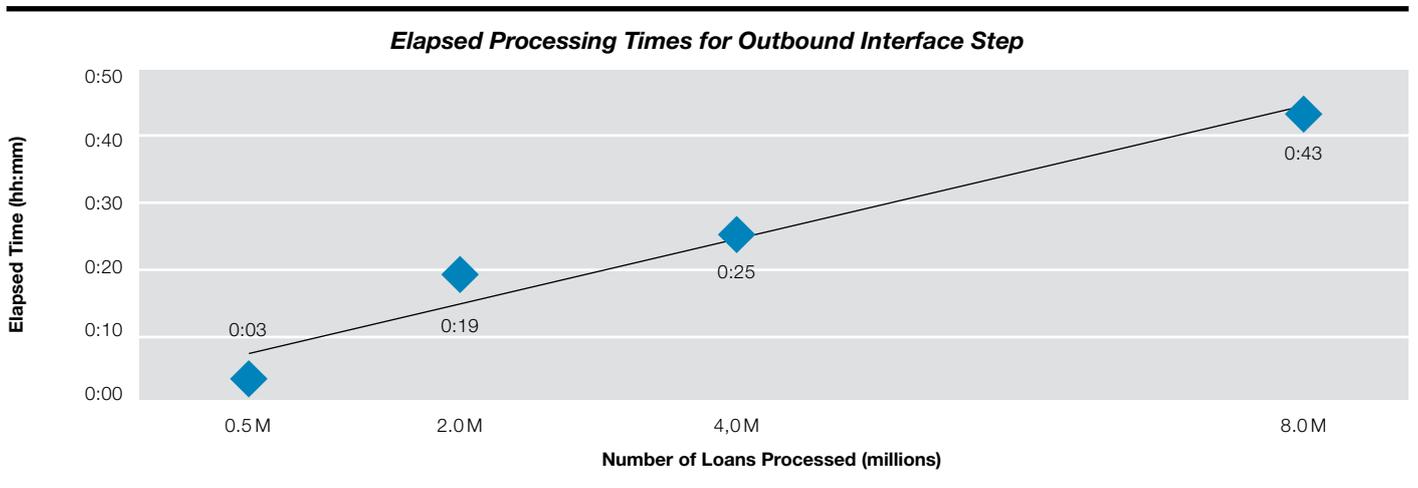


Figure 22. Accruals Elapsed Times

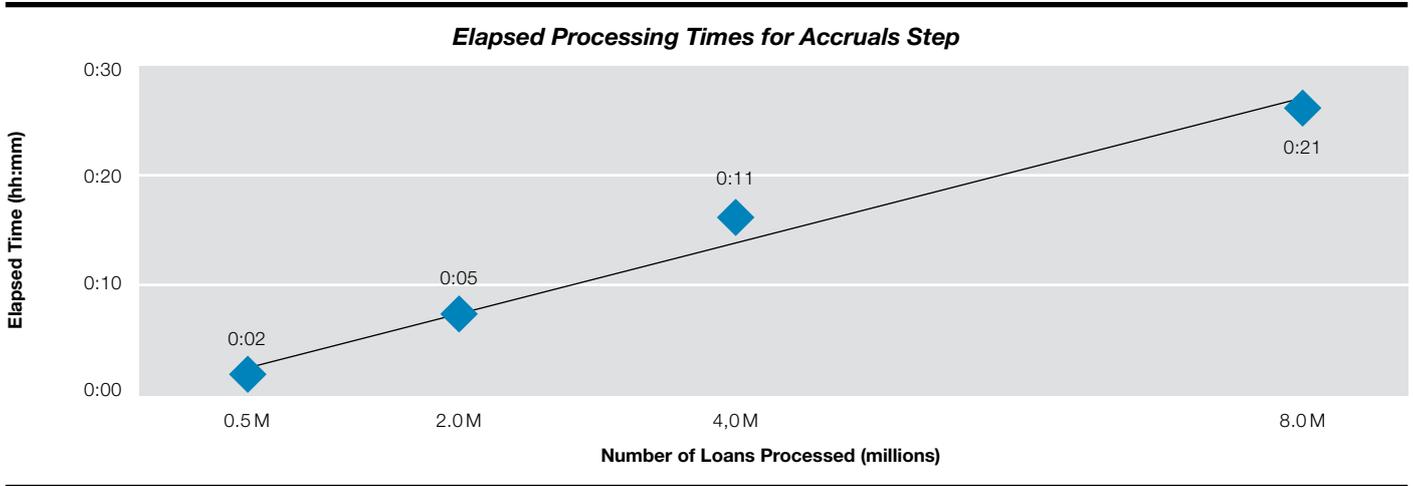


Figure 23. Settlement Elapsed Times

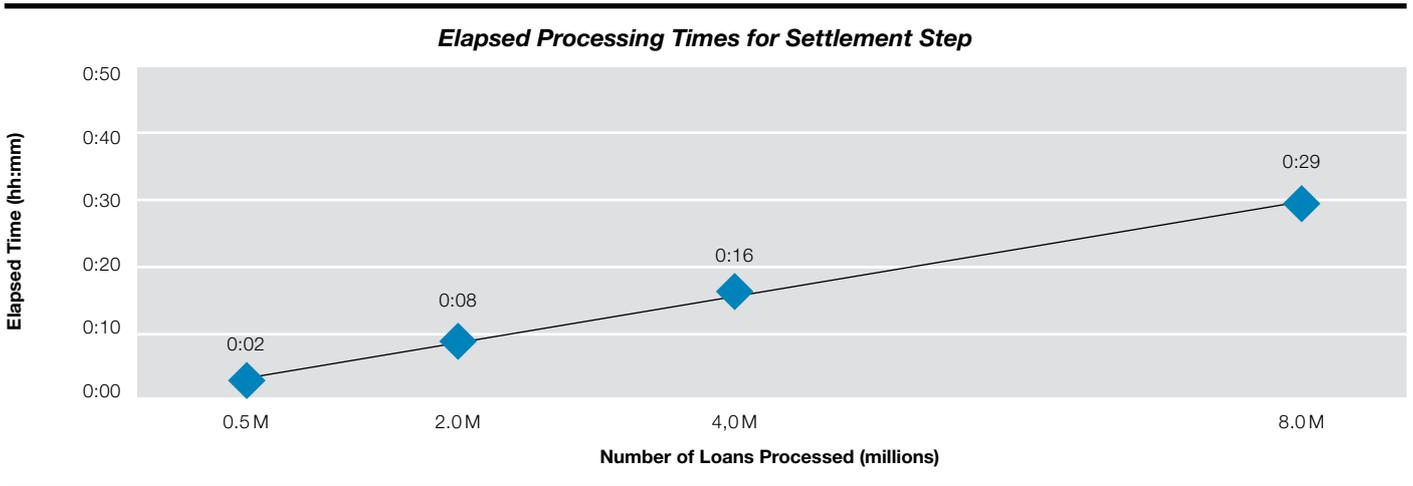
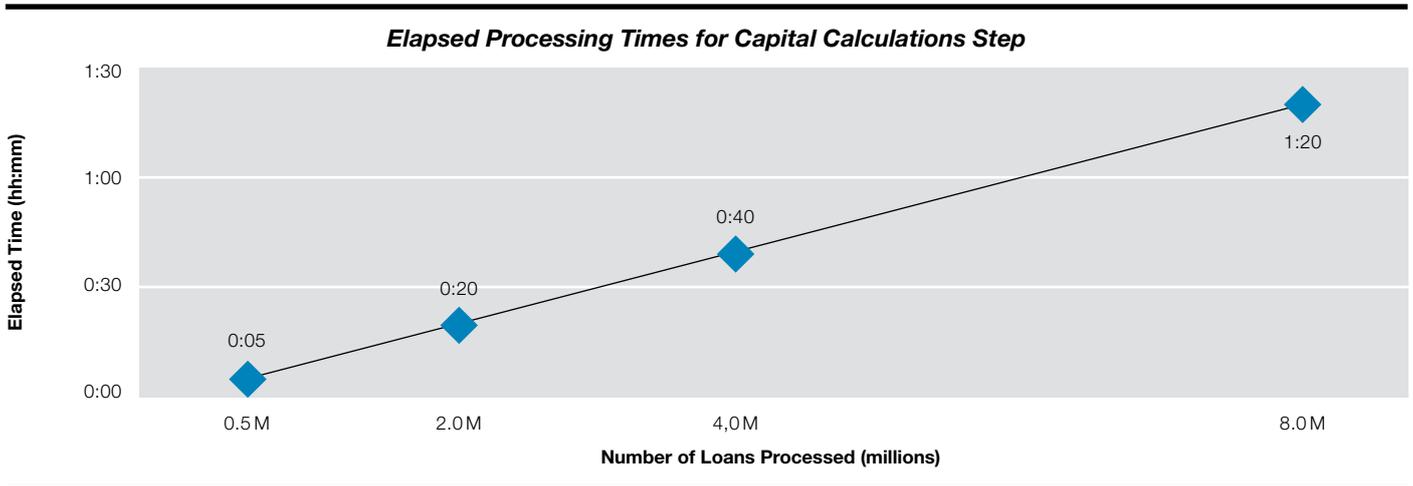


Figure 24. Capital Calculations Elapsed Times



11.3 Application Server on IBM Power System—Result details

The table shows the CPU, memory and network resource consumption for the test cycles: 500K, 2 million, 4 million, and 8 million loan accounts.

Table 17. Application Server Hardware Consumption

Application Server		Power 795 (4 LPARs), 256 cores, 4 GHz							Power 740 (1 LPAR), 4 cores, 3,55 GHz				
HW Resources consumed		Processing Batch and Dialog							Processing the Central Services				
		Cores				Memory	Network (r/w)		Cores		Memory	Network (r/w)	
Run	Volume	avg	max	avg %	max%	max TB	avg MB/s	max MB/s	avg	max	max GB	avg MB/s	max MB/s
Billing	500K	113	161	44%	63%	1.1	222	333	0.2	0.3	10	1.3	3.2
Invoicing	500K	29	50	11%	19%	1.1	120	202	0.1	0.1	10	1.9	3.7
Payment Distr	500K	137	200	53%	78%	1.1	194	289	1.2	1.9	10	7.0	15.7
Outbound	500K	5	7	2%	3%	1.1	44	70	0.1	0.4	10	2.8	9.0
Accrual	500K	79	159	31%	62%	1.1	113	228	0.4	0.7	10	2.4	4.7
Settlement	500K	73	144	29%	56%	1.1	129	264	0.3	0.5	10	1.7	3.3
Capital Calc	500K	158	223	62%	87%	1.1	92	129	0.2	0.4	10	0.8	1.5
Billing	2M	145	218	57%	85%	1.0	232	342	0.2	0.3	10	1.5	2.8
Invoicing	2M	34	50	13%	19%	1.0	139	213	0.1	0.1	10	2.2	4.0
Payment Distr	2M	138	203	54%	79%	1.0	190	301	1.3	1.6	10	6.7	13.0
Outbound	2M	6	8	2%	3%	1.0	90	142	0.1	0.4	10	2.1	5.8
Accrual	2M	153	209	60%	82%	1.0	185	258	0.7	1.0	10	4.2	6.1
Settlement	2M	143	194	56%	76%	1.0	203	291	0.5	0.7	10	3.0	4.1
Capital Calc	2M	158	218	62%	85%	1.0	83	121	0.2	0.4	10	0.7	1.6
Billing	4M	159	192	62%	75%	1.0	284	381	0.2	0.3	10	1.7	2.8
Invoicing	4M	41	55	16%	21%	1.0	167	231	0.1	0.1	10	2.7	4.1
Payment Distr	4M	133	204	52%	80%	1.0	174	305	1.2	1.7	10	6.0	14.3
Outbound	4M	9	13	4%	5%	1.0	162	283	0.1	0.5	10	3.1	7.4
Accrual	4M	152	209	59%	82%	1.0	221	308	0.7	0.9	10	4.1	6.0
Settlement	4M	155	200	60%	78%	1.0	243	334	0.5	0.7	10	3.1	4.1
Capital Calc	4M	170	225	66%	88%	1.0	98	141	0.2	0.3	10	0.7	1.5
Billing	8M	186	208	73%	81%	1.0	220	287	0.3	0.3	10	2.0	3.4
Invoicing	8M	32	48	12%	19%	1.0	131	219	0.0	0.2	10	2.1	3.9
Payment Distr	8M	140	202	55%	79%	1.0	225	338	1.5	2.3	10	5.7	16.5
Outbound	8M	10	17	4%	7%	1.0	214	418	0.3	2.0	10	2.9	9.8
Accrual	8M	168	208	66%	81%	1.0	222	279	0.7	0.9	10	4.6	6.0
Settlement	8M	165	192	64%	75%	1.0	255	301	0.6	0.7	10	3.5	4.4
Capital Calc	8M	152	225	59%	88%	1.0	82	130	0.2	0.3	10	0.6	1.5

Definition of the application server measurement values:

- **CPU** shows the avg/max values for the physical CPU utilization during each test step. The CPU figures are the sum (usr+sys) of the CPU used for the SAP application (usr) and the AIX operating system (sys) workload.
- **Mem** is the sum of all maximum allocated memory, including the OS.

We used the following calculation:

Total amount of dedicated memory – real free memory = allocated memory.

- **Network** represents the sum of read and write data rates in Mbytes/sec, for all network adapters in the system. This is the same as produced by the AIX command “netpmon – O dd”.

11.4 MIS4SAP – Monitors used

Table 18. Monitoring tools used

	Name	Type	Comment
SAP	enqt	interval	Get all enqueue server statistics from enqt process
	STAD	static	Get statistical records for the test user
	ST02 buffer	static	Get the buffer statistics
	ST10	static	Get table statistics
	Y_ST03	static	Get cumulated STAD statistics
	ST02 semaphore	static	Get statistics about the semaphores
	Profiles	static	Get all application server profiles
	SM21	static	Get SYSLOG entries
AIX	vmo	static	Get all virtual memory management parameters
	schedo	static	Get all scheduling OS parameters
	ioo	static	Gets all input and output OS options
	no	static	Get all networking options
	raso	static	Get all Reliability, Availability, Serviceability parameters
	nfso	static	Get all network filesystem options
	getsmtcfg	static	Get the smt configuration
	lscfg	static	Get the device configuration of the LPAR
	ps	static	Get process information
	vmstat	interval	Get virtual memory and cpu statistics during the test
	topas_nmon	interval	Get all statistics in nmon text output format
	lparstat	interval	Get LPAR statistics during the test
	mpstat	interval	Get multi processor statistics during the test
	iostat	interval	Get the IO operation's during the test
z/OS	RMF	interval	Monitors all System z resources

11.5 Software levels

For the performance test we used Banking Services from SAP 8.0 SP4, and for FI-CAX we used SAP ERP 6.0 Enhancement Package 5 SP 8.

We first installed the application software components on the Banking Services and FI-CAX system. For configuration set-up we used SAP Loans Management for Banking Rapid-Deployment Solution V2.80 (Loans RDS).

In addition, we implemented several SAP Notes before and during the test (see 11.6 SAP Notes for details).

Table 19. Software components for the Banking Services system

Product	Comments	Components
AIX 7.1	TL1 SP4	
SAP Kernel 7.20 EXT 64 Bit Unicode for AIX 7.1	Support Package level 411	
DBSL Patch No. 432		
DB2 Connect driver DB2 v9.7.0.5		
SAP NetWeaver 7.1 ABAP Enhancement Package 1	Support package stack 10	SAP_ABA 711SP10 SAP_BASIS 711SP10 PI_BASIS 711SP10 SAP_BW 711SP10
Banking Services from SAP 8.0	Support package 4	FINBASIS 700SP19 FSAPPL 400SP4 BI_CONT 711SP2 SEM-BW 700SP19
Components required by Loans RDS		BP-SOLBLD 71V2 BP-CANW 711V4 BP-BNK 800V2
Service tools		ST-A/PI 01P_710

Table 20. Components for FI-CAX system

Product version	Comments	Components
AIX 7.1	TL1 SP4	
SAP Kernel 7.20 EXT 64 Bit Unicode for AIX 7.1	Support Package level 411	
DBSL Patch No. 432		
DB2 Connect driver DB2 v9.7.0.5		
SAP NetWeaver 7.1 ABAP Enhancement Package 2	Support package stack 11	SAP_ABA 702SP11 SAP_BASIS 702SP11 PI_BASIS 702SP11 SAP_BS_FND 702SP9 SAP_BW 702SP11 SAP_AP 700SP27
SAP ERP 6.0 Enhancement Package 5 for Leasing/Contract A/R & A/P	Support package stack 8	SAP_APPL 605SP8 EA-APPL 605SP8 FI-CA 605SP8 FI-CAX 605SP8
Components required by Loans RDS		BP-SOLBLD 70V7 BP-CANW 702V5 BP-ERP 605V8
Service tools		ST-A/PI 01P_700

In addition, we installed the “Mass Data Generator” and the “Contract Lifecycle Tool” from SAP AGS for data generation and automated payment of open items. We initially set up DB2 version 10 dated January 2013. The final tests for the 8 million accounts were performed with a DB2 version 10 dated June 2013.

Table 21. Database components

Product version	Comments
z/OS release 01.13.00	R1.13
DB2 10 dated June 2013	

11.6 SAP Notes implemented

The following SAP Notes were implemented before and during the performance test. The test cycles for 8 million accounts were performed with the complete set of SAP Notes.

The test cycles for 4 million accounts were performed without notes 1872859 and 1879043. The test cycles for 500,000 and 2 million accounts were performed without notes 1851784, 1852036, 1741452, 1702200, 1855448, 1872859 and 1879043.

Table 22. SAP Notes implemented for the Banking Services system

Note	Relevant for performance
1592011 Error message is displayed when not all tracks are due	
1592933 Banking Exception Monitor corrections	
1610716 Correcting runtime objects with incorrect alignment	
1660424 PPC data and cashflow ratio data does not match	
1668456 DTP: Error msg RSO851 during upgrade or Content installation	
1670473 PPF: Dump SAPSQL_UPTO_NEGATIVE during asynchronous pack. Gen	
1706321 DVM Scoping Service: Turn off DANA processing	
1709946 Massman: ST-API release P corrections (general)	
1711284 PPF: Unnecessary msgs BANK_PP/601 and BANK_JC/006 in job log	
1728635 Solution Builder (BP-SOLBLD 71V2) Hotfixes 001	
1733268 Next execution date is incorrect after account change	
1734987 BCA_CAPITAL_CALC: Long runtime for account selection	yes
1738410 Correction for delivery Customizing of transaction types	yes
1742986 New Job distribution method for PPF	yes
1744135 Select which capitals to display in Tx. BCA_CN_KFG	
1745794 Data Collectors corrections ST-A/PI 01P* SP0	
1746131 Performance: Payment processing creates many RFC-calls	yes
1748775 Payment processing creating PDIs on single item level	yes
1753145 PPF: No processing during asynchronous package creation	
1757564 BCA_CAPITAL_CALC: No capital 'Interest accrued, not settled'	
1760389 Reimbursement after correction of a paid receivable	
1762945 Billing refund: Logic in cash flow	
1780254 Next settlement date is incorrect after calendar change	
1784133 Settlement payment order without item number	
1799956 Performance issues in clearing notification from FPOITR	yes
1800477 PPO order repeat fails with error BCA_CNBP_ACCT090	
1803592 Performance: balance buffer not cleared for Cashflow/Billing	
1805792 Perf: Balances for settlement; composite SAP Note	yes
1806952 Settlement package discarded due to reject of payment order	
1807760 ERC: Error when creating multiple contracts in single LUW	
1813979 Poor performance due to unbuffered table FEHT_ACTIVE	yes
1814232 Performance: Acct settlement mass run – composite SAP note	yes
1815542 Performance: Account settlement mass run, BKK92_POSTINGS	
1815792 BTE implementations are incorrectly activated	
1834303 Payment Distribution Performance Improvement	yes
1855448 Reading BCA_BL_ARCH_DATE table in the billing run	yes
1872859 Performance issue in the billing run	yes
1887627 Suppression of change documents in Payment Distribution	yes

Table 23. SAP Notes implemented for the FI-CAX system

Note number	Relevant for performance
1702200 FPOITR: No trigger entries for reversal of payments	
1741452 RFKKOI_START_ERROR_MODE: Processing of status 3	
1804627 Issue in repeating the PPO order for billing document create	
1821516 Errors in UKMS mapping issue during Payment Distribution	
1826790 Billing documents missing in FI-CAX system	
1851784 FPOITR: Performance improvement	yes
1852036 FPOITR: Performance improvement - deletion of old entries	yes
1879043 PPO order created for billing documents posted in FI-CAX	

11.7 Bufferpool Sizes

Table 24. Bufferpool sizes for FI-CAX and BaS system

FI-CAX		BaS	
Bufferpool	Size	Bufferpool	Size
BPOOL(BP0)	VPSIZE(20000)	BPOOL(BP0)	VPSIZE(40000)
BPOOL(BP2)	VPSIZE(320000)	BPOOL(BP2)	VPSIZE(640000)
BPOOL(BP3)	VPSIZE(480000)	BPOOL(BP3)	VPSIZE(640000)
BPOOL(BP40)	VPSIZE(15000)	BPOOL(BP40)	VPSIZE(25000)
BPOOL(BP22)	VPSIZE(320000)	BPOOL(BP42)	VPSIZE(320000)
BPOOL(BP23)	VPSIZE(720000)	BPOOL(BP43)	VPSIZE(320000)
BPOOL(BP24)	VPSIZE(320000)	BPOOL(BP44)	VPSIZE(640000)
BPOOL(BP25)	VPSIZE(480000)	BPOOL(BP45)	VPSIZE(320000)
BPOOL(BP26)	VPSIZE(320000)	BPOOL(BP46)	VPSIZE(480000)
BPOOL(BP27)	VPSIZE(480000)	BPOOL(BP47)	VPSIZE(640000)
BPOOL(BP28)	VPSIZE(320000)	BPOOL(BP48)	VPSIZE(320000)
BPOOL(BP29)	VPSIZE(320000)	BPOOL(BP49)	VPSIZE(320000)
BPOOL(BP30)	VPSIZE(320000)	BPOOL(BP20)	VPSIZE(320000)
BPOOL(BP31)	VPSIZE(320000)	BPOOL(BP21)	VPSIZE(320000)
BPOOL(BP32)	VPSIZE(480000)	BPOOL(BP22)	VPSIZE(480000)

FI-CAX		BaS	
Bufferpool	Size	Bufferpool	Size
BPOOL(BP33)	VPSIZE(480000)	BPOOL(BP23)	VPSIZE(320000)
BPOOL(BP34)	VPSIZE(320000)	BPOOL(BP24)	VPSIZE(640000)
BPOOL(BP35)	VPSIZE(480000)	BPOOL(BP25)	VPSIZE(320000)
BPOOL(BP36)	VPSIZE(320000)	BPOOL(BP26)	VPSIZE(480000)
BPOOL(BP37)	VPSIZE(480000)	BPOOL(BP27)	VPSIZE(320000)
BPOOL(BP8K2)	VPSIZE(320000)	BPOOL(BP28)	VPSIZE(320000)
BPOOL(BP8K0)	VPSIZE(60000)	BPOOL(BP29)	VPSIZE(320000)
BPOOL(BP8K1)	VPSIZE(60000)	BPOOL(BP30)	VPSIZE(640000)
BPOOL(BP16K0)	VPSIZE(6000)	BPOOL(BP31)	VPSIZE(640000)
BPOOL(BP16K1)	VPSIZE(6000)	BPOOL(BP32)	VPSIZE(320000)
BPOOL(BP32K)	VPSIZE(80000)	BPOOL(BP33)	VPSIZE(320000)
BPOOL(BP32K1)	VPSIZE(80000)	BPOOL(BP35)	VPSIZE(9600000)
BPOOL(BP32K2)	VPSIZE(80000)	BPOOL(BP36)	VPSIZE(320000)
BPOOL(BP32K3)	VPSIZE(80000)	BPOOL(BP37)	VPSIZE(320000)
		BPOOL(BP8K0)	VPSIZE(60000)
		BPOOL(BP8K1)	VPSIZE(60000)
		BPOOL(BP8K2)	VPSIZE(300000)
		BPOOL(BP16K0)	VPSIZE(4000)
		BPOOL(BP16K1)	VPSIZE(4000)
		BPOOL(BP32K)	VPSIZE(40000)
		BPOOL(BP32K1)	VPSIZE(40000)
		BPOOL(BP32K4)	VPSIZE(30000)
		BPOOL(BP32K5)	VPSIZE(30000)

11.8 Database Bufferpool Isolation

Table 25. Database bufferpool isolation for FI-CAX

FI-CAX			
Bufferpools	Database	Tablespace	Table/Index
BP24	DSN06608	DFKKRTRI	DFKK_TRIGGER_CL
BP26	DSN04534	DFKKRTRI	DFKK_TRIGGER_PAY
BP28	DSN07683	DFKKINVR	DFKKINV_TRIG
BP30	DSN03059	DFKKINVD	DFKKINVD0C_H
BP32	DSN04159	DFKKINVB	DFKKINVBILL_H
BP32K2	DSN05515	ARFCSDAT	ARFCSDATA
BP34	DSN06265	F1UKMFKE	/1UKM/KEYFCMM0
BP36	DSN04920	F1UKMFMG	/1UKM/MGPFCMM0
BP8K2	DSN00430	DFKKOP	M_RPLNA
BP8K2	DSN00430	DFKKOP	U_13604
BP8K2	DSN00430	DFKKOP	V_DFKKOP_EURO
BP8K2	DSN00430	DFKKOP	V_DFKKOP_RS
BP8K2	DSN00430	DFKKOP	V6_FKKOP
BP8K2	DSN00430	DFKKOP	E1_FKKOP
BP8K2	DSN00430	DFKKOP	M_RPLNB
BP8K2	DSN00430	DFKKOP	V2_FKKOP
BP8K2	DSN00430	DFKKOP	DFKKOP
BP8K2	DSN00430	DFKKOP	U_13707
BP8K2	DSN00430	DFKKOP	V1_FKKOP
BP8K2	DSN00430	DFKKOP	V3_FKKOP
BP8K2	DSN00430	DFKKOP	V7_FKKOP
BP8K2	DSN00430	DFKKOP	U_13679
BP8K2	DSN00430	DFKKOP	V_EWUDFKKOPEURO
BP8K2	DSN00430	DFKKOP	V4_FKKOP
BP8K2	DSN00430	DFKKOP	V5_FKKOP
BP8K2	DSN00430	DFKKOP	U_13635
BP23	DSN00430		DFKKOP~6
BP23	DSN00430		DFKKOP~5
BP23	DSN00430		DFKKOP~4

FI-CAX			
Bufferpools	Database	Tablespace	Table/Index
BP23	DSN00430		DFKKOP~3
BP23	DSN00430		DFKKOP~2
BP23	DSN00430		DFKKOP~1
BP23	DSN00430		DFKKOP~0
BP25	DSN06608		DFKK_TRIGGER_CL~1
BP25	DSN06608		DFKK_TRIGGER_CL~0
BP27	DSN04534		DFKK_TRIGGER_PAY~0
BP27	DSN04534		DFKK_TRIGGER_PAY~2
BP27	DSN04534		DFKK_TRIGGER_PAY~1
BP29	DSN04534		DFKKINV_TRIG~3
BP29	DSN07683		DFKKINV_TRIG~2
BP29	DSN07683		DFKKINV_TRIG~1
BP29	DSN07683		DFKKINV_TRIG~0
BP31	DSN07683		DFKKINVDOK_H~003
BP31	DSN03059		DFKKINVDOK_H~0
BP31	DSN03059		DFKKINVDOK_H~002
BP31	DSN03059		DFKKINVDOK_H~001
BP32K3	DSN05515		ARFCSDATA~0
BP33	DSN04159		DFKKINVBILL_H~REF
BP33	DSN04159		DFKKINVBILL_H~0
BP33	DSN04159		DFKKINVBILL_H~KTO
BP33	DSN04159		DFKKINVBILL_H~GPT
BP35	DSN06265		/1UKM/KEYFCMM0~RLM
BP35	DSN06265		/1UKM/KEYFCMM0~KID
BP35	DSN06265		/1UKM/KEYFCMM0~CCT
BP35	DSN06265		/1UKM/KEYFCMM0~ATT
BP35	DSN06265		/1UKM/KEYFCMM0~0
BP37	DSN04920		/1UKM/MGPFCCMM0~0
BP37	DSN04920		/1UKM/MGPFCCMM0~OBJ

Table 26. Database bufferpool isolation for BaS system

BaS			
Bufferpools	Database	Tablespace	Table/Index
BP20	DSN00584	ARFCRSTA	ARFCRSTATE
BP22	DSN01766	BCARCNR	V_F4_PAYREF_AH
BP22	DSN01766	BCARCNR	BCA_CN_LINK
BP22	DSN01766	BCARCNR	BCA_VIEW_FICOND
BP26	DSN01944	BCARPXAC	BCA_PXAC_EXT_ACC
BP28	DSN01782	BCARCNSP	V_F4_ACCOUNT_STA
BP28	DSN01782	BCARCNSP	V_ACCOUNTPLMDOC
BP28	DSN01782	BCARCNSP	V_ACCOUNTINDCOND
BP28	DSN01782	BCARCNSP	V_F4_ACCOUNT_INT
BP28	DSN01782	BCARCNSP	V_F4_ACCOUNT_CRE
BP28	DSN01782	BCARCNSP	BCA_CNSP_ACCT
BP28	DSN01782	BCARCNSP	V_F4_ACCOUNT_EXT
BP28	DSN01782	BCARCNSP	V_BCA_BL_TECHID
BP28	DSN01782	BCARCNSP	V_BCA_SO_SONO
BP30	DSN01761	BCARTRAN	VBCA_AR_FGRS_PBP
BP30	DSN01761	BCARTRAN	VBCA_AR_FGRS_UPD
BP30	DSN01761	BCARTRAN	BCA_TRANSFIG
BP32	DSN04040	BUT000	SUID_SH_BPPERSON
BP32	DSN04040	BUT000	/BA1/FSBP_SHLP01
BP32	DSN04040	BUT000	SRM_M_BUPAG
BP32	DSN04040	BUT000	M_BUPAA_VERSP
BP32	DSN04040	BUT000	V_BUT020_BW_VER
BP32	DSN04040	BUT000	SRM_M_BUPAR
BP32	DSN04040	BUT000	USS_BPADRC_VIEW
BP32	DSN04040	BUT000	U_11103
BP32	DSN04040	BUT000	V_BUT021_FS_BW
BP32	DSN04040	BUT000	M_BUPAR
BP32	DSN04040	BUT000	M_BUPAA
BP32	DSN04040	BUT000	M_BUPAG
BP32	DSN04040	BUT000	M_BUPAA_VERS
BP32	DSN04040	BUT000	V_BUT020_BW
BP32	DSN04040	BUT000	U_11129
BP32	DSN04040	BUT000	U_11105
BP32	DSN04040	BUT000	U_11102
BP32	DSN04040	BUT000	USS_BPSONST_VIEW
BP32	DSN04040	BUT000	SRM_M_BUPAP
BP32	DSN04040	BUT000	BUT000
BP32	DSN04040	BUT000	V_BUT100_BW_ORG

BaS			
Bufferpools	Database	Tablespace	Table/Index
BP32	DSN04040	BUT000	U_11100
BP32	DSN04040	BUT000	U_11104
BP32	DSN04040	BUT000	USS_JOIN_STH_00
BP32	DSN04040	BUT000	SUID_SH_ORGADDR
BP32	DSN04040	BUT000	M_BUPAP
BP32	DSN04040	BUT000	M_BUPAB
BP32	DSN04040	BUT000	M_BUPAA_DUMMY
BP32	DSN04040	BUT000	V_BUPAEXST
BP32	DSN04040	BUT000	V_DB_BUT
BP32	DSN04040	BUT000	M_BUPA_EXIT
BP32	DSN04040	BUT000	USS_BP_VIEW
BP32	DSN04040	BUT000	M_BUPAA_VERSC
BP32K4	DSN00587	ARFCSDAT	ARFCSDATA
BP36	DSN01764	BCARPREN	V_PRENOTE_KEY
BP36	DSN01764	BCARPREN	BCA_PRENOTE
BP36	DSN01764	BCARPREN	VBCA_AR_PRNO_UPD
BP36	DSN01764	BCARPREN	VBCA_AR_PRNO_SEL
BP36	DSN01764	BCARPREN	V_BCA_PRENOTE_U
BP42	DSN01784	BCARBLRS	BCA_BL_SCHED
BP44	DSN01767	BCARBLRI	BCA_BL_ITEM
BP46	DSN00006	FFSPDFRE	/FSPD/RESULT
BP48	DSN01776	FFSPDFIT	/FSPD/ITEM
BP48	DSN01776	FFSPDFIT	/FSPD/V_ITEM_UPD
BP8K2	DSN01760	BCARPAYM	VBCA_AR_PAYM_PBP
BP8K2	DSN01760	BCARPAYM	V_BCA_PAYMITEM_U
BP8K2	DSN01760	BCARPAYM	V_BCA_PAYMITEM_A
BP8K2	DSN01760	BCARPAYM	VBCA_AR_PAYM_UPD
BP8K2	DSN01760	BCARPAYM	BCA_PAYMITEM
BP8K2	DSN01760	BCARPAYM	VBCA_AR_PAYM_SEL
BP21	DSN00584		ARFCRSTATE~1
BP21	DSN00584		ARFCRSTATE~0
BP23	DSN01766		BCA_CN_LINK~S02
BP23	DSN01766		BCA_CN_LINK~S01
BP23	DSN01766		BCA_CN_LINK~0
BP25	DSN01791		BCA_CN_EV_ACBAL~0
BP27	DSN01944		BCA_PXAC_EXT_ACC~0
BP27	DSN01944		BCA_PXAC_EXT_ACC00
BP29	DSN01782		BCA_CNSP_ACCT~S02

Table 26. Database bufferpool isolation for BaS system

BaS			
Bufferpools	Database	Tablespace	Table/Index
BP29	DSN01782		BCA_CNSP_ACCT~S02
BP29	DSN01782		BCA_CNSP_ACCT~IBA
BP29	DSN01782		BCA_CNSP_ACCT~BAC
BP29	DSN01782		BCA_CNSP_ACCT~0
BP31	DSN01761		BCA_TRANSFIG~0
BP32K5	DSN00587		ARFCSDATA~0
BP33	DSN04040		BUT000~0
BP33	DSN04040		BUT000~EXT
BP33	DSN04040		BUT000~ADR
BP33	DSN04040		BUT000~TOD
BP33	DSN04040		BUT000~SRT
BP33	DSN04040		BUT000~NAM
BP33	DSN04040		BUT000~GUI
BP35	DSN01760		BCA_PAYMITEM~S03
BP35	DSN01760		BCA_PAYMITEM~S02
BP35	DSN01760		BCA_PAYMITEM~S01
BP35	DSN01760		BCA_PAYMITEM~0
BP35	DSN01760		BCA_PAYMITEM~S04
BP37	DSN01764		BCA_PRENOTE~001
BP37	DSN01764		BCA_PRENOTE~0
BP43	DSN01784		BCA_BL_SCHED~EXT
BP43	DSN01784		BCA_BL_SCHED~0
BP45	DSN01767		BCA_BL_ITEM~REF
BP45	DSN01767		BCA_BL_ITEM~EXT
BP45	DSN01767		BCA_BL_ITEM~AST
BP45	DSN01767		BCA_BL_ITEM~0
BP47	DSN00006		/FSPD/RESULT~001
BP47	DSN00006		/FSPD/RESULT~0
BP49	DSN01776		/FSPD/ITEM~003
BP49	DSN01776		/FSPD/ITEM~0
BP49	DSN01776		/FSPD/ITEM~002
BP49	DSN01776		/FSPD/ITEM~001

The following are the largest data tables in the FI-CAX system. This table illustrates the effect of data aging specifically on the DFKK* tables. This data was collected after data generation and aging of the 8 million loans.

Table 27. Largest FI-CAX tables after 8 million created and aged loans

FI-CAX tables	Number of rows
BUT000	8,250,000
FKKVKP	8,250,000
DFKKKO ^a	224,000,000
DFKKOPK ^b	224,000,000

FI-CAX tables	DATE_FROM	Number of rows
DFKKINVBILL_H	20000107	1,256,633
	20000114	1,250,000
	20000121	1,250,000
	20000128	1,250,000
	20000204	1,250,000
	20000211	1,250,000
	20000218	5,000,000
	20000225	5,000,000
	20000303	5,000,000
	20000310	10,000,000
	20000317	10,000,000
	20000324	10,000,000
	20000331	20,000,000
	20000407	20,000,000
	20000414	20,000,000

^a Estimate, data not recorded

^b Estimate, data not recorded

Table 27. Largest FI-CAX tables after 8 million created and aged loans

DFKKINVDOC_H	FAEDN	
	20000107	500,000
	20000114	502,211
	20000121	500,000
	20000128	500,000
	20000204	500,000
	20000211	500,000
	20000218	2,000,000
	20000225	2,000,000
	20000303	2,000,000
	20000310	4,000,000
	20000317	4,000,000
	20000324	4,000,000
	20000331	8,000,000
	20000407	8,000,000
	20000414	8,000,000
DFKKOP	ABRZU	
	20000107	1,256,633
	20000114	1,250,000
	20000121	1,250,000
	20000128	1,250,000
	20000204	1,250,000
	20000211	1,250,000
	20000218	5,000,000
	20000225	5,000,000
	20000303	5,000,000
	20000310	10,000,000
	20000317	10,000,000
	20000324	10,000,000
	20000331	20,000,000
	20000407	20,000,000
	20000414	20,000,000

11.9 BaS Profile Parameters

11.9.1 Default Profile

```
#####  
#.*   Default profile DEFAULT  
#####  
rstr/max_filesize_MB = 100  
#ES/TABLE = SHM_SEGS  
#ES/SHM_SEGS_VERSION = 2  
SAPDBHOST = vhost2AO  
j2ee/dbtype = db2  
j2ee/dbname = LD01  
j2ee/dbhost = vhost2AO  
SAPSYSTEMNAME = LAP  
SAPGLOBALHOST = is04d1p  
system/type = ABAP  
#APFQDN = wdf.sap.corp  
#APLOCALHOSTFULL = $(SAPLOCALHOST).$(SAPFQDN)  
is/HTTP/show_detailed_errors = FALSE  
icf/user_recheck = 1  
gw/acl_mode = 0  
icm/HTTP/ASJava/disable_url_session_tracking = TRUE  
service/protectedwebmethods = SDEFAULT  
rdisp/bufrefmode = sendon,exeauto  
ssf/name = SAPSECULIB  
#-----  
# SAP Message Server for ABAP  
#-----  
#-----  
# SAP Messaging Service parameters are set in the DEFAULT.PFL  
#-----  
# SAP Messaging Service for ABAP  
# Standalone Enqueue parameters  
#-----
```

rdisp/mshost = is04d1p
rdisp/msserv = sapmsLAP
rdisp/msserv_internal = 3950
enqueue/process_location = REMOTESA
enqueue/serverhost = is04d1p
enqueue/serverinst = 50
enqueue/con_retries = 120
enqueue/backup_disabled = 1

rdisp/max_wprun_time = 0
login/system_client = 001
rdisp/TRACE = 1
ipc/shm_psize_10 = 984000000
abap/heaplimit = 500000000
abap/buffersize = 700000
abap/heap_area_total = 5000000000
abap/heap_area_nondia = 5000000000
ztta/roll_area = 10000000
abap/shared_objects_size_MB = 100
zcsa/table_buffer_area = 120000000
rdisp/rfc_max_login = 100
rdisp/rfc_max_own_login = 100
rdisp/rfc_max_own_used_wp = 100
rdisp/rfc_max_queue = 100
rdisp/rfc_min_wait_dia_wp = 1
rdisp/rfc_use_quotas = 1
rdisp/wp_no_dia = 20
rdisp/wp_no_btc = 16
rdisp/ROLL_SHM = 65536
rdisp/ROLL_MAXFS = 65536
rdisp/PG_SHM = 131072
rdisp/PG_MAXFS = 262144
rdisp/max_wprun_time = 3600
rdisp/max_arq = 2000
rdisp/elem_per_queue = 100000
rdisp/max_comm_entries = 2000

rdisp/bufreftime = 120
rdisp/tm_max_no = 10000
rdisp/appc_ca_blk_no = 10000
gw/reg_info = /usr/sap/LAP/log/reginfo
gw/reg_no_conn_info = 0
gw/sec_info = /usr/sap/LAP/log/secinfo
ztta/roll_extension = 8000000000
ztta/roll_extension_dia = 8000000000
ztta/roll_extension_nondia = 4000000000
em/initial_size_MB = 160000
gw/max_overflow_size = 99900000
gw/cpic_timeout = 60
gw/max_conn = 4000
gw/max_sys = 2000
gw/max_overflow_usage = 80
gw/req_stack_size = 150
rsdb/obj/large_object_size = 8192

11.9.2 Central Instance Profile

SAPSYSTEMNAME = LAP
SAPSYSTEM = 40
INSTANCE_NAME = DVEBMGS40
DIR_CT_RUN = \$(DIR_EXE_ROOT)/\$(OS_UNICODE)/rs6000_64
DIR_EXECUTABLE = \$(DIR_INSTANCE)/exe
DIR_PROFILE = \$(DIR_INSTALL)/profile
_PF = \$(DIR_PROFILE)/LAP_DVEBMGS40_sapa3a05
SETENV_00 = DIR_LIBRARY=\$(DIR_LIBRARY)
SETENV_01 = LD_LIBRARY_PATH=\$(DIR_LIBRARY):%(LD_LIBRARY_PATH)
SETENV_02 = SHLIB_PATH=\$(DIR_LIBRARY):%(SHLIB_PATH)
SETENV_03 = LIBPATH=\$(DIR_LIBRARY):%(LIBPATH)
SETENV_04 = PATH=\$(DIR_EXECUTABLE):%(PATH)
SETENV_05 = DB2_CLI_DRIVER_INSTALL_PATH=\$(DIR_EXECUTABLE)/
db2_clidriver

```

# Parameter Added by DELESSIO for AIX Large Page use.
SETENV_06 = LDR_CNTRL=LARGE_PAGE_DATA=Y@TEXTPSIZE=64K@
STACKPSIZE=64K@SHMPSIZE=64K@NAMEDSHLIB64=sap0,16M,FIRST_TOUCH
# SAP Message Server parameters are set in the DEFAULT.PFL
#-----
ms/server_port_0 = PROT=HTTP,PORT=81$$
rdisp/wp_no_enq = 0
rdisp/wp_no_vb = 1
rdisp/wp_no_vb2 = 1
rdisp/wp_no_spo = 1
dbms/type = db2
dbs/db2/schema = SAPR3
dbs/db2/ssid = LD01
dbs/db2/hosttcp = vhost2AO
dbs/db2/user = R3USER
rsdb/max_blocking_factor = 10
rsdb/max_in_blocking_factor = 35
rsdb/min_blocking_factor = 3
rsdb/min_in_blocking_factor = 6
rsdb/prefer_fix_blocking = 1
rsdb/prefer_union_all = 1
rsdb/prefer_in_itab_opt = 1
ipc/shm_psize_10 = 1560000000
ipc/shm_psize_40 = 112000000
rfc/use_wp_instead_of_pid = 1600
gw/max_overflow_size = 99900000
gw/cpic_timeout = 60
gw/max_conn = 4000
gw/max_sys = 2000

```

11.9.3 Enqueue Profile

```
SAPSYSTEMNAME = LAP
SAPSYSTEM = 50
INSTANCE_NAME = ASCS50
DIR_CT_RUN = $(DIR_EXE_ROOT)/$(OS_UNICODE)/rs6000_64
DIR_EXECUTABLE = $(DIR_INSTANCE)/exe
SAPLOCALHOST = is04d1p
DIR_PROFILE = $(DIR_INSTALL)/profile
#-----
# SAP Message Server parameters are set in the DEFAULT.PFL
#-----
ms/standalone = 1
ms/server_port_0 = PROT=HTTP,PORT=81$$
#-----
# SAP Enqueue Server
#-----
enqueue/table_size = 409600
enqueue/snapshot_pck_ids = 1600
enqueue/server/max_query_requests = 5000
enqueue/server/max_requests = 5000
enqueue/server/max_clients = 5000
enqueue/async_req_max = 5000
enqueue/server/threadcount = 4
enqueue/encni/threadcount = 4
rdisp/enqname = $(rdisp/myname)
```

11.9.4 Dialogue Profile

```
#####  
#.* Instance profile LAP_D41_SAPA3A05  
#####  
rfc/use_wp_instead_of_pid = 1  
SAPSYSTEMNAME = LAP  
SAPSYSTEM = 41  
INSTANCE_NAME = D41
```

11.10 FI-CAX Profile Parameters

11.10.1 Default Profile

```
#####  
#.* Default profile DEFAULT  
#####  
rstr/max_filesize_MB = 100  
SAPDBHOST = vhost2BO  
j2ee/dbtype = db2  
j2ee/dbname = JD01  
j2ee/dbhost = vhost2BO  
SAPSYSTEMNAME = JAP  
SAPGLOBALHOST = is04d3p  
system/type = ABAP  
#APFQDN = wdf.sap.corp  
#APLOCALHOSTFULL = $(SAPLOCALHOST).$(SAPFQDN)  
is/HTTP/show_detailed_errors = FALSE  
icf/user_recheck = 1  
gw/acl_mode = 0  
icm/HTTP/ASJava/disable_url_session_tracking = TRUE  
service/protectedwebmethods = SDEFAULT  
ssf/name = SAPSECULIB  
ipc/shm_psize_10 = 984000000  
ipc/shm_psize_40 = 112000000
```

```
#-----  
# SAP Message Server for ABAP  
#-----  
#-----  
# SAP Messaging Service parameters are set in the DEFAULT.PFL  
#-----  
# SAP Messaging Service for ABAP  
# Standalone Enqueue parameters  
#-----  
rdisp/mshost = is04d3p  
rdisp/msserv = sapmsJAP  
rdisp/msserv_internal = 3930  
enqueue/process_location = REMOTESA  
enqueue/serverhost = is04d3p  
enqueue/serverinst = 30  
enqueue/con_retries = 120  
enqueue/backup_disabled = 1  
  
login/system_client = 001  
abap/heaplimit = 500000000  
abap/buffersize = 700000  
abap/heap_area_total = 5000000000  
abap/heap_area_nondia = 5000000000  
ztta/roll_area = 10000000  
abap/shared_objects_size_MB = 100  
zcsa/table_buffer_area = 120000000  
rdisp/rfc_max_login = 100  
rdisp/rfc_max_own_login = 100  
rdisp/rfc_max_own_used_wp = 100  
rdisp/rfc_max_queue = 100
```

rdisp/rfc_min_wait_dia_wp = 1
rdisp/rfc_use_quotas = 1
rdisp/wp_no_dia = 18
rdisp/wp_no_btc = 12
rdisp/ROLL_SHM = 65536
rdisp/ROLL_MAXFS = 65536
rdisp/PG_SHM = 131072
rdisp/PG_MAXFS = 262144
rdisp/max_wprun_time = 3600
rdisp/max_arq = 2000
rdisp/elem_per_queue = 100000
rdisp/max_comm_entries = 2000
rdisp/bufreftime = 120
rdisp/tm_max_no = 10000
rdisp/appc_ca_blk_no = 10000
gw/reg_info = /usr/sap/LAP/log/reginfo
gw/reg_no_conn_info = 0
gw/sec_info = /usr/sap/LAP/log/secinfo
ztta/roll_extension = 8000000000
ztta/roll_extension_dia = 8000000000
ztta/roll_extension_nondia = 4000000000
em/initial_size_MB = 160000
gw/max_overflow_size = 99900000
gw/cpic_timeout = 60
gw/max_conn = 4000
gw/max_sys = 2000
gw/max_overflow_usage = 80
gw/req_stack_size = 150
rsdb/obj/large_object_size = 8192
ssl/ssl_lib = \$(DIR_EXECUTABLE)

11.10.2 Central Instance Profile

```
#####  
#.  
Instance profile JAP_DVEBMGS20_SAPA3A07  
#.  
#####  
rfc/use_wp_instead_of_pid = 3300  
SAPSYSTEMNAME = JAP  
SAPSYSTEM = 20  
INSTANCE_NAME = DVEBMGS20  
DIR_CT_RUN = $(DIR_EXE_ROOT)/run  
DIR_EXECUTABLE = $(DIR_INSTANCE)/exe  
exe/saposcol = $(DIR_CT_RUN)/saposcol  
# sec/libsapsecu = $(ssl/ssl_lib)  
# ssf/ssfapi_lib = $(ssl/ssl_lib)  
rdisp/wp_no_dia = 50  
rdisp/wp_no_btc = 5  
exe/icmbnd = $(DIR_CT_RUN)/icmbnd  
# SAP Message Server parameters are set in the DEFAULT.PFL  
ms/server_port_0 = PROT=HTTP,PORT=81$$  
rdisp/wp_no_enq = 0  
rdisp/wp_no_vb = 1  
rdisp/wp_no_vb2 = 1  
rdisp/wp_no_spo = 1  
dbms/type = db2  
dbs/db2/schema = SAPR3  
dbs/db2/ssid = JD01  
dbs/db2/hosttcp = vhost2BO  
dbs/db2/user = R3USER  
rsdb/max_blocking_factor = 10  
rsdb/max_in_blocking_factor = 35  
rsdb/min_blocking_factor = 3  
rsdb/min_in_blocking_factor = 6  
rsdb/prefer_fix_blocking = 1  
rsdb/prefer_union_all = 1  
rsdb/prefer_in_itab_opt = 1
```

```

#ssl/ssl_lib = $(DIR_EXECUTABLE)$ (DIR_SEP)$ (FT_DLL_PREFIX)
sapcrypto$(FT_DLL)
#ssl/ssl_lib= $(DIR_EXECUTABLE)
ipc/shm_psize_10 = 1560000000
ipc/shm_psize_40 = 1120000000
gw/max_overflow_size = 999000000
gw/cpic_timeout = 60
gw/max_conn = 4000
gw/max_sys = 2000

```

11.10.3 Enqueue Profile

```

#.*****
#.* Instance profile JAP_ASCS30_is04d3p
#.*****
SAPSYSTEMNAME = JAP
SAPSYSTEM = 30
INSTANCE_NAME = ASCS30
DIR_CT_RUN = $(DIR_EXE_ROOT)/run
DIR_EXECUTABLE = $(DIR_INSTANCE)/exe
SAPLOCALHOST = is04d3p
DIR_PROFILE = $(DIR_INSTALL)/profile
_PF = $(DIR_PROFILE)/JAP_ASCS30_is04d3p
SETENV_00 = DIR_LIBRARY=$(DIR_LIBRARY)
SETENV_01 = LD_LIBRARY_PATH=$(DIR_LIBRARY):%(LD_LIBRARY_PATH)
SETENV_02 = SHLIB_PATH=$(DIR_LIBRARY):%(SHLIB_PATH)
SETENV_03 = LIBPATH=$(DIR_LIBRARY):%(LIBPATH)
SETENV_04 = PATH=$(DIR_EXECUTABLE):%(PATH)
SETENV_05 = DB2_CLI_DRIVER_INSTALL_PATH=$(DIR_EXECUTABLE)/
db2_clidriver
# Parameter Added by DELESSIO for AIX Large Page use.
# SETENV_06 = LDR_CNTRL=LARGE_PAGE_DATA=Y@TEXTPSIZE=64K@
STACKPSIZE=64K@
SHMPSIZE=64K@NAMEDSHLIB64=sap0,16M,FIRST_TOUCH

```

```

#-----
# Copy SAP Executables
#-----
Execute_00 = immediate $(DIR_CT_RUN)/sapcpe$(FT_EXE) pf=$(PF)
#-----
# Start SAP message server
#-----
_MS = ms.sap$(SAPSYSTEMNAME)_$(INSTANCE_NAME)
Execute_01 = local rm -f $_MS
Execute_02 = local ln -s -f $(DIR_EXECUTABLE)/msg_server$(FT_EXE) $_MS
Restart_Program_00 = local $_MS pf=$(PF)
#-----
# Start SAP enqueue server
#-----
_EN = en.sap$(SAPSYSTEMNAME)_$(INSTANCE_NAME)
Execute_03 = local rm -f $_EN
Execute_04 = local ln -s -f $(DIR_EXECUTABLE)/enserver$(FT_EXE) $_EN
Restart_Program_01 = local $_EN pf=$(PF)
#-----
# SAP Message Server parameters are set in the DEFAULT.PFL
#-----
ms/standalone = 1
ms/server_port_0 = PROT=HTTP,PORT=81$$
#-----
# SAP Enqueue Server
#-----
enqueue/table_size = 204800
enqueue/snapshot_pck_ids = 1600
enqueue/server/max_query_requests = 5000
enqueue/server/max_clients = 5000
enqueue/server/max_requests = 5000
enqueue/async_req_max = 5000
enqueue/server/threadcount = 3
enqueue/encni/threadcount = 4
rdisp/enqname = $(rdisp/myname)

```

11.10.4 Dialogue Profile

```

#.*
#.* Instance profile JAP_D21_SAPA3A05
#.*
rfc/use_wp_instead_of_pid = 1700
SAPSYSTEMNAME = JAP
SAPSYSTEM = 21
INSTANCE_NAME = D21
DIR_CT_RUN = $(DIR_EXE_ROOT)/run
DIR_EXECUTABLE = $(DIR_INSTANCE)/exe
SETENV_00 = DB2_CLI_DRIVER_INSTALL_PATH=$(DIR_EXECUTABLE)/
db2_clidriver
exe/saposcol = $(DIR_CT_RUN)/saposcol
exe/icmbnd = $(DIR_CT_RUN)/icmbnd
icm/server_port_0 = PROT=HTTP,PORT=80$$

```

12 Terms and definitions

Table 28. Terms and definitions

BP	Business partner
MDG	Mass data generator
BaS	Banking Services from SAP
FI-CAX	Contract accounting system for open item management
MassMan	Mass Run Analysis Tool
LPAR	Logical partition
DI	Dialog instance
Step	A step is an atomic process in the process chain, such as billing or settlement.
Cycle	A cycle is a complete run through the entire process chain: Billing, Invoicing, Post payment orders for direct debit, Payment distribution, Execute Outbound Interface, Accruals, Settlement, Calculate Capital Amounts.
Stage	A testing stage is a series of tests with a fixed number of loan accounts in the system, such as 2 or 8 million.

13 References to related material

13.1 Latest SAP on System z Banking Customer References

- Banco Galicia plans its global future, with help from SAP and IBM
[ibm.com/software/success/cssdb.nsf/CS/STRD-97XEFT?OpenDocument&Site=corp&cty=en_us](https://www.ibm.com/software/success/cssdb.nsf/CS/STRD-97XEFT?OpenDocument&Site=corp&cty=en_us)
- Nationwide transforms its core banking, service and business capabilities
[ibm.com/software/success/cssdb.nsf/cs/STRD-96UGVC?OpenDocument&Site=gicss67sap&cty=en_us](https://www.ibm.com/software/success/cssdb.nsf/cs/STRD-96UGVC?OpenDocument&Site=gicss67sap&cty=en_us)
- Video: Nationwide Building Society
<http://www.youtube.com/watch?v=-bhhsNesQCK>
- Banco Pastor slashes costs and boosts efficiency with SAP and IBM
[ibm.com/software/success/cssdb.nsf/CS/STRD-8GZF32?OpenDocument&Site=corp&cty=en_us](https://www.ibm.com/software/success/cssdb.nsf/CS/STRD-8GZF32?OpenDocument&Site=corp&cty=en_us)
- Commonwealth Bank of Australia Core Banking modernization
[ibm.com/software/success/cssdb.nsf/CS/STRD-8C6FY7?OpenDocument&Site=gicss67sap&cty=en_us](https://www.ibm.com/software/success/cssdb.nsf/CS/STRD-8C6FY7?OpenDocument&Site=gicss67sap&cty=en_us)
- Rock-solid reliability with SAP Banking applications on IBM zEnterprise
[ibm.com/software/success/cssdb.nsf/CS/STRD-92JH6X?OpenDocument&Site=corp&cty=en_us](https://www.ibm.com/software/success/cssdb.nsf/CS/STRD-92JH6X?OpenDocument&Site=corp&cty=en_us)
- Video: SAP Solutions on IBM zEnterprise, leading contributors in financial software and infrastructure for banking.
Video – Part 1:
<http://www.youtube.com/watch?v=laLeliMIbeA&feature=youtu.be>
Video – Part 2:
<http://www.youtube.com/watch?v=X2Ymwu3iKB0&feature=youtu.be>

13.2 Documents

IBM Techdocs library

- IBM zEnterprise System: SAP Loans Management for Banking Services with IBM DB2 for z/OS Performance Test Report
ibm.com/support/techdocs/atmastr.nsf/WebIndex/WP102350
- IBM zEnterprise System: SAP Bank Analyzer 8.0 AFI Loan Tests
ibm.com/support/techdocs/atmastr.nsf/WebIndex/WP102206
- IBM System zEnterprise, SAP Banking Services 7.0 150 Million Accounts Measurements
ibm.com/support/techdocs/atmastr.nsf/WebIndex/WP101978
- SAP Transaction Banking: Data Sharing Long Distance Coupling Links IBM System z
ibm.com/support/techdocs/atmastr.nsf/WebIndex/WP101692
- IBM zEnterprise System: SAP Enqueue Replication into System z Coupling Facility Performance Test Report
ibm.com/support/techdocs/atmastr.nsf/WebIndex/WP102248

SAP Community Network

- SAP Community Network: SAP on DB2 for z/OS
<http://scn.sap.com/community/db2-for-z-os>
- SAP for Banking on IBM System z Reference Architecture
<http://scn.sap.com/docs/DOC-14357>
- SAP Business Suite on IBM System z Reference Architecture
<http://scn.sap.com/docs/DOC-14431>

13.3 Additional links

- SAP infrastructure on zEnterprise
ibm.com/systems/z/solutions/sap.html
- IBM zEnterprise System scales best for SAP for Banking Solution delivering superior client value
<http://public.dhe.ibm.com/common/ssi/ecm/en/zsl03164usen/ZSL03164USEN.PDF>
- Details about the NMON tool:
ibm.com/developerworks/aix/library/au-analyze_aix/



© Copyright IBM Corporation 2013

IBM Deutschland GmbH
IBM-Allee 1
71139 Ehningen
ibm.com/de

IBM Austria
Obere Donaustraße 95
1020 Wien
ibm.com/at

IBM Switzerland
Vulkanstrasse 106
8010 Zürich
ibm.com/ch

Printed in Germany
November 2013

IBM, the IBM logo, ibm.com, IBM System x, GPFS and Cognos are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: ibm.com/legal/copytrade.shtml

Other product, company or service names may be trademarks or service marks of others. References in this publication to IBM products or services do not imply that IBM intends to make them available in all countries in which IBM operates.

SAP AG
Dietmar-Hopp-Allee 16
D-69190 Walldorf
Germany

SAP and the SAP logo are trademarks or registered trademarks of SAP AG in Germany and several other countries.

SAP and other SAP products and services mentioned herein, as well as their respective logos, are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.