SAP Enterprise Data Warehouse for Point of Sales Data Optimized for IBM DB2 for Linux, UNIX, and Windows on IBM Power Systems

Applies to:
SAP Retail, SAP NetWeaver BW 7.3

Summary
This article provides an overview of the optimizations for SAP Retail and SAP NetWeaver BW regarding efficient data management and processing of large Point of Sales (POS) data volumes on the parallel database IBM DB2 for Linux, UNIX, and Windows (in the following referred to as DB2 for LUW) on IBM POWER7 servers. Real-time processing of POS data is today’s business challenge. With a large proof of concept (PoC), we show how you can achieve these business demands using DB2 for LUW on a mid-range IBM POWER7 server. In addition, this document provides information about the valuable DB2 and POWER7 features that are used to achieve excellent performance results. We also provide best practices and recommendations on how to run a large SAP EDW system for POS data most efficiently on DB2 for LUW on POWER7.

Authors:  Brigitte Bläser (IBM)
          Karl Fleckenstein (IBM)
          Stefan Hoffmann (SAP)
          Oliver Schall (SAP)
          Olaf Rutz (IBM)

Company:  SAP AG, IBM Germany Research & Development GmbH
Created on: 6 April 2011
Author Bio

Brigitte Bläser is a senior software engineer at IBM Deutschland Research & Development GmbH in Böblingen. She is a member of the joint IBM/SAP development team that enables SAP applications for DB2 for Linux, UNIX and Windows and is responsible for integrating DB2 for LUW solutions into SAP NetWeaver BW.

Karl Fleckenstein is a Senior Technical Staff Member and the lead architect in the SAP/IBM DB2 for LUW development team. He is responsible for DB2 for LUW solutions within SAP NetWeaver applications. Karl joined IBM 1990 and worked as software engineer in several database related projects. He has more than 20 years of experience with DB2 and more than 13 years of experience with SAP basis and SAP NetWeaver BW technology. Karl is a frequent speaker at IBM's and SAP's technical conferences and has published various technical white papers with SAP/DB2 for LUW solutions. He is a member of the IBM German Technical Expert Council (TEC).

Dr. Stefan Hoffmann is a Principal Technology Consultant and Head of the In-Memory Innovation Center for Retail in St. Ingbert / Germany. He is based in Germany and his role is to advice internal and external customer in the architecture of In Memory based solutions for Trade Industry customers. Stefan joined SAP in 1998 and started as software engineer in the development team of SAP Business Warehouse. After 4 years within software development he changed role to SAP consulting and worked as a technical consultant in several projects. He has more than 10 years of experience with SAP basis and SAP BI technology.

Dr. Oliver Schall is a Technology Expert Consultant in the Global Retail Hub located in St. Ingbert / Germany. He is specialized in In-Memory technology as well as SAP basis, SAP BI technology and SAP Enterprise Portal. Before joining SAP in 2008, Oliver worked as a researcher at the Max Planck Institute for Computer Science and published several papers in international journals and gave presentations at renowned conferences.

Olaf Rutz is a senior software engineer at IBM Deutschland Research & Development GmbH in Böblingen. He is the technical team lead of the SAP on AIX team and has more than 15 years experience working with SAP on IBM System z, i and p platforms.
Table of Contents

Introduction ........................................................................................................................................... 5

Enterprise Data Warehouse Solutions for POS Data ........................................................................... 6

General .................................................................................................................................................. 6

POS Inbound Processing Engine (PIPE) ................................................................................................. 6

POS Analytics ......................................................................................................................................... 8

Layered Scalable Architecture .............................................................................................................. 8

Terminology ............................................................................................................................................ 9

Service Offering ................................................................................................................................... 11

Innovative Features Used ..................................................................................................................... 12

Features of DB2 for LUW ..................................................................................................................... 12

Database Partitioning Feature (DPF) ...................................................................................................... 12

Multidimensional Clustering (MDC) ...................................................................................................... 13

Data and Index Compression ............................................................................................................... 15

Compression of Temporary Tables ..................................................................................................... 16

Automatic Statistics Collection and Real-Time Statistics .................................................................... 16

Automatic Storage Management ......................................................................................................... 16

DB2 Specific Near-Line Storage (NLS) Solution .................................................................................... 17

Features in SAP NetWeaver BW 7.3 ..................................................................................................... 18

MPP-Optimized DSO Data Activation .................................................................................................... 18

Parallel Data Extraction from Write-Optimized and Standard DSOs .................................................... 18

IBM Power Systems Features .............................................................................................................. 18

POWER7 ............................................................................................................................................... 18

Active Memory Expansion (AME) ........................................................................................................ 19

Test Environment ................................................................................................................................ 21

Hardware Setup .................................................................................................................................... 21

SAP System Setup ................................................................................................................................ 22

Test Scenarios ....................................................................................................................................... 24

Results 26

ETL Performance Evaluation ................................................................................................................. 26

NLS Performance Evaluation .............................................................................................................. 31

NLS: Data Archiving Tests .................................................................................................................. 32

NLS Query Performance Tests .......................................................................................................... 33

POWER7 AME Tests ............................................................................................................................ 35

POWER7 AME Activation for the SAP Application Server LPAR ..................................................... 35

Best Practices ....................................................................................................................................... 37

SAP Configuration for AIX .................................................................................................................... 37

Active Memory Expansion Planning and Advisory Tool ................................................................. 37

DB2 for LUW ......................................................................................................................................... 38

Summary ............................................................................................................................................... 40

Appendix ............................................................................................................................................... 41

List of Abbreviations ........................................................................................................................... 41

DB2 Configuration Details .................................................................................................................. 41

DB2 Registry ......................................................................................................................................... 41
Database Manager Configuration ................................................................. 42
Database Configuration ................................................................................. 44
Bufferpools (KB) .......................................................................................... 48
Table of Figures .............................................................................................. 49
List of Tables .................................................................................................. 49
Related Content ............................................................................................. 50
Copyright .......................................................................................................... 51
Introduction

Efficient data management and processing of large POS data volumes has become more and more important for the business success of large retail companies.

Current business challenges require a fast data consolidation and data cleansing of POS data and providing the POS data up to the reporting layer in a timely manner to allow querying of operational data in real-time.

This document describes why the combination of IBM Power Systems™, AIX®, DB2® for LUW, SAP NetWeaver Business Warehouse (SAP NetWeaver BW), and the SAP Retail application provides an excellent response to these demands. It is proven that a large POS data volume of eight billion records can very efficiently be processed on an SAP NetWeaver BW system on DB2 for LUW with a mid-range POWER7 server.

In addition, information about SAP’s Enterprise Data Warehouse solution for POS data is also provided in this document.

Especially the optimizations for SAP Retail and SAP NetWeaver BW that use the parallel database feature of DB2 for LUW are the key for fast data processing of POS data. In addition, the new POWER7 feature Advanced Memory Expansion significantly reduces physical memory requirements of the SAP application servers and the database server. The chapter “Innovative Features Used” provides an overview of these optimizations and the DB2 features, such as, database partitioning, multi-dimensional clustering, automatic storage management, compression and the DB2-specific near-line storage (NLS) solution.

In chapter “Test Environment”, the hardware and system setup, and the test scenarios used in this PoC are described. Chapter “Results” provides a detailed description of the test results, and in chapter “Best Practices” you find best practices for the optimal setup of an SAP Retail POS data management system on DB2 for LUW, IBM POWER7, and AIX.
Enterprise Data Warehouse Solutions for POS Data

General

For many organizations, POS data is scattered across the enterprise in multiple systems and applications. The result is data fragmentation and information redundancy, which makes a thorough analysis of your business very hard to do. Today, these challenges prevent organizations from achieving goals, such as:

- Producing holistic reports in a timely manner
- Querying operational data in real-time
- Ensuring that reports contain the most accurate and up-to-date information

More and more customers are looking for a central platform that acts as entry for POS data into their organization.

For many years, SAP has been offering the POS Data Management solution (SAP POS DM) to Trade Industry customers. SAP POS DM is a packaged solution that can be installed as a standalone application. It acts as a data hub for POS data in an organization or you can install it as part of SAP Enterprise Data Warehouse.

POS DM provides a single point of entry for sales data from various distribution channels, including web sales, via standard interfaces.

POS DM is able to load intraday sales data and provides services for loading intraday sales data deltas to SAP ERP Retail since SAP ECC6.0 EHP2. SAP ECC supports near real-time stock and inventory counts during opening hours.

The sales data in POS DM can also be transferred to non-SAP systems. Thus customers can use POS DM as a single central system for the entry and validation of sales data and for the distribution of the data to all targets that require the sales data. POS DM offers the following functions and features:

- Data cleansing
  Very often data quality is an issue. Since POS DM is cleansing the POS data using already cleansed master data from SAP NetWeaver BW, it is ensured that all other systems which receive that data use the same data basis. Since this also includes POS Analytics, it is ensured that real-time data is cleansed in the same way as historical data. This procedure leads to accurate data and is SAP’s answer to ‘one version of the truth’.

- Real-time reporting
  In today’s business, the overall success is determined by the ability to query in real-time what is happening in the store. Only few applications offer such possibilities. SAP POS DM allows fast loading of cleansed data in the data warehouse environment, which ensures reports to have the most accurate and up-to-date information. Standard interfaces allow direct access to data that is not (yet) loaded into the reporting layer which shortens the latency time between data in the data warehouse and data in the operational layer.

- Trickle feed
  Customers need to query operational data in real-time. This requires an architecture that provides fast data provisioning techniques and allows trickle feed scenarios. Since the POS Inbound Processing Engine (PIPE) part within POS DM is able to forward data immediately to requesting systems, a trickle feed into SAP NetWeaver BW is easy to establish. Assuming that a customer requires near-real time POS data in the data warehouse, the simplest approach is to continuously feed the BW system with new data from the POS system and insert the new data into separate InfoCube fact tables. With the modeling and scheduling tools within SAP NetWeaver BW, a customer can establish such a process without any additional change in their landscape.

- Stable response times
Increasing numbers of information consumers need to query on transactional level. Since the data volume they have to access can be tremendously high, the architecture has to provide mechanisms that allow access on such a granularity with stable response times.

- **Ad-Hoc query**

Due to the high granularity of data, it is often impossible to offer ad-hoc query capabilities to a larger number of users because those unpredictable queries cannot be optimized in advance. The runtime problems that are related to this type of queries often lead to the fact that customer do not allow end users to access this layer of granularity. With POS DM in combination with Business Warehouse Accelerator (BWA), a larger number of users can access this layer of granularity with a stable response time and an easy-to-use user interface.

Due to the high granularity of POS data, it is often required to use aggregation levels in reporting to be able to query the data. To build these aggregation levels is time consuming and prolongs the time to process POS data. Within POS DM, there are several options to shorten these processing times. With BWA in place, there is no need any longer for those aggregation levels, which leads to very short processing times.

- **Product affinity (basket analysis)**

To get a better insight into the shopping behavior of customers, organizations need to be able to analyze the POS data on a transactional level. POS DM provides with product affinity metrics a solution that is embedded into the BWA. Product affinity metrics enable organizations to find product affinities that help them to run more efficient marketing campaigns and to optimize their assortment.

Key benefits of the application for customers of the trade industry are as follows:

- Complete POS data management solution in a stand-alone mode
- Rapid implementation through best practices resulting in quicker return of investment (ROI)
- Enhanced performance through usage of in-memory technology
- Extendable architecture for future product development
- Seamless links to existing SAP applications
- Integration with non-SAP applications via SAP Business Objects Data Services or SAP Process Integration
- Covers requirements beyond traditional retailing
- Improved handling and controlling of POS inbound processing and analysis
- Support for trickle polling and intraday flash sales
- Complete sales audit application included

The complete solution can be divided into two major applications:

**POS Inbound Processing Engine (PIPE)**

This application is used to process POS transactions in parallel into a centralized data hub. Within this architecture, the central container TLOG stores transactions from various POS systems and distributes them to systems in the organization that require POS data. SAP delivers standard interfaces and algorithms that you can use to check, cleans, aggregate, and distribute the data and to monitor the flow of the data to the systems in your landscape.

The PIPE provides a complete suite of functions for checking the correctness and for cleansing POS data (Sales Audit). Besides mass handling of POS transactions and reverse postings with monitoring capabilities, various analytical applications can be supplied on top of the central storage TLOG.

The integrated sales-audit functionality of the PIPE ensures that all data is accurate and complete.

Since the PIPE has been built to be integrated with SAP NetWeaver BW, retail data is delivered directly to the heart of your company – there is no need to process the retail data in the merchandise management
system first. This improves the performance of all your IT systems and allows documents and relevant data to be used throughout your organization.

Seamless integration into existing SAP applications enables a fast ROI and offers a low TCO.

**POS Analytics**

SAP POS DM software improves decision making at all levels by gathering facts and figures from your retail outlets and making them available throughout your enterprise. It also provides standardized reports at the click of a mouse through a Web-based user interface, enabling Information consumers to access, explore, and analyze incoming sales information faster and more easily than ever before. In addition, POS analytics offers vital support for long-term strategic and operational decision making, which helps you capitalize on emerging business opportunities. The application is integrated directly with SAP Business Warehouse (SAP NetWeaver BW) software, allowing you to scrutinize the performance of your store from different perspectives. It also links to a range of follow-on processes, such as inventory management, financials, billing, and credit-card settlement.

In addition, the application supports trickle polling, giving your planners an up-to-the-minute picture of everyday’s activities. This approach reduces strain on other systems, such as merchandise management, that often work through the night processing mission-critical data.

A predefined Enterprise Data Warehouse (EDW) model accelerates implementation and maximizes return on investment (ROI). The application comes ready-to-run, with POS processing and analytics features that are tailored to a variety of standard roles. These standard roles range from store managers to replenishment planners.

The implemented EDW model follows the methodology of layered scalable architectures and provides a complete, robust architecture that covers the needs of all size retailers in regards to POS reporting.

**Layered Scalable Architecture**

The layered scalable architecture of the underlying EDW model (see Figure 1) can be divided into the following main areas:

![Layered Scalable Architecture (LSA) for Point of Sales Data Management](image)
Application Area:
- Persistence layer for business driven transformations
- In the application area, business departments find the data targets that reflect their given business requirements for an analytical application. Sometimes, these applications are also called architected data marts.

EDW Area:
- No business driven transformations; only integration transformations & cleansing based on common definitions and delivering reusable parts (Single Point of Truth)
- Each area is divided in sub areas (layers) that support specific functions.
- Parallel processing in the database (as available with DB2) is an important feature to achieve the real time POS DM requirements and good performance for the Extract/Transform/Load (ETL) processing within the layered scalable architecture.

Terminology
The following terminology is used in this document to describe the components of the LSA.

Data Domains
Refers to the logical or semantic grouping of data into separate objects that have been implemented to enhance loading stability (activation parallelization, object contention, and so on) and query performance, and so on.

Domains are always based on the source system and are grouped according to region or time-zone.

Layers
Refers to the sub-division of the LSA areas according to their functions, rules, and restrictions. The LSA is divided into the following layers:

- **Data Acquisition Layer**
  The data acquisition layer has the task to quickly accept the extracted records from the source and to distribute them further. It decouples the loads from the source systems from internal BW processing (spread workload). Therefore, you can extract data for a DataSource multiple times per day instead of fill specific BW applications only once.
  The acquisition layer is the earliest point in time where you can create multiple outbound data flows from a single inbound data flow and thus parallelize the BW processing.

- **Harmonization & Quality Layer**
  The harmonization & quality layer ensures the availability of the complete set of data in a cleansed and incorporated way.

- **Data Propagation Layer**
  The propagation layer is the EDW layer that is meant to provide digestible data for the application area. That is, data that is ready for consumption by BW applications.
  The propagation layer is mandatory for all transactional data and does not reflect the domain concept of a layered scalable architecture. Instead of splitting this area into several sub-domains, all data coming from the various sources is routed through this object.
  To achieve the best possible performance, a write-optimized DataStore Object (DSO) is used. Generally, the propagation layer provides all commonly-calculated figures that are relevant for all higher level cubes and DSOs.
The main benefit of the application layer is that it provides a persistent platform of data with the necessary calculations so that loading of applications is not delayed.

- **Corporate Memory**

  The data in the acquisition layer is not retained. To enable reloading of data independently of the source system, an additional layer is needed, which is the corporate memory. The corporate memory stores all fields delivered from the DataSource. Thus, the corporate memory is comprehensive. Information is not aggregated. It is a write-optimized DSO and should be designed without a semantic key. All extracted records are stored. The corporate memory is historically complete.

  It is not expected that any transformations - apart from 1:1 mappings, data splits or concatenations - takes place that go into the corporate memory layer. Therefore, considerations about performance and time dependency of master data that is used in transformations are not important.

  To simplify the selection of data that is required for reloading and recovery from corporate memory DSOs, the load date and request number must be loaded in delta update mode into all write-optimized corporate memory DSOs. Data is not deleted from corporate memory.

  Corporate memory is used appropriately when a transformation in the data model is modified and data must be reloaded to reflect these changes, and no new fields are added from PIPE or the underlying POS system.

  In case a reload of data into the business transformation layer (see next section) is required, data is read from the corporate memory objects and stored in the data propagation layer. From there, data is distributed into the various business transformation layer objects.

- **Business Transformation Layer**

  The business transformation layer is the first layer of the application area. It is a mandatory layer and it is used for complex application-specific business logic, such as the following:

  - Merging and joining (lookups) of data from different DataSources
  - Complex operations and lookups that fill additional characteristics and that are not delivered in the propagation layer increase the need for a dedicated persistency.

  Normally, this layer consists of standard DSOs. However, there are exceptions where a write-optimized DSO can be used to optimize load performance or support persistency. For example, this is the case when you merge several high data volume standard DSOs from the propagation layer via full loads.

  The rational for placing a DSO in the transformation layer or the propagation layer is the application specificity of the transformations and the reusability of the result. For example, if one DataSource can be used by many applications but only one application requires certain transformations to take place, then these transformations should be performed in the transformation layer. To be more precise: whenever the material information in the sales data must be enriched with assortment-specific information and the assortment enrichment is relevant for only one other area, it makes sense to enrich the data at the transformation layer.

- **Reporting Layer**

  The reporting layer comprises multidimensional InfoProviders that store data at the various aggregation levels depending on the requirements of the application. In exceptional cases, DSOs can also be used if, for example, only pure list reporting is required.

  To achieve a highly flexible data model in terms of adding or removing InfoProviders to queries (which is also called logical partitioning), it is not allowed to do reporting directly on a basic InfoProvider but rather access InfoProviders using MultiProviders.
The reporting layer can be divided into different sub-categories, for example, high performance cubes (that are more aggregated) versus more detailed or granular cubes. Query performance is the key criteria to judge whether your InfoCube is correctly aggregated.

You should not attempt to provide all kinds of information for all users in a single cube. Instead, we recommend that you create multiple cubes that contain the same set of data but with different levels of detail.

**Service Offering**

SAP offers the Enterprise Data Warehouse Service for POS (EDW Service for POS) as a complete productized service (SAP SIL Service) to jump start a retailer’s POS data warehouse. Based on the latest SAP NetWeaver BW & EIM architecture, the EDW Service for POS delivers preconfigured data-models and an integration architecture to implement and operate a high-volume POS data warehouse.

The EDW Service for POS provides a POS Enterprise Data Warehouse data model in form of a transport package that you can install using the standard transport mechanism of SAP. Implementation of the packaged solution into the customer’s BW system is jointly performed by SAP Consulting and the customer. During workshops, SAP consultants explain LSA principles like semantic partitioning and customize the customer system accordingly. An operational guide explains how to manage and operate the LSA EDW model.

Furthermore, knowledge transfer is performed by the consultants during introductory workshops regarding SAP NetWeaver BW Accelerator and SAP Business Objects Explorer.

The EDW Service for POS serves as reference architecture for SAP NetWeaver BW & EIM (classic SAP customer) and provides a best practice POS data model that covers the most common retail use cases for POS analytics as well as lean integration with SAP Business Objects front end tools.

SAP consultants transfer knowledge with regard to a high performance analytics setup using SAP Business Warehouse Accelerator and SAP Explorer Accelerated. In addition to the introduction of the LSA data model, the EDW Service for POS comes with comprehensive documentation material. SAP Consultants are enabling the customer to run the data warehouse (DWH) by providing initial customer workshops.

The EDW Service for POS is mainly based on the SAP POS Data Management solution that runs on the SAP NetWeaver BW platform. The data model can be integrated best if the customer already has the SAP POS Data Management solution, which is the strategic and state-of-the-art POS data collection and processing hub in combination with an EDW model of a Layered Scalable Architecture.

The service lowers the total costs of implementation dramatically, as the EDW data model is shipped as business content in its own namespace. Customers who are familiar with content activation could implement the solution themselves. In combination with the documentation, customers are able to implement the EDW for POS within a few days instead of a whole project cycle (which includes blueprinting, testing, and so on)

Based on the EDW approach, the EDW Service for POS enables reporting down to receipt level and thus provides deep insight into shopping behavior.

By offering the EDW Service for POS for a fixed price, the service provides full control of implementation costs and offers the possibility to set a comprehensive data foundation for POS Analytics.
Innovative Features Used

This chapter describes the features of IBM DB2 for LUW, IBM Power Systems, and SAP NetWeaver BW 7.3 that are exploited to achieve a significant reduction of TCO, excellent performance and scalability, a reduction of maintenance efforts, and a full data life-time support of this optimized POS DM on DB2 for LUW and IBM Power Systems solution.

Features of DB2 for LUW

Database Partitioning Feature (DPF)

The database partitioning feature (in the following referred to as DPF) allows the segmentation of a DB2 database into several database partitions that can be located on a single large Symmetric Multiprocessing (SMP) database server (scale-up) or on several smaller discrete servers (scale-out).

DPF follows the shared-nothing architecture. That is, each database partition is a part of the database and consists of its own data, indexes, configuration files, and transaction logs. Tables can be distributed automatically over several database partitions using a hash key that is defined when the table is created.

Since data is distributed across database partitions, the throughput and bandwidth of many processors is aggregated and thus reduces the runtime of SQL access and database maintenance operations. Data retrieval and data manipulation requests are automatically decomposed into sub-requests, and executed in parallel on the database partitions where data resides. The fact that databases are split across database partitions is transparent to users issuing SQL statements.

DPF has the following advantages:

- Scalability options
  - You can attach additional servers to provide support for growing data and user volumes
  - Increased concurrent access to data for concurrently running applications or processes
- Excellent data access performance
  - Faster SQL access because the data can be processed in parallel on the database partitions on which it resides
- Faster database maintenance operations:
  - Operations such as table and index reorganization or index creation are faster because they are executed in parallel on the database partitions where the table or index is located. Collection of table or index statistics is faster because it is only executed on one database partition.
  - Faster database backup and faster recovery because these functions run in parallel on all database partitions
- TCO reduction:
  - Efficient usage of the hardware resources, that is, you start, for example, with a small database server and add another server later when the system is getting larger.

DPF has been supported in SAP NetWeaver BW as of SAP Business Information Warehouse 1.2. The DB2 database of an SAP NetWeaver BW system can be created as a multi-partition database consisting of an administration partition and several data partitions.

The administration partition performs the following tasks:

- Storing the DB2 system catalog
- Storing SAP basis tables and the SAP NetWeaver BW management, master data, and dimension tables
• Storing small Persistent Staging Area (PSA), DataStore Object (DSO), and InfoCube fact tables. As soon as these tables become large, we recommend that you distribute them over the data partitions.

• Serving as the coordinator partition for the SAP application. This means that the SAP application servers connect to the administration partition.

The data partitions store the following large SAP NetWeaver BW tables:

• InfoCube and aggregate fact tables
• DataStore Object tables
• PSA tables

Figure 2: Standard SAP NetWeaver BW Database Layout on DB2 for LUW

The creation and management of database partitions has been fully integrated into SAP tools such as the SAP installation tool SAPinst, the SAP DBA Cockpit, and the SAP Database Performance Warehouse.

In this PoC, the DB2 database was created with the administration partition and seven data partitions. The tables of the DSOs and the fact tables of the InfoCubes of the layered scalable architecture for POS DM were distributed over the seven data partitions. All partitions were located in one AIX LPAR.

Multidimensional Clustering (MDC)

Multidimensional clustering (MDC) allows the clustering of the data in a table along one or more columns. The columns selected for clustering are called MDC dimensions. Records with the same values in the MDC dimension columns are stored physically together in blocks of pages. This means that MDC physically organizes data in a table into blocks according to the values in the clustering columns.

The following Figure 3 shows an MDC table with two dimensions (Region and Year):
Figure 3: Multidimensional Clustering

In an MDC table, each unique combination of MDC dimension values forms a logical cell, which can physically consist of one or more blocks of pages. Each data page belongs to exactly one block, and all blocks consist of the same number of pages.

The size of a block is equal to the tablespace extent size so that block boundaries border on extent boundaries, which is beneficial for I/O operations. For DB2 databases in an SAP environment, the extent size is 2 by default, and the page size is 16 KB. Therefore, an MDC block is 32 KB in size.

The logical cell only has enough blocks associated with it to store the records having the dimension values of that logical cell. In case there are no records in the table that have the dimension values of a particular logical cell, no blocks are allocated for that logical cell.

MDC introduces block-based indexes. These indexes are much smaller than regular record-based indexes because the block indexes point to MDC blocks instead of individual records. Therefore, block indexes consume a lot less disk space and can be scanned faster. Structurally, they are the same as regular indexes.

The following figure shows an example of block indexes for the table in Figure 3. In addition to the two dimension block indexes on Region and Year, a combined block index on Region and Year is also generated.

Figure 4: Multidimensional Clustering – Block Indexes

MDC provides the following benefits:

- Less data reorganization:
With MDC, a table can be physically clustered on more than one key or dimension simultaneously. An MDC table maintains and guarantees its clustering across all dimensions automatically and continuously. It is not necessary to reorganize MDC tables to restore the physical order of the data.

In addition, starting with DB2 V9.7 for LUW, empty blocks can be returned to the tablespace very efficiently so that the space can be used by other tables. The reclaim of empty extents of MDC tables is an online operation that can be automated with DB2’s automatic table reorganization feature.

- **Improved SQL query performance:**
  The performance of queries with predicates on the clustered dimensions of an MDC table is improved because these queries access only those pages that have records with the qualifying dimension values. The qualifying pages are grouped into blocks, which results in most efficient sequential prefetching.

- **Reduced database locking:**
  MDC supports the loading of data from external source systems (roll-in). If different processes insert data into separate cells, locking is reduced to lock MDC blocks instead of single rows. Note that you can enable block locking for an individual MDC table by setting the LOCKSIZE parameter to BLOCKINSERT in the database system catalog. In SAP NetWeaver BW systems, block locking is enabled automatically where applicable.

- **Fast data deletion:**
  MDC supports fast deletion of large amounts of old data that is no longer needed for reporting (roll-out). If all data of an MDC block is deleted, the pages of the block are marked as deleted and the block indexes are updated. Starting with DB2 V9.5, additional record-based indexes defined on MDC tables are cleaned up asynchronously in the background when the deletion of the data is completed. This improves the data deletion process by factors. Empty blocks can be reused as soon as the rollout is committed.

MDC is fully supported in SAP NetWeaver BW 7.0 and higher releases. A back-port with limited functionality is available for SAP NetWeaver BW 3.5, SAP Business Information Warehouse 3.0B, and SAP Business Information Warehouse 3.1 Content. For more information, see SAP Note 942909.

For more information about MDC and its use in SAP NetWeaver BW, see the following guide: SAP NetWeaver Business Warehouse 7.3 – Administration Tasks: IBM DB2 for Linux, UNIX, and Windows, [https://websmp101.sap-ag.de/~sapidb/011000358700001420572010E.PDF](https://websmp101.sap-ag.de/~sapidb/011000358700001420572010E.PDF).

With the focus of this PoC in the data staging capabilities, MDC was beneficial for fast deletion of data in the following situations:

- Deletion of processed requests from PSA tables after the data has been written to corporate memory
- Deletion of requests that are older than 10 days from the standard DSO change log tables
- Deletion of data that has been archived to near-line storage (NLS) from InfoCubes and DSOs

MDC is also used in the MPP-optimized DSO data activation that is described later in this document in the Features of SAP NetWeaver 7.3 section.

**Data and Index Compression**

With data and index compression, the disk space needed for tables and indexes can be reduced considerably. Data compression was introduced in DB2 V9.1 for LUW, and index compression was introduced in DB2 V9.7 for LUW. DB2 data compression is based on compression dictionaries that are created automatically when data is written into a table, which is enabled for compression. Compression dictionaries can also be created manually as follows:

- By reorganizing a table with the RESETDICTIONARY option. The REORG command (re)builds the compression dictionary and (re)compresses all the data.
- By executing DB2 INSPECT command with the ROWCOMPESTIMATE option. The INSPECT command creates a compression dictionary and inserts it into the table if the table is enabled for
compression. Existing data is not compressed but data inserted or updated in the future is compressed with the dictionary.

In DB2 V9.7 for LUW, indexes are created and are then automatically compressed if the table is enabled for compression.

In SAP NetWeaver BW, PSA tables, DSO tables, and InfoCube and aggregate fact tables can be created with compression enabled. If at least DB2 V9.5 for LUW is used and if the IBM DB2 Storage Optimization Feature is licensed, these tables are created with compression automatically enabled. Dimension and master data tables are not automatically compressed. The IBM DB2 Storage Optimization Feature is included in the SAP IBM DB2 OEM license at no extra cost.

In this PoC, data and index compression is used. The size of the tables of the layered scalable architecture for POS DM could be reduced between 82% and 88%, the size of their indexes between 53% and 71%.

Compression of Temporary Tables

Compressing temporary tables is especially beneficial for SAP NetWeaver BW systems because large temporary tables might be required to handle complex BW queries. The compression of temporary tables offers reduced IO, better use of the buffer pool, and the possibility to decrease the size of the temporary tablespace. Compression of temporary tables is based on a dictionary and works in the same way as data compression of regular tables.

Automatic Statistics Collection and Real-Time Statistics

Automatic statistics collection lets the DB2 engine check and determine whether database statistics need to be updated. The database triggers the RUNSTATS utility in the background for all tables that need updated statistics. A database administrator does not need to create additional jobs manually to check and update database statistics.

The DB2 engine distinguishes between asynchronous and synchronous automatic statistics collection:

The evaluation engine for asynchronous statistics collection runs every two hours and checks table and index statistics. Tables for which no statistics exist or which have outdated statistics are scheduled for RUNSTATS. Tables scheduled for RUNSTATS are processed in the online maintenance window. Asynchronous statistics collection collects full table statistics with distribution and sampled detailed index statistics by default.

If the SQL compiler requires new statistics to create an optimal access plan, the SQL compiler invokes synchronous statistics collection. Depending on the statistics required by the SQL compiler and the size of the table, statistics might be fabricated, collected on a small sample of the data only, or collected fully. Synchronous statistics collection is called real-time statistics or automatic statement statistics and it was introduced in DB2 V9.5 for LUW.

SAP NetWeaver BW systems rely on DB2 automatic statistics collection. If automatic statistics collection and real-time statistics are enabled, calls to the RUNSTATS command in the SAP NetWeaver BW ABAP code are not executed.

During this PoC, automatic statistics collection and real-time statistics were activated.

Automatic Storage Management

If DB2 databases are created with automatic storage, you have to only define a storage pool that is associated to the database. This storage pool is defined by the storage path(s) for tablespaces and the database storage path. The storage path(s) for tablespaces store the data of the tablespaces that are enabled for automatic storage. With automatic storage enabled, tablespace sizes are administered automatically. Database administrators do not have to define container sizes or to add new containers to tablespaces manually any more. Monitoring of free space for the database is reduced to monitoring free space in the file system, which is a rather simple task that you can automate quite easily.

In DB2 V9.7 for LUW, automatic storage and DMS tablespaces are automatically enabled for the reclaimable storage feature. With reclaimable storage, the tablespace extents can be consolidated below the high-water mark and unused extents can be returned to the system for reuse.
The DB2 database used in this PoC was created with automatic storage on four storage paths and tablespaces enabled for reclaimable storage.

**DB2 Specific Near-Line Storage (NLS) Solution**

There is an integrated DB2-specific NLS solution for SAP NetWeaver BW available.

NLS is a new category of data persistency that is similar to archiving. Using an NLS database enables you to transfer historical read-only data of your InfoProviders (for example, InfoCubes and DataStore objects) to an additional NLS database. The database server and the storage devices of the NLS solution can be located on the same server as the SAP NetWeaver BW system or on a dedicated server. However, you can still access the separated data in the NLS database transparently from an SAP NetWeaver BW system.

The following figure (Figure 5) shows a basic setup of the NLS solution with DB2 for LUW as NLS database:

![Figure 5: Basic Setup of NLS with DB2 for LUW as NLS Database](https://websmp101.sap-ag.de/~sapidb/011000358700001299002010E.PDF)

You can access NLS data online via the OLAP processor and the additional NLS interface that consists of the following parts:

- General NLS interface, which is database independent
- DB2-specific interface that establishes a database connection to the NLS database server and accesses the NLS data

The DB2-specific NLS solution is available as of the following SAP NetWeaver SAP NetWeaver releases and DB2 releases:

- DB2 V9.7 for LUW (for the installation of the NLS database)
- Enhancement Package 1 of SAP NetWeaver 7.0 Support Package 6 or higher with at least DB2 V9.1 or higher

This solution is delivered with the SAP NetWeaver 7.0 product and you can install it using the SAP installation tool SAPinst. To monitor the NLS database, you can use the NLS-specific extensions in the DBA Cockpit.

For more information about the DB2-specific SAP NetWeaver BW NLS solution, see the following guide: How-To Guide: Enabling SAP NetWeaver Business Warehouse System to Use IBM DB2 for Linux, UNIX, and Windows as Near-Line Storage (NLS), [https://websmp101.sap-ag.de/~sapidb/011000358700001299002010E.PDF](https://websmp101.sap-ag.de/~sapidb/011000358700001299002010E.PDF).
Excellent performance numbers (see chapter "Details of the SAP NetWeaver BW Near-Line Storage (SAP NW BW NLS) tests") are measured in this PoC for:

- Data transfer to the NLS database
- BW query response time

You can use the DB2-specific NLS feature to reduce the size of the BW database and the TCO of the overall SAP EDW POS solution.

**Features in SAP NetWeaver BW 7.3**

**MPP-Optimized DSO Data Activation**

MPP-optimized DSO data activation is a new feature of SAP NetWeaver BW 7.3. It explores MPP architectures such as DB2 for LUW DPF to improve the performance of the activation of new data in standard DSOs. In general, the record-based processing of DSO activation is replaced by a few complex SQL statements that process all records of a DSO request in parallel.

As a result, the performance of DSO data activation can improve by factors compared to SAP NetWeaver BW 7.0 and Enhancement Package 1 of SAP NetWeaver BW 7.0.

For standard DSOs to qualify for MPP data activation, ‘SID generation during Activation’ and the ‘Unique Data records’ flag have to be deactivated in SAP NetWeaver BW. With DB2 for LUW as the underlying database, you have to distribute the DSO over at least four database partitions for new data to be activated with the MPP-optimized data activation.

In this PoC, the MPP-optimized DSO activation is applied for the DSOs in the layer business transformation of the layered scalable architecture for POS DM.

**Parallel Data Extraction from Write-Optimized and Standard DSOs**

In SAP NetWeaver BW 7.3, you can configure data transfer processes (DTPs) from write-optimized DSOs and DSO change-logs for parallel extraction of data (conditionally available starting SP3). In earlier SAP NetWeaver BW releases, serial extraction from DSOs was performed together with several write processes that transfer the data to the target InfoProvider. In this PoC, a parallelism setting of 25 batch processes was used.

**IBM Power Systems Features**

**POWER7**

The new POWER7 processor has been available since early 2010 and is built in 45 nm technology. Compared to the previous POWER6 CPU, the number of cores on a chip increased by a factor of 4 to 8 cores. In addition, the number of threads available within a core doubled to 4 and the L3 cache moved to the chip itself. With those characteristics, the POWER7 processor is an ideal candidate to implement the SAP Enterprise Data Warehouse for Point of Sales (POS) Data Solution as described in this document. As you have seen in the previous sections, one important aspect of this solution is the very high degree of parallelization on all levels. The increased number of cores and threads supports this improved solution design in a perfect manner.

Additionally, the ability of AIX to intelligently schedule processes with a larger number of threads in the cores helps to adapt to the characteristics of the workload. When only few processes are runnable at a given time, these are evenly spread to the primary threads of the available cores by the AIX scheduler. Resultingly, all
resources of a dedicated core are available to each of the processes. As soon as the number of run-able processes exceeds the number of available physical cores, the additional processes are scheduled to the secondary or tertiary threads of a core.

For more information about POWER7 technologies, see the following IBM Web site:

http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/FQ128785

For more information about the improved SMT modes, see the following IBM Web site.


Active Memory Expansion (AME)

Another feature of the POWER7 processor is active memory expansion (AME). AME uses intelligent compression technologies to increase the use of physical memory. In the example described in this document, we used AME to run the given workload with a smaller physical memory footprint while minimizing performance impact. You will find some measurement results later in this document.

In traditional operating systems, physical memory is used through the virtual memory manager (VMM). This is also true for AIX unless advanced technologies such as AME or active memory sharing (AMS, which is not described in this document) are used. The VMM provides each process in the system with a complete address space and this is called the virtual memory. Each process has its own virtual memory and this address space is partitioned evenly into pages. A page has typically a size of 4 KB. The sum of all virtual memory from all processes can exceed the amount of physical memory in the system. Only if a process is accessing a page in its virtual address range, the operating system provides a physical memory page to this process. The mechanism to detect that a virtual address is not backed by physical memory and to provide it to the process is called page fault. It might happen that the memory demand in the system across all processes is exceeding the available amount of physical memory. In this case, a page fault operation does not find a free physical memory page to provide to the process. The operating system takes a rarely-used page from any process and writes its content to a specific area on disk, which is called paging device or paging file. Now this available physical memory page can be used for the process that demanded a new memory page. If there are free physical memory pages, a typical page fault operation can be handled in micro or even nanoseconds. If the page fault operation has to include disk operation, the time to handle this request increases to milliseconds. This is the reason why systems doing intensive paging are seeing dramatic performance degradations.

By using AME, the VMM has now better options to handle page fault operations. The underlying physical memory is divided by the operating system into two areas: the compressed and uncompressed pool. Pages in the uncompressed pool can be used by the VMM in exactly the same way as in the traditional case. When overall memory demand within an OS instance is exceeding the size of the uncompressed pool, the algorithm is changed. Instead of swapping a memory page to a HDD device, the VMM can compress the content of this page and write it to the compressed pool. This compression operation and in-memory write is much faster than an I/O operation to disk. Typically, this only takes a few microseconds - compared to the milliseconds required for the disk operation. After the page has been written to the compressed pool, the VMM can use the emptied uncompressed page for the demanding process. The sizes of the compressed and uncompressed pool are managed automatically by the operating system. They are controlled by an AME parameter called compression factor. This compression factor specifies up to which degree the operating system is allowed to compress pages. The higher the compression factor is, the more compression and decompression operations may occur. If the overall memory demand in the system is increasing beyond the amount of memory that can be held in the compressed and uncompressed pool, the system will use traditional paging as well. Therefore, you can consider AME as a two-stage paging.

From the perspective of a process, the size of the physical memory is expanded by the compression factor, why it is called expanded memory.
Figure 6: Active Memory Expansion

AME has multiple advantages which depend on the access pattern to the memory and the amount of physical memory needed by an application. Applications tending to access just a few pages frequently - while majority of instance context is contained in rarely used pages - can benefit most from AME. Typically, an SAP ABAP instance is better suited for the use of AME than a JAVA instance, because garbage collection in Java accesses a large number of pages in the Java heap and thus might cause more compression and decompression activity.

The following are examples of well behaving use cases for AME:

- Use AME to increase the amount of expanded memory for a system without increasing the size of the physical memory
  
  If a system has a peak memory requirement beyond its physical memory size every once in a while AME can help a lot. Instead of moving data to a paging device, the system starts to compress pages to memory, which is much faster as paging. Therefore, instead of experiencing drastic performance degradations during high load phases, these phases can be handled with just a minimal impact on the system performance.
  
  In addition, if the memory demand of an application increased, for example due to a higher number of active users, but the system cannot be upgraded by additional physical memory (all DIMM slots are used), using AME could be very beneficial.

- Run an application with lower physical memory footprint
  
  If there is enough memory available in the system to cover all high load phases and the memory access pattern shows a lot of cold (rarely used) pages, you can increase the AME compression factor. This allows you to reduce the physical memory in the system so that you use system resources more efficiently. The other way round: AME supports consolidation of incremental OS instances onto a Power System while keeping the same physical memory size.

With regards to the solution described in this document, we ran the application with decreasing amounts of physical memory while keeping the expanded memory size always on the same level.

For more information about AME, see the following references:

- Active Memory Expansion: Overview and Usage Guide,

- Active Memory Expansion Performance
Test Environment

Hardware Setup

On the hardware side, we used a POWER7 based Power 750 running with 3.3GHz for the database and the application server. This machine was partitioned into several LPARs. The presented setup used one LPAR for the database server, one LPAR for the application server and one LPAR for the Virtual I/O Server (VIOS). Not all machine resources were used for the test setup.

The NLS part was implemented as a LPAR running on a POWER6 based p570 with 3.5 GHz. Disk storage for both machines was provided by a Fiber Channel attached DS4800.

Figure 7: System Overview

In detail, the LPAR configuration was as follows:

Application Server:
- 1 LPAR on the Power 750 (POWER7@3.3GHz)
- 10 dedicated CPU cores with SMT4 enabled
- 32 GB physical memory

Database Server:
- 1 LPAR on the Power 750 (POWER7@3.3GHz)
- 6 dedicated CPU cores with SMT4 enabled
- 64 GB physical memory

NLS Server:
- 1 LPAR on the p570 (POWER6@3.5 GHz)
• 4 dedicated CPU cores with SMT2 enabled
• 32 GB physical memory

VIO Server:
• 1 LPAR on the Power 750 (POWER7@3.3Ghz)
• 1.4 CPUs entitlement, 2 virtual CPUs uncapped
  **Important:** The 1.4 CPU entitlement was not required for this use case. During our tests, we saw a very low CPU utilization on the VIO server that was typically less than 5%.
• 5 GB physical memory
• 1 dedicated dual port 4GB fiber channel card to attach to the DS4800

The application server LPAR and the database server LPAR were using all Ethernet and Fibre Channel connections through virtual adapters provided by the VIO server.

**SAP System Setup**

This section describes the SAP system configuration in more detail (see Figure 8).
Figure 8: SAP System and DB2 Database Layout

Details of the SAP NetWeaver BW system:

- The SAP NetWeaver BW application servers and the database server were running in two LPARs of the IBM Power 750 server.
- Two SAP NetWeaver BW application server instances were running in the same Power System LPAR (LPAR1). Both instances were connected to the first database partition (= DB Partition 0) of the database server.
- The database server was running in the second Power System LPAR (LPAR2). The database server was created with eight database partitions (administration partition + 7 data partitions) and it was connected to the SAN storage system DS4800.
- In the DS4800 system, 24 Fiber Channel disks were assigned to each database partition. The smaller tables (SAP base tables, dimension tables and the master data tables) were located on the disks assigned to database partition 0 (= P0). The large PSA, DSO and InfoCube tables of the different SAP NetWeaver BW staging areas were distributed over all disks assigned to the other database partitions P1 to P7.
- In the SAP NetWeaver BW application, an ABAP remote database connection was configured for the data transfer to and from the NLS system.
Details of the NLS system:

- The NLS server was installed on a separate Power System 570 server (POWER6).
- The data was stored on the same DS4800 storage system that was also used for the SAP NetWeaver BW system, however on separate disks.
- We used two alternative database server configurations in this PoC:
  - An NLS database server with eight database partitions
    12 Fiber Channel disks were assigned to each database partition. The dimension tables were stored on database partition 0. The large NLS tables (DSO and InfoCube fact tables) were distributed on all eight database partitions. The configuration with eight database partitions is shown in Figure 5.
  - An NLS database server with one database partition
    The same hardware resources (CPU cores, memory, I/O capacity) as used for the eight database partition scenario were assigned to the one database partition to have a comparable environment for the SAP NetWeaver BW NLS performance tests. For example, 96 (= 8 * 12) Fiber Channel disks were assigned to the one database partition. All tables were distributed over the 96 disks.

Test Scenarios

To perform an automatic mass data load to SAP NetWeaver BW, two process chains were used. Both process chains were run simultaneously and influenced each other by using background process events. The purpose of the first process chain was to dispatch data from the PIPE into the delta queue of DataSource 0RT_PA_TRAN_CONTROL and to trigger a subsequent InfoPackage to transfer the data into the PSA table. The process chain consisted of the following steps:

![Figure 9: Process Chain for Data Upload from PIPE](image)

1. Two days of point of sales data (between 23 and 24 million records) were dispatched from the PIPE and written to the delta queue. If a sufficient number of unprocessed requests were available in the PSA table or dispatching was locked, the process chain waited in this step.
2. The data that was written to the BW delta queue was extracted and inserted into the PSA table of 0RT_PA_TRAN_CONTROL.
3. To notify the second process chain that a request for processing was available in the PSA, a background event was raised.
4. The process chain was scheduled periodically and triggered itself once it was finished.

While the first process chain supplied the PSA table with new requests of data, the second chain loaded the data from the PSA to the different data targets. More specifically, different meta process chains were triggered that loaded the requests through the different layers of the EDW LSA. The following steps were processed:
1. First, it was checked whether unprocessed requests were available in the PSA table. If this was not the case, the process chain waited until unprocessed requests became available. If these requests were available, the chain continued.

2. The first meta chain loaded requests into the data propagation layer that consisted of a write optimized DSO. During the load, mostly general business enrichment took place that were computed on the SAP application server. Therefore, the main computation load during this step was on the application server in comparison to the DB2 database.

3. After data was loaded to the propagation layer, the first process chain was unlocked again and allowed to transfer new data to the PSA table. The next meta chain following the data propagation layer conducted several steps.

   At first, data was loaded to the business transformation layer that consisted of standard DSOs. These standard DSOs were semantically partitioned by quarter, which resulted in eight DSOs for two years. During this step, specific business enrichment that was processed on the SAP Application Server took place.

   After the loading process, the data in the standard DSOs was activated using the new MPP-optimized activation method that mainly created workload on the database server.

   Once the data had been activated, it was copied to the reporting layer that consisted of eight semantically-partitioned InfoCubes. Since no enrichment was performed in the transformation, the workload was mainly on the database LPAR.

   Requests that had been loaded into the standard DSOs more than 10 days earlier were deleted from the change logs.

Figure 10: Process Chain for Data Load into the InfoProviders of the LSA for POS DM
4. After the data had been enabled for reporting, it was loaded from the PSA table to the write-optimized DSOs in the corporate memory layer (eight DSOs for two years that were semantically partitioned by quarter). This was a 1:1 transformation to store the data unmodified to provide flexibility for cases where remodeling was necessary or redundancy was required to repair errors.

5. To prepare the process chain for the next run, successfully processed data was deleted from the PSA table and the data propagation DSO was purged.

6. The process chain was scheduled periodically.

The DTPs were configured with a parallelism of 25 batch processes. The DSO data activations ran in parallel on the seven database partitions where the DSOs were located. The data deletion jobs from PSA, change logs, and write-optimized DSOs used MDC fast rollout.

**Results**

The results of the following tests are discussed in this section:

- Extract/transform/load (ETL) operations from the PIPE to the InfoProviders of the layered scalable architecture for POS DM
- NLS: Archiving from InfoCubes and DataStore objects into the NLS database and execution of reporting queries in the BW system and the NLS system
- Impact of the advanced memory expansion (AME) feature on the ETL operations with AME active on the SAP application server.

**ETL Performance Evaluation**

The following table lists the available CPU and memory resources for the SAP application servers and the database during the data load tests:

<table>
<thead>
<tr>
<th>RAM size, GB</th>
<th>SAP NW BW Application Servers</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>32</td>
</tr>
<tr>
<td>Threads per core</td>
<td>4</td>
</tr>
<tr>
<td>CPU cores, total</td>
<td>6</td>
</tr>
<tr>
<td>CPU threads, total</td>
<td>24</td>
</tr>
</tbody>
</table>

*Table 1: CPU and Memory Resources for Database and Application Server LPAR*
The following table and Figure 11 show the achieved throughput when loading the POS data into the different layers of the layered scalable architecture for POS DM:

<table>
<thead>
<tr>
<th>Operation</th>
<th>Parallelism</th>
<th>Throughput</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load from the BW delta queue into PSA</td>
<td>10</td>
<td>75 mio records/hour</td>
<td></td>
</tr>
<tr>
<td>Load from PSA into the Data Propagation write-optimized DSO</td>
<td>25</td>
<td>48 mio records/hour</td>
<td>Complex transformations executed in the SAP application server</td>
</tr>
<tr>
<td>Load from the Data Propagation DSO into the Business Transformation standard DSOs</td>
<td>25</td>
<td>132 mio records/hour</td>
<td></td>
</tr>
<tr>
<td>DSO Data Activation</td>
<td>7 (DB2 inter-partition parallelism)</td>
<td>90 mio records/hour</td>
<td></td>
</tr>
<tr>
<td>Load from the Business Transformation DSOs into the POS &amp; Promotion Analysis InfoCubes</td>
<td>25</td>
<td>155 mio records/hour</td>
<td>Without SID generation</td>
</tr>
<tr>
<td>Load from PSA into the Corporate Memory write-optimized DSOs</td>
<td>25</td>
<td>206 mio records/hour</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Results of the Data Load Tests

**SAP NetWeaver BW: From Data Source to InfoProvider (IBM PoC) – Including DataSource split**

Figure 11: Results of Data Load Tests
To improve the performance of loading from the BW delta queue instead of only one DataSource, five DataSources were configured and the data transfer was split into five parallel processes, each one loading into a different Data Source. Each of the parallel processes ran approximately with a parallel degree of two.

The parallelization of DataSources needs to be implemented individually and is not available in the standard setup of the layered scalable architecture for POS DM. We were able to achieve a throughput of 47 Million records per hour without parallelization of the DataSources.

The following figures (Figure 12 and Figure 13) show the CPU usage on the SAP application server LPAR and the database server LPAR while 30 InfoPackages were being loaded from the PIPE through the LSA into the InfoCube for POS & Promotion Analysis. This also includes loading from PSA into corporate memory. Approximately, the overall throughput was **15.15 million records per hour** from PIPE into the InfoCube and into the corporate memory DSO.

![Figure 12: CPU Usage of SAP Application Server LPAR](image1)

**Figure 12: CPU Usage of SAP Application Server LPAR**

![Figure 13: CPU Usage of Database Server LPAR](image2)

**Figure 13: CPU Usage of Database Server LPAR**

The peaks on the application server LPAR occur during data load from PSA into the data propagation DSO. The data transfer process (DTP) calls complex transformations coded in ABAP, which causes the high CPU usage.

The peaks with high workload on the database LPAR occur during MPP-optimized DSO data activation and data load into the InfoCubes and into corporate memory.

The following table (Table 3) shows the average and maximum CPU usage on the application server and the database LPAR.
The following figure (Figure 14) shows the CPU usage on the SAP application servers and the database server while one InfoPackage was being loaded. The following abbreviations are used:

- **PSA** = Load from PIPE into PSA
- **DP DSO** = DTP from PSA into Data Propagation DSO (write-optimized)
- **BT DSO** = DTP from Data Propagation DSO into Business Transformation DSO (standard)
- **Data Act.** = Business Transformation DSO data activation
- **Cube** = DTP from Business Transformation DSO into POS & Promotion Analysis InfoCube
- **CM DSO** = DTP from PSA into Corporate Memory DSO (write-optimized)

### Table 3: CPU Usage during Data Load

<table>
<thead>
<tr>
<th></th>
<th>User CPU</th>
<th>System CPU</th>
<th>IO Wait</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average</td>
<td>Maximum</td>
<td>Average</td>
</tr>
<tr>
<td>Database Percent</td>
<td>42.34</td>
<td>98.8</td>
<td>4.78</td>
</tr>
<tr>
<td>Total</td>
<td>2.54</td>
<td>5.93</td>
<td>0.29</td>
</tr>
<tr>
<td>Application Servers Percent</td>
<td>52.71</td>
<td>96.3</td>
<td>2.18</td>
</tr>
<tr>
<td>Total</td>
<td>5.27</td>
<td>9.63</td>
<td>0.2</td>
</tr>
</tbody>
</table>

The following table shows the CPU usage for the database and application servers during data load.
Figure 14: CPU Usage during Load of a single InfoPackage

The figure also shows that the two process chains were running and thereby partly overlapping. While a new request was loaded from PIPE into PSA, the data of the previous request was activated in the business transformation DSO and loaded from the business transformation DSO into the InfoCube.
The following table (Table 4) shows the average and maximum CPU usage for each of the data load and activation operations.

<table>
<thead>
<tr>
<th>User CPU</th>
<th>System CPU</th>
<th>IO Wait</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Avg</td>
<td>Max</td>
</tr>
<tr>
<td>PSA → Data Propagation DSO</td>
<td>Database</td>
<td>%</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>%</td>
</tr>
<tr>
<td></td>
<td>Application Servers</td>
<td>%</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>%</td>
</tr>
<tr>
<td>Data Propagation DSO → Business Transformation DSO</td>
<td>Database</td>
<td>%</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>%</td>
</tr>
<tr>
<td></td>
<td>Application Servers</td>
<td>%</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>%</td>
</tr>
<tr>
<td>DSO data activation</td>
<td>Database</td>
<td>%</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>%</td>
</tr>
<tr>
<td></td>
<td>Application Servers</td>
<td>%</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>%</td>
</tr>
<tr>
<td>Business Transformation DSO → InfoCube</td>
<td>Database</td>
<td>%</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>%</td>
</tr>
<tr>
<td></td>
<td>Application Servers</td>
<td>%</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>%</td>
</tr>
<tr>
<td>PSA → Corporate Memory</td>
<td>Database</td>
<td>%</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>%</td>
</tr>
<tr>
<td></td>
<td>Application Servers</td>
<td>%</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>%</td>
</tr>
<tr>
<td>PIPE → PSA</td>
<td>Database</td>
<td>%</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>%</td>
</tr>
<tr>
<td></td>
<td>Application Servers</td>
<td>%</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>%</td>
</tr>
</tbody>
</table>

Table 4: CPU Usage during Load of one InfoPackage

Some tests for DSO data activation were run with MPP-optimized DSO data activation turned off. The throughput achieved in these runs was 43.7 million records per hour, which is about 48.5% of the throughput achieved with MPP-optimized DSO data activation.

**NLS Performance Evaluation**

During this PoC, the performance of the following NLS-related operations was measured:

- Data archiving from the SAP NetWeaver BW database to the NLS database and adjacent data delete in the SAP NetWeaver BW database.
- BW queries that access data in the SAP NetWeaver BW and the NLS database

The necessary system resources of an NLS system depend mainly on the NLS performance requirements and the data volume that is stored in the NLS database. In a real-life scenario, the NLS system is smaller than the SAP NetWeaver BW database server.

In the following list, the characteristics of the NLS system are compared with the resources of the SAP NetWeaver BW system:

- CPU power (measured in physical cores): 67% of the SAP NetWeaver BW system
- CPU parallelism (measured in available parallel threads): 33% of the SAP NetWeaver BW system
- RAM size (measured in Gigabytes): 50% of the SAP NetWeaver BW system
- Storage performance (measured in amount of disk spindles): 50% of the SAP NetWeaver BW system

<table>
<thead>
<tr>
<th>CPU architecture of the DB server</th>
<th>NLS System</th>
<th>SAP NW BW System</th>
<th>Ratio NLS / BW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of CPU cores of the DB server</td>
<td>POWER6</td>
<td>POWER7</td>
<td>67%</td>
</tr>
<tr>
<td>CPU threads per core</td>
<td>2</td>
<td>4</td>
<td>50%</td>
</tr>
<tr>
<td>Total number of CPU threads</td>
<td>8</td>
<td>24</td>
<td>33%</td>
</tr>
<tr>
<td>RAM size of the DB server in GB</td>
<td>32</td>
<td>64</td>
<td>50%</td>
</tr>
<tr>
<td>Amount of disk spindles used in the storage system</td>
<td>96</td>
<td>192</td>
<td>50%</td>
</tr>
</tbody>
</table>

Table 5: System Configuration used during NLS Tests

**NLS: Data Archiving Tests**

**Copy and Verification Phase**

The NLS interface performs the following steps during this phase:

- Reads data of the SAP NetWeaver BW object with (?) specified archiving time interval
- Writes data into the NLS database
- Verifies data to make sure that the transferred NLS data are correct and consistent

The original SAP NetWeaver BW fact table holds about 745 Million records. In total, 500 Million records were copied to the NLS database and were verified with 15 parallel running NLS archiving processes. Each process transferred approx. 33 million records.

In total, **185 million records per hour** were copied and verified.

The following table shows performance details of the SAP NetWeaver BW, the SAP application and NLS servers. During this test, NLS archiving was the only workload running on the SAP NetWeaver BW system.

<table>
<thead>
<tr>
<th>Average user CPU usage</th>
<th>Average CPU cores used</th>
<th>Average read/write activity in million bytes per second</th>
</tr>
</thead>
<tbody>
<tr>
<td>14%</td>
<td>0,9</td>
<td>8</td>
</tr>
<tr>
<td>56%</td>
<td>5,9</td>
<td>-</td>
</tr>
<tr>
<td>48%</td>
<td>2</td>
<td>25</td>
</tr>
</tbody>
</table>

Table 6: System Performance Characteristics during NLS Tests

The 15 parallel processes read the data from the SAP NetWeaver BW database server, transform the data into the new NLS object structure, and send the NLS data to the NLS database server. Significant CPU resources with a usage of 56% on the SAP application server LPAR are consumed for these processes. The CPU and I/O usage on the SAP NetWeaver BW database server is relatively low with a 14% CPU usage and 8 million bytes per second I/O activity. Approximately half of the CPU resources are consumed on the NLS database server, while its Read/Write activity was 25 MB per second in average. I/O usage on both database servers is low compared to a maximum I/O data transfer capacity of about 512 MB per second on both servers.
Deletion Phase
The final step of the NLS data archiving process is the data deletion phase. We tested the data deletion process with the following two different table layouts of a BW InfoCube:

- **MDC fast delete feature:**
  A BW InfoCube is created with MDC on the time characteristic column of the fact table that is used for data selection of an NLS data archiving package (that is, the MDC fast delete feature is used for data deletion).

- **Default DB2 data delete feature:**
  The time characteristics column of the fact table of the BW InfoCube is not created as an MDC column (that is, standard DB2 data deletion is used).

To ensure that the test conditions were the same in both tests, two identical InfoCubes with the same amount of data were used.

On both InfoCubes, data deletion was processed with 15 delete processes that were running sequentially, that is one package for one InfoCube at a time. Each process deleted approximately 33 million records.

At first, both BW InfoCubes had 745 Million records. 500 million records were deleted, and 245 Million records were left.

As described in chapter “Innovative Features Used”, MDC fast delete is a block-based data deletion method and, therefore, much faster than the normal row-oriented data deletion method.

The deletion tests illustrated that MDC fast delete is approx. factor 3.64 faster. For that reason, we recommend that you create the time-specific column in the fact table to be used for the selection of NLS data as MDC column.

<table>
<thead>
<tr>
<th>Runtime to delete 500 million records (hours)</th>
<th>MDC Fast Delete</th>
<th>Normal row based data delete (Non-MDC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of deleted records (million per hour)</td>
<td>0.915</td>
<td>3.33</td>
</tr>
<tr>
<td></td>
<td>546</td>
<td>150</td>
</tr>
</tbody>
</table>

Table 7: Deletion Performance

Overall Performance of the Data Transfer
As mentioned above, 185 Million records per hour were transferred from a BW InfoCube to the NLS database with 15 processes running in parallel. After that, 546 Million records per hour were deleted with MDC fast delete.

This means that 135 Million records per hour is the overall data transfer rate for NLS data archiving (including the data delete phase) measured in this environment.

You can use SAP NetWeaver BW process chains to automate the complete NLS data transfer process using parallel data copy and data verification, and sequential data deletion processing steps.

The NLS data archiving throughput is independent from the BW object type. It does not matter, whether an InfoCube or a DSO is the source of the NLS data archiving process. Comparable performance results for NLS archiving of DSO data were achieved.

NLS Query Performance Tests
The question how fast BW queries are running after data has been transferred to the NLS database is quite important. In the following section, we describe the results of the query performance tests of different scenarios.

Test Scenarios of the BW Query Tests

- Scenario 1:
  The BW database is created with seven database partitions. All records are stored in a BW InfoCube. The fact table is distributed on seven database (db) partitions.
- Scenario 2:
The NLS database is created with eight database partitions. All records are stored in an NLS InfoCube. The fact table is distributed on eight database (db) partitions.

- Scenario 3:
The NLS database has only one database partition. The same NLS database server as in scenario 2 is used. All NLS system resources are assigned to the one database partition. All records are stored in an NLS InfoCube. The fact table is located on one database (db) partition only.

In all three scenarios, the InfoCube contains the same data with 350 million records.

The same BW queries were processed in all three scenarios as follows:

- Single BW query run:
  One BW query aggregates 65 Million records and returns 2000 rows as result.

- Two BW query run:
  Two BW queries run in parallel. Each query reads different data from a different time interval. Each query aggregates 65 Million records and returns 2000 rows as result.

All BW queries access the InfoCube tables. There are no aggregates defined.

The following graphic displays the BW query results (Figure 15). The given query runtimes provide an average of three query runs.

![Figure 15: SAP NetWeaver BW Query Runtimes Accessing BW InfoCube versus NLS InfoCube](image)

The following are the results from the single BW query run:

- The runtime of the BW query accessing the BW InfoCube (scenario 1) was 50 seconds.
- The runtime of the same BW query, which was accessing the NLS data that was distributed over eight DB partitions (scenario 2), was only 33 seconds.
- The runtime of the query where the NLS object was located on only one DB partition was 296 seconds (scenario 3).
- The runtime of the BW query accessing the partitioned and well-distributed NLS data (scenario 2) is faster than the runtime of the BW query on the BW object (scenario 1) due to following reasons:
  - The NLS fact table is distributed on eight database partitions compared to seven database partitions of the BW object, which results in a higher degree of parallel processing.
The BW InfoCube is more complex than the NLS object and, therefore, more tables have to be joined.

- Clock speed of the CPU processor of the NLS database server (3.5 GHz) is about 6% higher than the one of the BW database server (3.3 GHz).

The following are the results of two BW queries running in parallel:

- The two parallel BW queries that accessed the BW object finished in 58 seconds (scenario 1). This runtime is only 8 seconds slower compared to the single BW query run because there are enough CPU resources (6 cores and 24 threads) available to run the two queries in parallel. Each query runs with parallel degree 7. Therefore 14 threads are needed for running the two queries in parallel.

- The two parallel BW queries that accessed the NLS object that was distributed on eight database partitions finished after 65 seconds (scenario 2). This is roughly double the time compared to the run of the single BW query because both queries competed for the CPU resources of the NLS database server (4 cores and 8 threads). Each BW query runs with parallel degree 8. Therefore, two queries running in parallel require 16 threads. Missing CPU resources result in sequential processing of the BW queries and longer runtimes.

- It takes 363 seconds to finish both queries that were accessing the NLS object stored on one database partition only (scenario 3). The query runtime increased by 22% compared to the run of the single BW query. This is a much lower runtime increase compared to scenario 2 (NLS distributed on eight database partitions) because each query runs in one thread and there are enough CPU resources (4 cores and 8 threads) on the NLS database server available to run the two queries in parallel.

Conclusion

- DPF can improve the BW query runtime on NLS objects significantly, for example, 33 seconds on an NLS database with eight db partition versus 296 seconds on an NLS database with one db partition. This has also been proven by many other SAP NetWeaver BW performance evaluations for BW queries running on BW objects.

- BW queries accessing an NLS object can run faster than BW queries accessing BW objects, depending on the parallel degree that the NLS database system is able to process.

- There are many options to configure an NLS system depending on performance and TCO requirements. You achieve a good performance of the NLS queries if your NLS system has enough CPU, main memory and I/O resources to process BW queries in parallel. You can also configure the NLS system with less hardware resources and reduced TCO to run the NLS queries with a lower parallel degree, which results in longer NLS query response time.

POWER7 AME Tests

The following section describes the performance evaluation tests with the POWER7 Active Memory Expansion (AME) feature.

We ran the AME tests with different AME memory expansion factors for the SAP application server LPAR.

POWER7 AME Activation for the SAP Application Server LPAR

As test case for several different AME settings, we executed the ETL process chains for one data package with 23 million records. With AME turned off, the main memory usage of the SAP application server LPAR was between 12 and 32 GB.

The following AME configuration settings were tested with the same ETL workload:

- AME = OFF and 32 GB physical main memory
- AME = 1.6 and 20 GB physical main memory
- AME = 2.6 and 12 GB physical main memory
- AME = OFF and 20 GB physical main memory

Based on initial recommendations from the AME planning tool “Active Memory Expansion Planning and Advisory Tool” (amepat), we expected the data load performance with AME 1.6 to be almost the same as the performance without AME activation.

In addition, we expected the data load performance with AME compression factor 2.6 would slightly decrease compared to the non-AME test scenario due to the higher degree of memory compression. Therefore, we used those two expansion factors to verify the performance impact of main memory reduction provided by AME.

The following table and figure 16 provide the results of the different AME tests:

<table>
<thead>
<tr>
<th></th>
<th>AME = OFF</th>
<th>AME = 1.6</th>
<th>AME=2.6</th>
<th>AME=OFF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical Memory (GB)</td>
<td>32</td>
<td>20</td>
<td>12</td>
<td>20</td>
</tr>
<tr>
<td>Expanded Memory (GB)</td>
<td>32</td>
<td>32</td>
<td>32</td>
<td>20</td>
</tr>
<tr>
<td>Average User CPU (%)</td>
<td>48</td>
<td>48</td>
<td>47.5</td>
<td>33.3</td>
</tr>
<tr>
<td>Average Sys CPU (%)</td>
<td>1.8</td>
<td>2.8</td>
<td>6.4</td>
<td>2</td>
</tr>
<tr>
<td>Average I/O wait (%)</td>
<td>0</td>
<td>0</td>
<td>0.3</td>
<td>10.2 (paging)</td>
</tr>
<tr>
<td>Runtime (seconds)</td>
<td>7398</td>
<td>7413</td>
<td>7920</td>
<td>10600</td>
</tr>
</tbody>
</table>

Table 8: Results of POWER7 AME Tests for the SAP Application Server LPAR

![Figure 16: Results of POWER7 AME Tests for the SAP Application Server LPAR](image)

The effective or expanded memory (32 GB) stays the same with the settings AME=OFF, AME=1.6 and AME=2.6. To accomplish this, the physical memory was reduced to 20 GB with AME = 1.6 and to 12 GB with AME=2.6. We also performed a test with AME=OFF and with 20 GB physical memory to evaluate the performance impact in case there is a constraint on physical memory.

During the first three tests, we saw only little impact on the average CPU usage and the average I/O wait time. The values were approximately the same.
As expected, the average system CPU usage increased with a larger AME expansion factor due to more compression and decompression activities.

Setting the AME expansion factor to 1.6 did not impact the performance of ETL processing. That is, physical memory could be decreased from 32 GB to 20 GB without increasing the runtime.

The runtime of ETL processing increased by 7% with AME set to 2.6 and a physical memory of 12 GB.

The last performance test with AME set to OFF and a physical memory reduction to 20 GB showed a significant increase of the runtime by 43%. There were a lot of OS paging activities because main memory size was too small for this workload. This is a good example that shows how SAP application servers can benefit from AME.

**Best Practices**

In the following, we provide recommendations for the configuration of the SAP system, the DB2 database, and for the usage of the amepat tool.

**SAP Configuration for AIX**

Two SAP instance profile parameters were set on the application server as they are recommended when running an SAP instance on AIX (for more information, see also SAP Note 1048686 – Recommended AIX Settings for SAP):

```
ES/TABLE=SHM_SEGS
ES/SHM_SEGS_VERSION=2
```

On AIX, with those parameters, the SAP memory management performs better than with the default value ES/TABLE=UNIX_STD. In addition, the memory usage adapts better to the current workload and to the active users on the system.

**Active Memory Expansion Planning and Advisory Tool**

AIX provides the tool amepat. It supports you in the planning process for later production use of AME. The tool can run in advance with the given workload to profile the memory usage, the access pattern to the memory, and the compressibility of the memory pages. amepat has been available since AIX 6.1 TL4 SP2 and can even run on older POWER generations which do not support AME. This supports the planning process when you want to move workload from older machines to POWER7 based servers. In this PoC, we used amepat to identify the compression factor at which we should expect compression mechanisms to impact process execution time. Therefore, we profiled the workload in advance, set the suggested compression factor as starting point and checked the behavior later with the tool again. It turned out that, for the workload under test, a compression factor of 1.6 provided almost the same performance with minimal increased CPU requirements. This is also reflected in the recommendations from the amepat output as follows:
Figure 17: Sample Output of amepat

According to this amepat output, the CPU requirements increase significantly as soon as the expansion factor exceeds 1.6. The measured data (see also chapter “POWER7 AME Tests”) supports the estimation from the tool.

During our test we observed the following:

- The quality of the recommendations given by amepat depends on the number of samples and the workload on the system. This is somehow obvious since only with representative statistical data reliable predictions can be made. So when you use amepat make sure that the workload you measure is representative and that there are enough samples available.

- We also recommend monitoring the system with tools like nmon or topasrec before and after the change of the expansion factor to confirm that the system is behaving as expected and planned.

- When using amepat it is useful to collect the data with the option –R. This allows you to afterwards create different reports from amepat without rerunning the workload.

DB2 for LUW

- DPF layout in general

You find recommendations about the number of database partitions and the data distribution for a DB2 database for an SAP NetWeaver BW system in the separate document “SAP NetWeaver Business Warehouse 7.3 – Administration Tasks: IBM DB2 for Linux, UNIX, and Windows” that is available at:

http://service.sap.com/instguidesnw →<Your SAP NetWeaver Main Release> → Operations → Database-Specific Guides

- Usage of MDC

As of SAP NetWeaver 7.3, the usage of MDC for PSA and PSA-like tables (for example, DSO activation queue and DSO change log tables) is the default. With MDC, the performance of the deletion of InfoPackages can be improved considerably. For InfoCubes, selecting MDC on the package dimension column of F fact tables speeds up the manual deletion of InfoPackages and the deletion of InfoPackages during InfoCube compression.
• Recommendations for MPP-Optimized DSO Data Activation

We recommend that you use DB2 V9.7 FP3SAP or higher for optimal performance of the MPP-optimized DSO data activation. This Fix Pack contains APAR IC69515 that speeds up the insertion of records into the change log table.

Implement SAP Note 1515687 or Support Package 2 of SAP NetWeaver BW 7.3. To activate the optimizations provided in this SAP Note, set RSADMIN parameter DB6_MPP_DSO_TABLES_COLLOCATED to YES, make sure that the activation queue tables are empty, and reactivate the DSOs that are eligible for MPP-optimized data activation. The activation queue tables are dropped and re-created with the same distribution key as the active table and the new join table. This reduces the amount of data that needs to be transferred between database partitions during processing of MPP-optimized data activation.

• Recommendations for NLS

MDC on time dimension key column of InfoCubes instead of 0FISCPER or 0CALMONTH can speed up deletion of data from InfoCubes that has been archived to NLS.

You can use DPF to speed up processing of queries that access data in the NLS database.
Summary

Fast data processing and efficient management of Point of Sales (POS) data in an Enterprise Data Warehouse (EDW) with a low total cost of ownership (TCO) are the challenging business demands of today's retail business. The SAP Retail solution with the new layered scalable architecture running on IBM DB2 for LUW on IBM POWER7 servers covers these business requirements perfectly.

The IBM POWER7 series, launched in 2010, delivers unprecedented performance for both transactional and throughput computing.

IBM DB2 is an industry-leading database solution that delivers maximum performance, scalability and reliability with a wide range of operating systems and platforms. DB2 has been constantly optimized to support SAP environments over the last 10 years. It boasts a wealth of built-in features and configurations that help you get the most from your SAP software, lowering TCO significantly boosting performance.

Extensive testing has demonstrated that this winning combination of SAP BW/SAP Retail, DB2 for LUW and IBM POWER7 meets the needs of today's retailers by delivering high performance, fast processing and efficient management of large data volumes of POS data in real time. The system used in our evaluation, based on a mid-sized POWER7 server, was able to handle data volumes of over 8 billion POS data records.

IBM has optimized its hardware and database to offer the best possible platform for solutions from the SAP for Retail portfolio. These enhancements help retail organizations slash TCO, boost performance and scalability, reduce maintenance effort, and support the entire data lifecycle.
Appendix

List of Abbreviations

AME = Active Memory Expansion
AMS = Active Memory Sharing
BWA = Business Warehouse Accelerator
DM = Data Management
DPF = Database Partitioning Feature
DSO = DataStore object
DTP = Data Transfer Process
DWH = Data Warehouse
EDW = Enterprise Data Warehouse
EIM = Enterprise Information Management
ETL = Extract/Transform/Load
LPAR = Logical Partition
LSA = Layered Scalable Architecture
MDC = Multi-Dimensional Clustering
MPP = Massive Parallel Processing
NLS = Near-Line Storage
OLAP = Online Analytical Processing
PIPE = POS Inbound Processing Engine
PoC = Proof of Concept
POS = Point of Sales
PSA = Persistent Staging Area
ROI = Return on Investment
SAN = Storage Area Network
SMT = Symmetric Multi-Threading
TCO = Total Cost of Ownership
VIOS = Virtual IO Server
VMM = Virtual Memory Manager

DB2 Configuration Details

DB2 Registry

[e] DB2DB0FF=05S
[i] DB2_BLOCKING_WITHHOLD_LOBLOCATOR=NO [DB2_WORKLOAD]
[i] DB2_AGENT_CACHING_FMP=OFF [DB2_WORKLOAD]
[i] DB2_TRUST_MDC_BLOCK_FULL_HINT=YES [DB2_WORKLOAD]
[i] DB2_CREATE_INDEX_COLLECT_STATS=YES [DB2_WORKLOAD]
[i] DB2_ATS_ENABLE=YES [DB2_WORKLOAD]
[i] DB2_RESTRICT_DDF=YES [DB2_WORKLOAD]
[i] DB2_DUMP_SECTION_ENV=YES [DB2_WORKLOAD]
[i] DB2_OPT_MAX_TEMP_SIZE=10240 [DB2_WORKLOAD]
Database Manager Configuration

Node type = Enterprise Server Edition with local and remote clients

Database manager configuration release level = 0x0d00

CPU speed (millisec/instruction) (CPUSPEED) = 2.755341e-07
Communications bandwidth (MB/sec) (COMM_BANDWIDTH) = 1.000000e+02

Max number of concurrently active databases (NUMDB) = 8
Federated Database System Support (FEDERATED) = NO
Transaction processor monitor name (TP_MON_NAME) =

Default charge-back account (DFT_ACCOUNT_STR) =

Java Development Kit installation path (JDK_PATH) = /db2/db2os5/sqllib/java/jdk64

Diagnostic error capture level (DIAGLEVEL) = 3
Notify Level (NOTIFYLEVEL) = 3
Diagnostic data directory path (DIAGPATH) = /db2/05/db2dump
Size of rotating db2diag & notify logs (MB) (DIAGSIZE) = 0

Default database monitor switches

Buffer pool (DFT_MON_BUFPOOL) = ON
Lock (DFT_MON_LOCK) = ON
Sort (DFT_MON_SORT) = ON
Statement (DFT_MON_STMT) = ON
Table (DFT_MON_TABLE) = ON
Timestamp (DFT_MON_TIMESTAMP) = ON
Unit of work (DFT_MON_UOW) = ON
Monitor health of instance and databases (HEALTH_MON) = OFF

SYSADM group name (SYSADM_GROUP) = DBOS5ADM
SYSCtrl group name (SYSCtrl_GROUP) = DBOS5CTL
SYsMAINT group name (SYsMAINT_GROUP) = DBOS5MNT
SYsMON group name (SYsMON_GROUP) = DBOS5MON

Client Userid-Password Plugin (CLNT_PW_PLUGIN) =
Client Kerberos Plugin (CLNT_KRB_PLUGIN) =
Group Plugin (GROUP_PLUGIN) =

<table>
<thead>
<tr>
<th>Configuration Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSS Plugin for Local Authorization</td>
<td>(LOCAL_GSSPLUGIN) =</td>
</tr>
<tr>
<td>Server Plugin Mode</td>
<td>(SRV_PLUGIN_MODE) = UNFENCED</td>
</tr>
<tr>
<td>Server List of GSS Plugins</td>
<td>(SRVCON_GSSPLUGIN_LIST) =</td>
</tr>
<tr>
<td>Server Userid-Password Plugin</td>
<td>(SRVCON_PW_PLUGIN) =</td>
</tr>
<tr>
<td>Server Connection Authentication</td>
<td>(SRVCON_AUTH) = NOT_SPECIFIED</td>
</tr>
<tr>
<td>Cluster manager</td>
<td>(CLUSTER_MGR) =</td>
</tr>
<tr>
<td>Database manager authentication</td>
<td>(AUTHENTICATION) = SERVER_ENCRYPT</td>
</tr>
<tr>
<td>Alternate authentication</td>
<td>(ALTERNATE_AUTH_ENC) =</td>
</tr>
<tr>
<td>Cataloging allowed without authority</td>
<td>(CATALOG_NOAUTH) =</td>
</tr>
<tr>
<td>Trust all clients</td>
<td>(TRUST_ALLCLNTS) = YES</td>
</tr>
<tr>
<td>Trusted client authentication</td>
<td>(TRUST_CLNTAUTH) = CLIENT</td>
</tr>
<tr>
<td>Bypass federated authentication</td>
<td>(FED_NOAUTH) = NO</td>
</tr>
<tr>
<td>Default database path</td>
<td>(DFTDBPATH) = /db2/OS5</td>
</tr>
<tr>
<td>Database monitor heap size (4KB)</td>
<td>(MON_HEAP_SZ) = AUTOMATIC(90)</td>
</tr>
<tr>
<td>Java Virtual Machine heap size (4KB)</td>
<td>(JAVA_HEAP_SZ) = 2048</td>
</tr>
<tr>
<td>Audit buffer size (4KB)</td>
<td>(AUDIT_BUF_SZ) = 0</td>
</tr>
<tr>
<td>Size of instance shared memory (4KB)</td>
<td>(INSTANCE_MEMORY) = 3040870</td>
</tr>
<tr>
<td>Backup buffer default size (4KB)</td>
<td>(BACKBUFSZ) = 1024</td>
</tr>
<tr>
<td>Restore buffer default size (4KB)</td>
<td>(RESTBUFSZ) = 1024</td>
</tr>
<tr>
<td>Agent stack size</td>
<td>(AGENT_STACK_SZ) = 4096</td>
</tr>
<tr>
<td>Sort heap threshold (4KB)</td>
<td>(SHEAPTHRES) = 0</td>
</tr>
<tr>
<td>Directory cache support</td>
<td>(DIR_CACHE) = NO</td>
</tr>
<tr>
<td>Application support layer heap size (4KB)</td>
<td>(ASLHEAPSZ) = 16</td>
</tr>
<tr>
<td>Max requester I/O block size (bytes)</td>
<td>(RORIOBLK) = 65000</td>
</tr>
<tr>
<td>Query heap size (4KB)</td>
<td>(QUERY_HEAP_SZ) = 1000</td>
</tr>
<tr>
<td>Workload impact by throttled utilities(UTIL_IMPACT_LIM)</td>
<td>= 10</td>
</tr>
<tr>
<td>Priority of agents</td>
<td>(AGENTPRI) = SYSTEM</td>
</tr>
<tr>
<td>Agent pool size</td>
<td>(NUM_POOLAGENTS) = AUTOMATIC(100)</td>
</tr>
<tr>
<td>Initial number of agents in pool</td>
<td>(NUM_INITAGENTS) = 5</td>
</tr>
<tr>
<td>Max number of coordinating agents</td>
<td>(MAX_COORDAGENTS) = AUTOMATIC(200)</td>
</tr>
<tr>
<td>Max number of client connections</td>
<td>(MAXgetConnection) = AUTOMATIC(MAX_COORDAGENTS)</td>
</tr>
<tr>
<td>Keep fenced process</td>
<td>(KEEPFENCED) = YES</td>
</tr>
<tr>
<td>Number of pooled fenced processes</td>
<td>(FENCED_POOL) = 10</td>
</tr>
<tr>
<td>Initial number of fenced processes</td>
<td>(NUM_INITFENCED) = 0</td>
</tr>
<tr>
<td>Index re-creation time and redo index build (INDEXREC)</td>
<td>= RESTART</td>
</tr>
<tr>
<td>Transaction manager database name</td>
<td>(TM_DATABASE) = 1ST_CONN</td>
</tr>
<tr>
<td>Transaction resync interval (sec)</td>
<td>(RESYNC_INTERVAL) = 100</td>
</tr>
<tr>
<td>SPM name</td>
<td>(SPM_NAME) =</td>
</tr>
<tr>
<td>SPM log size</td>
<td>(SPM_LOG_FILE_SZ) = 256</td>
</tr>
<tr>
<td>SPM resync agent limit</td>
<td>(SPM_MAX_RESYNC) = 20</td>
</tr>
<tr>
<td>SPM log path</td>
<td>(SPM_LOG_PATH) =</td>
</tr>
<tr>
<td>TCP/IP Service name</td>
<td>(SVCCNAME) = sapdb2OS5</td>
</tr>
<tr>
<td>Discovery mode</td>
<td>(DISCOVER) = SEARCH</td>
</tr>
<tr>
<td>Discover server instance</td>
<td>(DISCOVER_INST) = ENABLE</td>
</tr>
<tr>
<td>SSL server keydb file</td>
<td>(SSL_SVR_KEYDB) =</td>
</tr>
<tr>
<td>SSL server stash file</td>
<td>(SSL_SVR_STASH) =</td>
</tr>
<tr>
<td>SSL server certificate label</td>
<td>(SSL_SVR_LABEL) =</td>
</tr>
<tr>
<td>SSL service name</td>
<td>(SSL_SVCNAME) =</td>
</tr>
<tr>
<td>SSL cipher specs</td>
<td>(SSL_CIPHERSPECS) =</td>
</tr>
<tr>
<td>SSL versions</td>
<td>(SSL_VERSIONS) =</td>
</tr>
<tr>
<td>SSL client keydb file</td>
<td>(SSL_CLNT_KEYDB) =</td>
</tr>
<tr>
<td>SSL client stash file</td>
<td>(SSL_CLNT_STASH) =</td>
</tr>
<tr>
<td>Maximum query degree of parallelism</td>
<td>(MAX_QUERYDEGREE) = 1</td>
</tr>
<tr>
<td>Enable intra-partition parallelism</td>
<td>(INTRA_PARALLEL) = NO</td>
</tr>
<tr>
<td>Maximum Asynchronous TQs per query</td>
<td>(FEDERATED_ASYNC) = 0</td>
</tr>
<tr>
<td>No. of int. communication buffers(4KB)</td>
<td>(FCM_NUM_BUFFERS) = AUTOMATIC(4096)</td>
</tr>
</tbody>
</table>
No. of int. communication channels (FCM_NUM_CHANNELS) = AUTOMATIC(2048)
Node connection elapse time (sec) (CONN_ELAPSE) = 10
Max number of node connection retries (MAX_CONNRETRIES) = 5
Max time difference between nodes (min) (MAX_TIME_DIFF) = 60

db2start/db2stop timeout (min) (START_STOP_TIME) = 10

Database Configuration

Database Partition 0

Database Configuration for Database os5

Database configuration release level = 0x0d00
Database release level = 0x0d00

Database territory = en_US
Database code page = 1208
Database code set = UTF-8
Database country/region code = 1
Database collating sequence = IDENTIY_16BIT
Alternate collating sequence (ALT_COLLATE) =
Number compatibility = OFF
Varchar2 compatibility = OFF
Date compatibility = OFF
Database page size = 16384

Dynamic SQL Query management (DYN_QUERY_MGMT) = DISABLE
Statement concentrator (STMT_CONC) = OFF

Discovery support for this database (DISCOVER_DB) = ENABLE
Restrict access = NO
Default query optimization class (DFT_QUERYOPT) = 5
Degree of parallelism (DFT_DEGREE) = ANY
Continue upon arithmetic exceptions (DFT_SQLMATHWARN) = NO
Default refresh age (DFT_REFRESH_AGE) = 0
Default maintained table types for opt (DFT_MTTB_TYPES) = SYSTEM
Number of frequent values retained (NUM_FREQVALUES) = 10
Number of quantiles retained (NUM_QUANTILES) = 20

Decimal floating point rounding mode (DECFLT_ROUNDING) = ROUND_HALF_EVEN
Backup pending = NO
All committed transactions have been written to disk = NO
Rollforward pending = NO
Restore pending = NO
Multi-page file allocation enabled = YES
Log retain for recovery status = NO
User exit for logging status = NO

Self tuning memory (SELF_TUNING_MEM) = OFF
Size of database shared memory (4KB) (DATABASE_MEMORY) = COMPUTED(3089840)
Database memory threshold (DB_MEM_THRESH) = 10
Max storage for lock list (4KB) (LOCKLIST) = 160000
Percent. of lock lists per application (MAXLOCKS) = 90
Package cache size (4KB) (PCKCACHESZ) = 18000
Sort heap thres for shared sorts (4KB) (SHEAPTHRES_SHR) = 450000
Sort list heap (4KB) (SORTHEAP) = 15000

Database heap (4KB) (DBHEAP) = AUTOMATIC(2565)
Catalog cache size (4KB) (CATALOGCACHE_SZ) = 2560
Log buffer size (4KB) (LOGBUFSZ) = 2048
Utilities heap size (4KB) (UTIL_HEAP_SZ) = 50000
Buffer pool size (BUFFPAGE) = 10000
SQL statement heap (4KB) (STMTHEAP) = AUTOMATIC(4096)
Default application heap (4KB) (APPLHEAPSZ) = AUTOMATIC(256)
Application Memory Size (4KB) (APPL_MEMORY) = AUTOMATIC(40000)
<table>
<thead>
<tr>
<th>Configuration Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statistics heap size (4KB)</td>
<td>(STAT_HEAP_SZ) = AUTOMATIC(4384)</td>
</tr>
<tr>
<td>Interval for checking deadlock (ms)</td>
<td>(DLCHKTIME) = 10000</td>
</tr>
<tr>
<td>Lock timeout (sec)</td>
<td>(LOCKTIMEOUT) = 3000</td>
</tr>
<tr>
<td>Changed pages threshold</td>
<td>(CHNGPGS_THRESH) = 20</td>
</tr>
<tr>
<td>Number of asynchronous page cleaners</td>
<td>(NUMIOCLEANERS) = AUTOMATIC(2)</td>
</tr>
<tr>
<td>Number of I/O servers</td>
<td>(NUMIOSERVERS) = AUTOMATIC(8)</td>
</tr>
<tr>
<td>Index sort flag</td>
<td>(INDEXSORT) = YES</td>
</tr>
<tr>
<td>Sequential detect flag</td>
<td>(SEQUDETECT) = YES</td>
</tr>
<tr>
<td>Default prefetch size (pages)</td>
<td>(DFT_PREFETCH_SZ) = AUTOMATIC</td>
</tr>
<tr>
<td>Track modified pages</td>
<td>(TRACKMOD) = ON</td>
</tr>
<tr>
<td>Default number of containers</td>
<td>= 1</td>
</tr>
<tr>
<td>Default tablespace extentsize (pages)</td>
<td>(DFT_EXTENT_SZ) = 2</td>
</tr>
<tr>
<td>Max number of active applications</td>
<td>(MAXAPPLS) = AUTOMATIC(191)</td>
</tr>
<tr>
<td>Average number of active applications</td>
<td>(AVG_APPLS) = AUTOMATIC(3)</td>
</tr>
<tr>
<td>Max DB files open per application</td>
<td>(MAXFILOP) = 61440</td>
</tr>
<tr>
<td>Log file size (4KB)</td>
<td>(LOGFILSIZ) = 163800</td>
</tr>
<tr>
<td>Number of primary log files</td>
<td>(LOGPRIMARY) = 60</td>
</tr>
<tr>
<td>Number of secondary log files</td>
<td>(LOGSECOND) = 80</td>
</tr>
<tr>
<td>Changed path to log files</td>
<td>(NEWLOGPATH) =</td>
</tr>
<tr>
<td>Path to log files</td>
<td>/SAN_disks/db2_N0_disk1/OS5/log_dir/NODE0000/</td>
</tr>
<tr>
<td>Overflow log path</td>
<td>(OVERFLOWLOGPATH) =</td>
</tr>
<tr>
<td>Mirror log path</td>
<td>(MIRRORLOGPATH) =</td>
</tr>
<tr>
<td>First active log file</td>
<td></td>
</tr>
<tr>
<td>Block log on disk full</td>
<td>(BLK_LOG_DSK_FUL) = YES</td>
</tr>
<tr>
<td>Block non logged operations</td>
<td>(BLOCKNONLOGGED) = NO</td>
</tr>
<tr>
<td>Percent max primary log space by transaction (MAX_LOG)</td>
<td>0</td>
</tr>
<tr>
<td>Num. of active log files for 1 active UOW(NUM_LOG_SPAN)</td>
<td>0</td>
</tr>
<tr>
<td>Group commit count</td>
<td>(MINCOMMIT) = 1</td>
</tr>
<tr>
<td>Percent log file reclaimed before soft chkpt (SOFTMAX)</td>
<td>50</td>
</tr>
<tr>
<td>Log retain for recovery enabled</td>
<td>(LOGRETAIN) = OFF</td>
</tr>
<tr>
<td>User exit for logging enabled</td>
<td>(USEREXIT) = OFF</td>
</tr>
<tr>
<td>HADR database role</td>
<td>= STANDARD</td>
</tr>
<tr>
<td>HADR local host name</td>
<td>(HADR_LOCAL_HOST) =</td>
</tr>
<tr>
<td>HADR local service name</td>
<td>(HADR_LOCAL_SVC) =</td>
</tr>
<tr>
<td>HADR remote host name</td>
<td>(HADR_REMOTE_HOST) =</td>
</tr>
<tr>
<td>HADR remote service name</td>
<td>(HADR_REMOTE_SVC) =</td>
</tr>
<tr>
<td>HADR instance name of remote server</td>
<td>(HADR_REMOTE_INST) =</td>
</tr>
<tr>
<td>HADR timeout value</td>
<td>(HADR_TIMEOUT) = 120</td>
</tr>
<tr>
<td>HADR log write synchronization mode</td>
<td>(HADR_SYNCMODE) = NEARSYNC</td>
</tr>
<tr>
<td>HADR peer window duration (seconds)</td>
<td>(HADR_PEER_WINDOW) = 0</td>
</tr>
<tr>
<td>First log archive method</td>
<td>(LOGARCHMETH1) = OFF</td>
</tr>
<tr>
<td>Options for logarchmeth1</td>
<td>(LOGARCHOPT1) =</td>
</tr>
<tr>
<td>Second log archive method</td>
<td>(LOGARCHMETH2) = OFF</td>
</tr>
<tr>
<td>Options for logarchmeth2</td>
<td>(LOGARCHOPT2) =</td>
</tr>
<tr>
<td>Failover log archive path</td>
<td>(FAILARCHPATH) =</td>
</tr>
<tr>
<td>Number of log archive retries on error</td>
<td>(NUMARCHRETRY) = 5</td>
</tr>
<tr>
<td>Log archive retry Delay (secs)</td>
<td>(ARCHRETRYDELAY) = 20</td>
</tr>
<tr>
<td>Vendor options</td>
<td>(VENDOROPT) =</td>
</tr>
<tr>
<td>Auto restart enabled</td>
<td>(AUTORESTART) = ON</td>
</tr>
<tr>
<td>Index re-creation time and redo index build</td>
<td>(INDEXREC) = SYSTEM (RESTART)</td>
</tr>
<tr>
<td>Log pages during index build</td>
<td>(LOGINDEXBUILD) = OFF</td>
</tr>
<tr>
<td>Default number of loadrec sessions</td>
<td>(DFT_LOADREC_SES) = 1</td>
</tr>
<tr>
<td>Number of database backups to retain</td>
<td>(NUM_DB_BACKUPS) = 12</td>
</tr>
<tr>
<td>Recovery history retention (days)</td>
<td>(REC_HIS_RETENTN) = 60</td>
</tr>
<tr>
<td>Auto deletion of recovery objects</td>
<td>(AUTO_DEL_REC_OBJ) = OFF</td>
</tr>
<tr>
<td>TSM management class</td>
<td>(TSM_MGMTCLASS) =</td>
</tr>
<tr>
<td>TSM node name</td>
<td>(TSM_NODENAME) =</td>
</tr>
<tr>
<td>TSM owner</td>
<td>(TSM_OWNER) =</td>
</tr>
<tr>
<td>TSM password</td>
<td>(TSM_PASSWORD) =</td>
</tr>
<tr>
<td>Automatic maintenance</td>
<td>(AUTO_MAINT) = ON</td>
</tr>
</tbody>
</table>
Automatic database backup (AUTO_DB_BACKUP) = OFF
Automatic table maintenance (AUTO_TBL_MAINT) = ON
Automatic runstats (AUTO_RUNSTATS) = ON
Automatic statement statistics (AUTO_STMT_STATS) = ON
Automatic statistics profiling (AUTO_STATS_PROF) = OFF
Automatic profile updates (AUTO_PROF_UPD) = OFF
Automatic reorganization (AUTO_REORG) = OFF

Auto-Revalidation (AUTO_REVAL) = DEFERRED
Currently Committed (CUR_COMMIT) = ON
CHAR output with DECIMAL input (DEC_TO_CHAR_FMT) = NEW
Enable XML Character operations (ENABLE_XMLCHAR) = YES
WLM Collection Interval (minutes) (WLM_COLLECT_INT) = 0

Monitor Collect Settings
Request metrics (MON_REQ_METRICS) = BASE
Activity metrics (MON_ACT_METRICS) = NONE
Object metrics (MON_OBJ_METRICS) = BASE
Unit of work events (MON_UOW_DATA) = NONE
Lock timeout events (MON_LOCKTIMEOUT) = WITHOUT_HIST
Deadlock events (MON_DEADLOCK) = WITHOUT_HIST
Lock wait events (MON_LOCKWAIT) = NONE
Lock wait event threshold (MON_LW_THRESH) = 5000000
Number of package list entries (MON_PKGLIST_SZ) = 32
Lock event notification level (MON_LCK_MSG_LVL) = 1
SMTP Server (SMTP_SERVER) =
SQL conditional compilation flags (SQL_CCFLAGS) =
Section actuals setting (SECTION_ACTUALS) = NONE

Database Partitions 1-7

Database Configuration for Database os5

Database configuration release level = 0x0d00
Database release level = 0x0d00
Database territory = en_US
Database code page = 1208
Database code set = UTF-8
Database country/region code = 1
Database collating sequence = IDENTITY_16BIT
Alternate collating sequence (ALT_COLLATE) =
Number compatibility = OFF
Varchar2 compatibility = OFF
Date compatibility = OFF
Database page size = 16384

Dynamic SQL Query management (DYN_QUERY_MGMT) = DISABLE
Statement concentrator (STMT_CONC) = OFF
Discovery support for this database (DISCOVER_DB) = ENABLE
Restrict access = NO
Default query optimization class (DFT_QUERYOPT) = 5
Degree of parallelism (DFT_DEGREE) = ANY
Continue upon arithmetic exceptions (DFT_SqlMATHWarn) = NO
Default refresh age (DFT_REFRESH_AGE) = 0
Default maintained table types for opt (DFT_MTTB_TYPES) = SYSTEM
Number of frequent values retained (NUM_FREQVALUES) = 10
Number of quantiles retained (NUM_QUANTILES) = 20

Decimal floating point rounding mode (DECFLT_ROUNDING) = ROUND_HALF_EVEN
Backup pending = NO
All committed transactions have been written to disk = NO
Rollforward pending = NO
Restore pending = NO
Multi-page file allocation enabled = YES
Log retain for recovery status = NO
User exit for logging status = NO

Self tuning memory (SELF_TUNING_MEM) = OFF
Size of database shared memory (4KB) (DATABASE_MEMORY) = COMPUTED(2136384)
Database memory threshold (DB_MEM_THRESH) = 10
Max storage for lock list (4KB) (LOCKLIST) = 160000
Percent. of lock lists per application (MAXLOCKS) = 90
Package cache size (4KB) (PCCKACHESZ) = 180000
Sort heap thres for shared sorts (4KB) (SHEAPTHRES_SHR) = 450000
Sort list heap (4KB) (SORTHEAP) = 150000

Database heap (4KB) (DBHEAP) = AUTOMATIC(2565)
Catalog cache size (4KB) (CATALOGCACHE_SZ) = 2560
Log buffer size (4KB) (LOGBUFSZ) = 2048
Utilities heap size (4KB) (UTIL_HEAP_SZ) = 50000
Buffer pool size (pages) (BUFFPAGE) = 10000
SQL statement heap size (4KB) (STMTHEAP) = AUTOMATIC(4096)
Application Memory Size (4KB) (APPL_MEMORY) = AUTOMATIC(40000)
Statistics heap size (4KB) (STAT_HEAP_SZ) = AUTOMATIC(4384)

Interval for checking deadlock (ms) (DLCHKTIME) = 10000
Lock timeout (sec) (LOCKTIMEOUT) = 3600

Changed pages threshold (CHNGPGS_THRESH) = 20
Number of asynchronous page cleaners (NUM_IOCLEANERS) = AUTOMATIC(2)
Number of I/O servers (NUM_IOSERVERS) = AUTOMATIC(8)
Index sort flag (INDEXSORT) = YES
Sequential detect flag (SEQDETECT) = YES
Default prefetch size (pages) (DFT_PREFETCH_SZ) = AUTOMATIC

Track modified pages (TRACKMOD) = ON

Default number of containers = 1
Default tablespace extentsize (pages) (DFT_EXTENT_SZ) = 2

Max number of active applications (MAXAPPLS) = AUTOMATIC(117)
Average number of active applications (AVG_APPLS) = AUTOMATIC(3)
Max DB files open per application (MAXFILOP) = 61440

Log file size (4KB) (LOGFILSIZE) = 163800
Number of primary log files (LOGPRIMARY) = 60
Number of secondary log files (LOGSECOND) = 80
Changed path to log files (NEWLOGPATH) =
Path to log files = /SAN_disks/db2_N0_disk1/OS5/log_dir/NODE0001/

Overflow log path (OVERFLOWLOGPATH) =
Mirror log path (MIRRORLOGPATH) =
First active log file =
Block log on disk full (BLK_LOG_DSK_FUL) = YES
Block non logged operations (BLOCKNONLOGGED) = NO
Percent max primary log space by transaction (MAX_LOG) = 0
Num. of active log files for 1 active UOW (NUM_LOG_SPAN) = 0

Group commit count (MINCOMMIT) = 1
Percent log file reclaimed before soft chkpt (SOFTMAX) = 50
Log retain for recovery enabled (LOGRETAIEN) = OFF
User exit for logging enabled (USEREXIT) = OFF

HADR database role = STANDARD
HADR local host name (HADR_LOCAL_HOST) =
HADR local service name (HADR_LOCAL_SVC) =
HADR remote host name (HADR_REMOTE_HOST) =
HADR remote service name (HADR_REMOTE_SVC) =
HADR instance name of remote server (HADR_REMOTE_INST) =
HADR timeout value (HADR_TIMEOUT) = 120
HADR log write synchronization mode (HADR_SYNCHMODE) = NEARSYNC
HADR peer window duration (seconds) (HADR_PEER_WINDOW) = 0

First log archive method (LOGARCHMETH1) = OFF
Options for logarchmeth1 (LOGARCHOPT1) =
Second log archive method (LOGARCHMETH2) = OFF
Options for logarchmeth2 (LOGARCHOPT2) =
<table>
<thead>
<tr>
<th>Configuration Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failover log archive path</td>
<td>FAILARCHPATH =</td>
</tr>
<tr>
<td>Number of log archive retries on error</td>
<td>NUMARCHRETTRY = 5</td>
</tr>
<tr>
<td>Log archive retry Delay (secs)</td>
<td>ARCHRETRYDELAY = 20</td>
</tr>
<tr>
<td>Vendor options</td>
<td>VENDOROPT =</td>
</tr>
<tr>
<td>Auto restart enabled</td>
<td>AUTOORESTART = ON</td>
</tr>
<tr>
<td>Index re-creation time and redo index build</td>
<td>INDEXREC = SYSTEM (RESTART)</td>
</tr>
<tr>
<td>Log pages during index build</td>
<td>LOGINDEXBUILD = OFF</td>
</tr>
<tr>
<td>Default number of loadrec sessions</td>
<td>DFLT_LOADREC_SES = 1</td>
</tr>
<tr>
<td>Number of database backups to retain</td>
<td>NUM_DB_BACKUPS = 12</td>
</tr>
<tr>
<td>Recovery history retention (days)</td>
<td>REC_HIS_RETENTN = 60</td>
</tr>
<tr>
<td>Auto deletion of recovery objects</td>
<td>AUTO_DEL_REC_OBJ = OFF</td>
</tr>
<tr>
<td>TSM management class</td>
<td>TSM_MGMTCLASS =</td>
</tr>
<tr>
<td>TSM node name</td>
<td>TSM_NODENAME =</td>
</tr>
<tr>
<td>TSM owner</td>
<td>TSM_OWNER =</td>
</tr>
<tr>
<td>TSM password</td>
<td>TSM_PASSWORD =</td>
</tr>
<tr>
<td>Automatic maintenance</td>
<td>AUTO_MAINT = ON</td>
</tr>
<tr>
<td>Automatic database backup</td>
<td>AUTO_DB_BACKUP = OFF</td>
</tr>
<tr>
<td>Automatic table maintenance</td>
<td>AUTO_TBL_MAINT = ON</td>
</tr>
<tr>
<td>Automatic runstats</td>
<td>AUTO_RUNSTATS = ON</td>
</tr>
<tr>
<td>Automatic statement statistics</td>
<td>AUTO_STMT_STATS = ON</td>
</tr>
<tr>
<td>Automatic statistics profiling</td>
<td>AUTO_STATS_PROF = OFF</td>
</tr>
<tr>
<td>Automatic profile updates</td>
<td>AUTO_PROF_UPD = OFF</td>
</tr>
<tr>
<td>Automatic reorganization</td>
<td>AUTO_REORG = OFF</td>
</tr>
<tr>
<td>Auto-Revalidation</td>
<td>AUTO_REVAL = DEFERRED</td>
</tr>
<tr>
<td>Currently Committed</td>
<td>CUR_COMMIT = ON</td>
</tr>
<tr>
<td>CHAR output with DECIMAL input</td>
<td>DEC_TO_CHAR_FMT = NEW</td>
</tr>
<tr>
<td>Enable XML Character operations</td>
<td>ENABLE_XMLCHAR = YES</td>
</tr>
<tr>
<td>WLM Collection Interval (minutes)</td>
<td>WLM_COLLECT_INT = 0</td>
</tr>
<tr>
<td>Monitor Collect Settings</td>
<td></td>
</tr>
<tr>
<td>Request metrics</td>
<td>MON_REQ_METRICS = BASE</td>
</tr>
<tr>
<td>Activity metrics</td>
<td>MON_ACT_METRICS = NONE</td>
</tr>
<tr>
<td>Object metrics</td>
<td>MON_OBJ_METRICS = BASE</td>
</tr>
<tr>
<td>Unit of work events</td>
<td>MON_UOW_DATA = NONE</td>
</tr>
<tr>
<td>Lock timeout events</td>
<td>MON_LOCKTIMEOUT = WITHOUT_HIST</td>
</tr>
<tr>
<td>Deadlock events</td>
<td>MON_DEADLOCK = WITHOUT_HIST</td>
</tr>
<tr>
<td>Lock wait events</td>
<td>MON_LOCKWAIT = NONE</td>
</tr>
<tr>
<td>Lock wait event threshold</td>
<td>MON_LW_THRESH = 5000000</td>
</tr>
<tr>
<td>Number of package list entries</td>
<td>MON_PKGLIST_SZ = 32</td>
</tr>
<tr>
<td>Lock event notification level</td>
<td>MON_LCK_MSG_LVL = 1</td>
</tr>
<tr>
<td>SMTP Server</td>
<td>SMTP_SERVER =</td>
</tr>
<tr>
<td>SQL conditional compilation flags</td>
<td>SQL_CCFLAGS =</td>
</tr>
<tr>
<td>Section actuals setting</td>
<td>SECTION_ACTUALS = NONE</td>
</tr>
</tbody>
</table>

**Bufferpools (KB)**

Partition 0: 7 GB  
Partitions 1 - 7: 4 GB
Table of Figures

Figure 1: Layered Scalable Architecture (LSA) for Point of Sales Data Management........ 8
Figure 2: Standard SAP NetWeaver BW Database Layout on DB2 for LUW.................. 13
Figure 3: Multidimensional Clustering ........................................................................ 14
Figure 4: Multidimensional Clustering – Block Indexes ................................................ 14
Figure 5: Basic Setup of NLS with DB2 for LUW as NLS Database............................ 17
Figure 6: Active Memory Expansion............................................................................ 20
Figure 7: System Overview ......................................................................................... 21
Figure 8: SAP System and DB2 Database Layout ....................................................... 23
Figure 9: Process Chain for Data Upload from PIPE.................................................. 24
Figure 10: Process Chain for Data Load into the InfoProviders of the LSA for POS DM... 25
Figure 11: Results of Data Load Tests .......................................................................... 27
Figure 12: CPU Usage of SAP Application Server LPAR ............................................ 28
Figure 13: CPU Usage of Database Server LPAR ....................................................... 28
Figure 14: CPU Usage during Load of a single InfoPackage ....................................... 30
Figure 15: SAP NetWeaver BW Query Runtimes Accessing BW InfoCube versus NLS InfoCube................................................................. 34
Figure 16: Results of POWER7 AME Tests for the SAP Application Server LPAR..... 36
Figure 17: Sample Output of amepat .......................................................................... 38

List of Tables

Table 1: CPU and Memory Resources for Database and Application Server LPAR ....... 26
Table 2: Results of the Data Load Tests ........................................................................ 27
Table 3: CPU Usage during Data Load ......................................................................... 29
Table 4: CPU Usage during Load of one InfoPackage ................................................ 31
Table 5: System Configuration used during NLS Tests .............................................. 32
Table 6: System Performance Characteristics during NLS Tests ................................ 32
Table 7: Deletion Performance .................................................................................... 33
Table 8: Results of POWER7 AME Tests for the SAP Application Server LPAR ....... 36
Related Content

Enterprise Data Warehouse Service for POS,
http://wiki.sdn.sap.com/wiki/display/Retail/Enterprise+Data+Warehouse+Service+for+POS

SAP NetWeaver BW – What’s New With SAP NetWeaver BW 7.3,

SAP NetWeaver Business Warehouse 7.3 – Administration Tasks: IBM DB2 for Linux, UNIX, and Windows,
https://websmp101.sap-ag.de/~sapidb/011000358700001420572010E.PDF

Enabling SAP NetWeaver Business Warehouse System to Use IBM DB2 for Linux, UNIX, and Windows as Near-Line Storage (NLS),
https://websmp101.sap-ag.de/~sapidb/011000358700001299002010E.PDF

FAQ: Along with the POWER7 processor comes some new terminology such as Intelligent Threads, TurboCore, Active Memory Expansion. What do these new terms mean?,
http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/FQ128785

Understanding processor utilization on POWER systems – AIX,

Active Memory Expansion: Overview and Usage Guide,

Active Memory Expansion Performance,