PUBLIC
SAP Data Services
Document Version: 4.3 (14.3.00.00) – 2024-03-08

Sizing Guide

THE BEST RUN SAP

Content

1 About this guide

Decide your hardware and system needs to support installation and operation of SAP Data Services and supporting software.

The concept of installing software is simple. However, based on how you plan to use the software, and the resources required to run the software, installation can be complex.

To achieve a successful installation, plan ahead. Part of planning ahead includes deciding your system and hardware needs so that Data Services and supporting software fits comfortably on your system and supports current and future operations. We refer to this process as sizing.

The methodology and scenarios that we offer demonstrate the thought process and tasks involved in sizing. The performance and functioning of an actual system may vary for many reasons. You should not consider the examples that we offer here as a guarantee of success of a particular deployment.

Our goal in this guide is to provide performance numbers using real-world scenarios instead of presenting record-breaking performance numbers in an isolated and unrealistic test environment. This approach enables you to compare different approaches and results to your environment. Use the estimates from this guide as the foundation for your sizing study.

We recommend that you work with your SAP account representative to ensure that you get the advice of a sizing expert when you make final decisions about your Data Services deployment.

2 About SAP Data Services

SAP Data Services delivers a single enterprise-class solution for data integration, data quality, data profiling and text data processing.

Data Services enables you to integrate, transform, improve, and deliver trusted data to critical business processes. It provides one development UI, metadata repository, data connectivity layer, run-time environment, and management console – enabling IT organizations to lower total cost of ownership and accelerate time to value.

With Data Services, IT organizations can maximize operational efficiency with a single solution to improve data quality and gain access to heterogeneous sources and applications.

2.1 SAP Data Integration

Data integration features in SAP Data Services enable you to integrate data from differing sources.

Data integration features provide the following benefits:

- Easy-to-configure transforms for typically complex tasks like slow-changing dimensions and hierarchy flattening.
- Ability to build large jobs that include error handling, dependency handling, and job restarting.
- Generate extensive operational statistics.
- Connectivity to many types of sources and targets, most that use the vendor native format for maximum performance.
- Easy-to-use parallelization and performance optimization options.
- Functionalities to simplify daily operations and project hand-over, like web-based management console, auto-documentation features, and impact lineage information.

Parent topic: About SAP Data Services [page 4]

Related Information

SAP Text Data Processing [page 5]
SAP Data Quality [page 6]

2.2 SAP Text Data Processing

Text data processing in SAP Data Services enables you to analyze text and identify patterns, word types, entity relationships, sentiment, and so on.

Text data processing provides the following benefits:

- Analyzes text to identify and extract entities like people, dates, places, and organizations in multiple languages.
- Identifies patterns, activities, events, and relationships among entities, and enables their extraction.
- Goes beyond conventional character matching tools for information retrieval and understands word semantics.
- Supports extraction in 31 languages.
- Supports binary document formats such as PDF and Microsoft Word, in addition to text, HTML, and XML.
- Supports user-specified entity lists in a custom dictionary, which enables the storage of entities, management of name variations, and standardization of known entity names using a dictionary.
- Supports user-written rules to customize extraction output, and provides predefined rules that support sentiment analysis, enterprises, and the public sector.
- Supports broad, heterogeneous, applications and systems for SAP and non-SAP sources and targets
- Provides high performance and scalability, and meets high volume needs through parallel processing, grid computing, and bulk loading support.

Parent topic: About SAP Data Services [page 4]

Related Information

SAP Data Integration [page 4]
SAP Data Quality [page 6]

2.3 SAP Data Quality

Data quality features in SAP Data Services enable you to correct, standardize, and eliminate duplicates from your data.

Data quality features provide the following benefits:

- Provides dashboards that show the impact of data quality problems on all downstream systems and applications.
- Applies data quality transformations to all types of data, regardless of industry or data domain, such as structured to unstructured data as well as customer, product, supplier, and material information.
- Provides intuitive business-user interfaces and data quality blueprints that aid in the process of standardizing, correcting, and matching data to reduce duplicates and identify relationships.
- Provides comprehensive global data quality coverage with support for over 230 countries.
- Provides comprehensive reference data for multiple countries.
- Supports broad, heterogeneous applications and systems for both SAP and non-SAP sources and targets.
- Provides prepackaged native integration of data quality best practices for SAP environments.
- Optimizes developer productivity and application maintenance through intuitive transformations, a centralized business rule repository, and object reuse.
- Provides high performance and scalability with software that meets large data volumes through parallel processing, grid computing, and bulk loading.
- Provides flexible options for technology deployment, such as an enterprise platform and intuitive APIs that enable developers to quickly create data quality deployments and functionality.

Parent topic: About SAP Data Services [page 4]

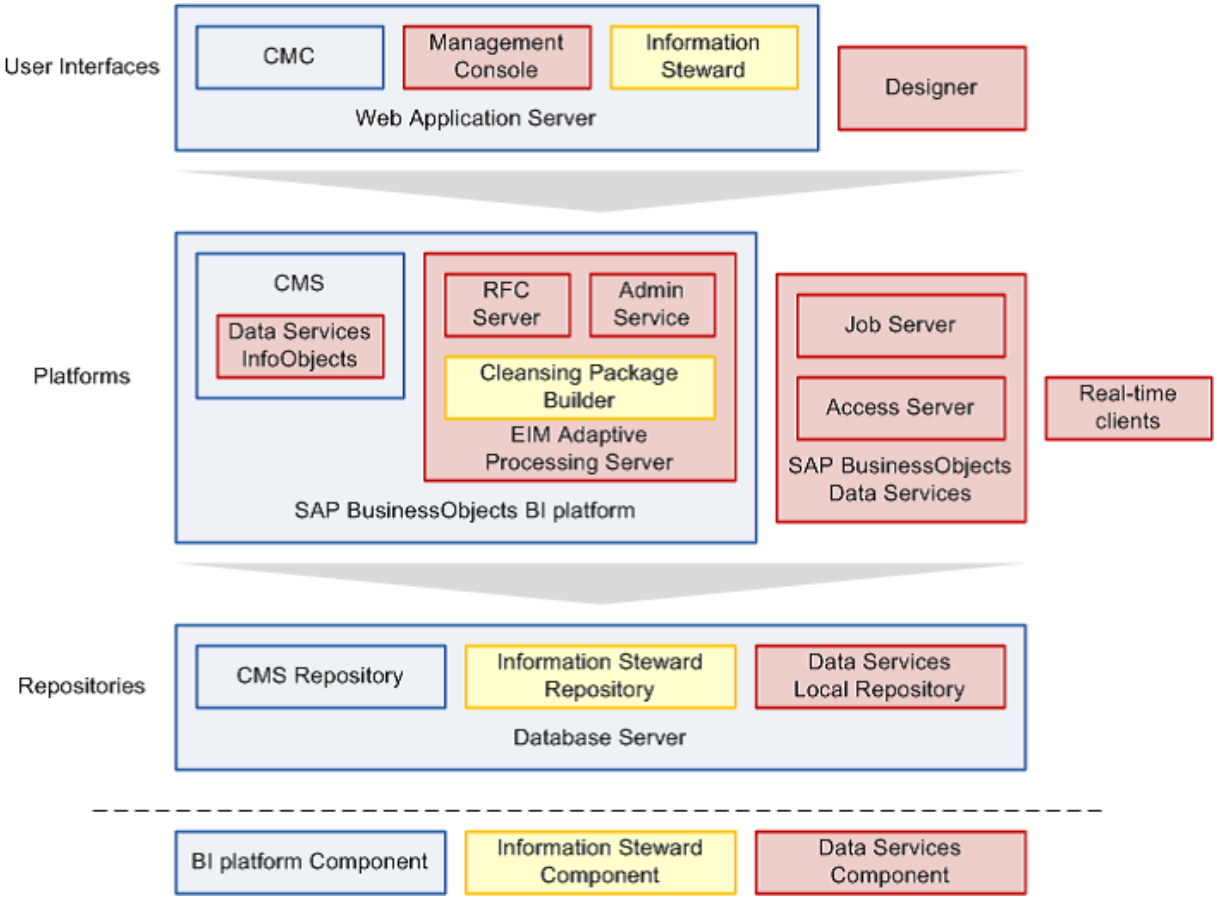
Related Information

SAP Data Integration [page 4]
SAP Text Data Processing [page 5]

3 SAP Data Services architecture

SAP Data Services architecture encompasses web-based user interfaces, platforms, and repositories to provide a complete solution.

The following diagram illustrates how SAP Data Services components fit in with other software in the SAP portfolio.



For more details about Data Services architecture, see the *Administrator Guide*.

4 Factors that influence performance

Improve the performance of SAP Data Services on your system by using optimization techniques such as caching, parallel processing, push-down operations, and grid computing.

The following table describes some of the factors that can influence processing performance in Data Services. For complete information about enhancing performance, see the *Performance Optimization Guide*.

Performance factors

Factor	Information
Access to source and targets	Your system bandwidth to the source and target can affect how fast Data Services passes data through the data flow.
Availability of additional RAM	Allocating enough free RAM within the system for caching improves data flow performance. Free RAM enables Data Services to cache lookup data and reference data for data quality transforms.
Configuration and system landscape	When your system landscape has Data Services installed on the target database system and the source RDBMS on a separate machine, a job can run multiple resource-intensive processes. However, understanding your hardware and system needs and changing your configuration can reduce the competition for resources and enhance job throughput.
Degree of Parallelization (DOP)	Running multiple processes in parallel positively influences performance when your system uses the appropriate hardware; increasing the DOP setting generally increases throughput.
Grid computing	Using grid computing on multiple servers can process different elements of the same job in parallel. Creating a server group of several job servers along with Data Services load-balancing algorithm ensures that the server with the lowest workload executes the processes that are spawned by the job.
Parallel file reader	The *Parallel Process Threads* option enables you to specify the number of parallel threads. Data Services parallel file reader uses the setting to convert text strings from flat files to columns. Data Services uses the schema defined in the file format for column format.

Data integration performance [page 9]
Improve the performance of data integration by implementing some basic processing enhancements.

Data quality performance [page 9]
Data quality processes can slow performance because they use system resources for complex processes and for searching reference files.

Text data processing performance [page 10]
The characteristics of a source document, including the format, length, and density, can impact the performance of text data processing.

4.1 Data integration performance

Improve the performance of data integration by implementing some basic processing enhancements.

The following table contains some basic ideas for improving the data integration process in SAP Data Services.

Data integration performance factors

Factor	Information
Loader method	Each loading method has pros and cons based on database vendor implementation. Depending on the database type and version, the choice of loading method can make dramatic differences in performance. Loading methods include regular load, bulk load, and auto correct load.
Transactional loaders	Loading data in one transaction means that the dataflow cannot use parallel processes to speed up the loading.
Lookup and join settings	Consider the amount of data to process versus the size of the lookup table when you choose to use a lookup table for implementing joins. Data Services enables you to choose the lookup strategy; choose carefully.
Heterogeneous sources or all in one database	If all data is in one database or a database link exists between the databases, the Data Services optimizer has more options to determine whether to push down parts or all processing to the database.

Parent topic: Factors that influence performance [page 8]

Related Information

Data quality performance [page 9]
Text data processing performance [page 10]

4.2 Data quality performance

Data quality processes can slow performance because they use system resources for complex processes and for searching reference files.

The following factors can influence the performance of data quality processes:

- **Complexity of processing:** A data quality transform can do varying degrees of simple or complex processing based on the options set in the transform. Generally, more complex processing such as matching requires more hardware resources.

- **Location of reference data:** Several of the data cleanse transforms use reference data located on your file system, and they can be input—output (I/O) or network dependent. The speed with which Data Services access reference data affects transform performance.

Parent topic: Factors that influence performance [page 8]

Related Information

Data integration performance [page 9]
Text data processing performance [page 10]

4.3 Text data processing performance

The characteristics of a source document, including the format, length, and density, can impact the performance of text data processing.

The following list contains performance considerations for text data processing:

- **Format:** XML and HTML require detagging before processing the text. Detagging has more overhead than processing text directly. Additionally, converting a binary document into a textual representation during processing has overhead.
- **Length:** Longer input documents require more processing time.
- **Density:** More density and fact-rich input documents require more processing time.
- **Rules:** Using one or more rules to customize extraction can require more processing time.

Parent topic: Factors that influence performance [page 8]

Related Information

Data integration performance [page 9]
Data quality performance [page 9]

5 Server hardware and system landscape

The server hardware that you use, and your system landscape have a direct effect on system performance.

To generate the results for the scenarios in this guide, we used the system landscape that is included in the following tables.

The following table contains the hardware and versions for the scenarios.

Hardware

Hardware	Version
MODE	Z820 CPU: Intel(R) Xeon(R) CPU E5-2670 v2 @ 2.50 GHz
OS	SUSE Linux Enterprise Server 11 (x86_64)
DISK	SSD 1.3 TB
MEMORY	RAM 512 GB
CPU CORE	40

The following table contains the software and versions for the scenarios.

Software and Version

Software	Version
SAP BusinessObjects Enterprise (BOE) platform	4.1 SP4
SAP Data Services	4.2 SP4

The following table contains the relational database management systems (RDBMS) and versions for the scenario.

RDBMS

Name	Version
SAP HANA	1.0 Rev 90
Oracle	11G
SAP ERP	7.0

Configuration

Data Services software, source RDBMS, and target RDBMS are installed on the same server to simulate a 10G network environment.

6 Batch process sizing

We tested SAP Data Services batch jobs using a variety of situations including various source and target data and different database management systems.

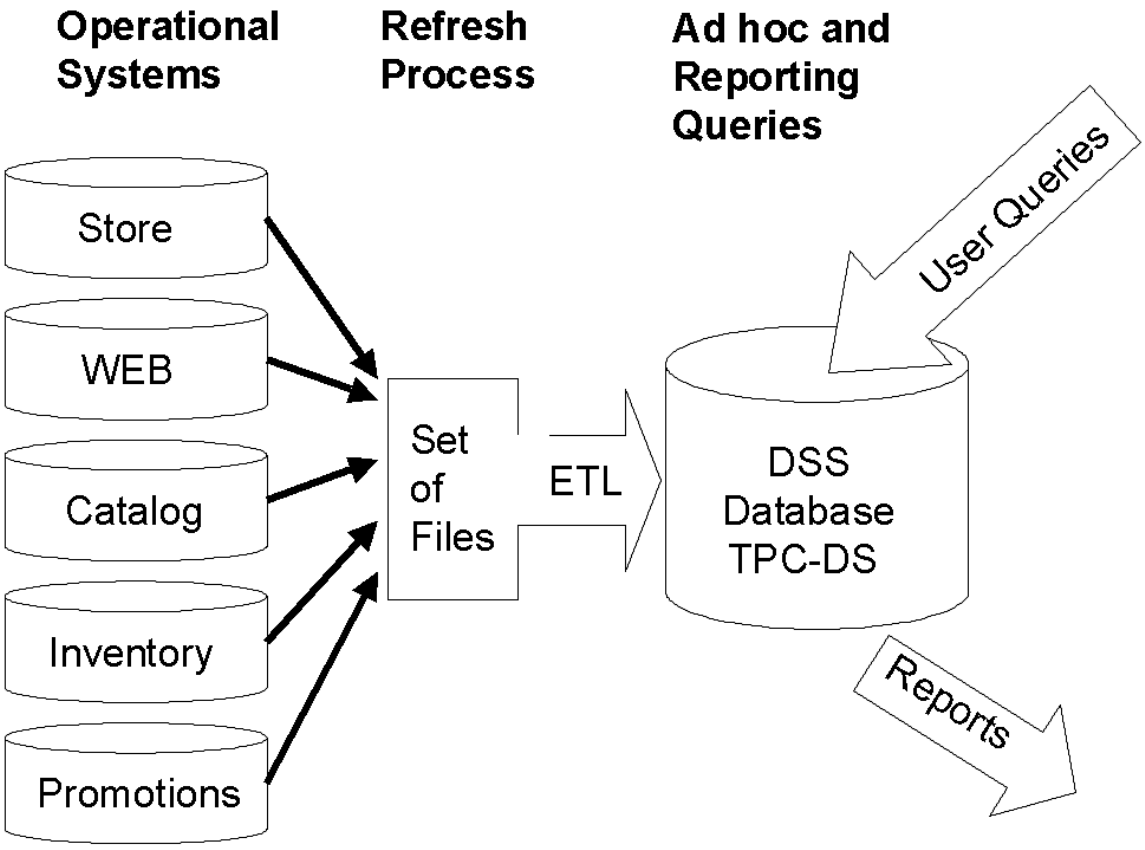
Test concept and use cases

One of the primary use cases for SAP Data Services is to move data from one application to another. Migration projects are relevant for all types of companies, including companies with SAP applications and non-SAP applications. Migration projects are also relevant for upgrading. The scenarios include a batch data flow that loads a data warehouse from a transactional system. However, you can apply most use cases to data migration situations.

Data Sets

For the scenario source data, we use the TPC Benchmark DS (TCP-DS) data model. TCP-DS represents a realistic enterprise resource planning (ERP) source system. For more information about TCP, see www.tcp.org.

The representative data that we use is generated automatically for all tables.



The following figure shows the resulting representative data.

CUSTOMER(hana_ds.PERF_USER)	
C_CUSTOMER_SK	int
C_CUSTOMER_ID	varchar(16)
C_CURRENT_CDEMO_SK	int
C_CURRENT_HDEMO_SK	int
C_CURRENT_ADDR_SK	int
C_FIRST_SHIPTO_DATE_SK	int
C_FIRST_SALES_DATE_SK	int
C_SALUTATION	varchar(10)
C_FIRST_NAME	varchar(20)
C_LAST_NAME	varchar(30)
C_PREFERRED_CUST_FLAG	varchar(1)
C_BIRTH_DAY	int
C_BIRTH_MONTH	int
C_BIRTH_YEAR	int
C_BIRTH_COUNTRY	varchar(20)
C_LOGIN	varchar(13)
C_EMAIL_ADDRESS	varchar(50)
C_LAST_REVIEW_DATE	varchar(10)

6.1 Scenario 1: Loading data from Oracle to SAP HANA

Increase the number of parallel file threads and loaders to improve overall performance.

In scenario 1, the SAP Data Services batch job reads data from an Oracle database and loads generated data to an SAP HANA database. We processed the data using eight loaders and eight table partitions. The following table shows the processing statistics.

# of Rows	Size of Data	Throughput (GB/ hour)	CPUs	Memory
2 Million	0.25 GB	149.49 GB	8 cores	4 GB
10 Million	~1.38 GB	187.67 GB	8 cores	4 GB
50 Million	~6.92 GB	188.45 GB	8 cores	4 GB
100 Million	~ 13.68 GB	160.19 GB	8 cores	4 GB
500 Million	~ 68.3 GB	148.91 GB	8 cores	4 GB

Evaluation:

There is a bottleneck in the database that is preventing the reader or loader from using the Job Server CPUs efficiently. Before you tune the system, enhance throughput from the database side. For example, add more disks for faster input/output.

In general, we can improve the overall performance throughput in this scenario by increasing the number of parallel file threads and loaders.

Parent topic: Batch process sizing [page 12]

Related Information

Scenario 2: Loading data from flat file to SAP HANA [page 15]
Scenario 3: loading dimension table from Oracle to SAP HANA [page 16]
Scenario 4: Loading initial and delta with table comparison [page 17]
Scenario 5: Joining tables to load a fact table [page 18]

6.2 Scenario 2: Loading data from flat file to SAP HANA

Increase file threads and loaders to enhance performance when a data flow loads customer data from a flat file to an SAP HANA database.

A batch data flow reads customer data from a flat file. The data flow processes the data and loads generated data to SAP HANA. The following table contains statistics when we execute the job using eight loaders.

# of Rows	Size of Data	Throughput (GB/hour)	CPUs	Memory
2 Million	0.25 GB	154.64 GB	8 cores	4 GB
10 Million	~1.38 GB	276.86 GB	16 cores	4 GB
50 Million	~6.92 GB	296.25 GB	16 cores	4 GB
100 Million	~ 13.68 GB	278.58 GB	16 cores	4 GB
500 Million	~ 68.3 GB	282.09 GB	16 cores	4 GB

After the job completed execution, we increased the number of parallel threads and loaders to see we can improve the overall performance throughputs. The following table shows the processing statistics.

# of File Threads	# of Loaders	# of Rows	Size of Data	Throughput (GB/hour)	CPUs	Memory
4	4	100 Million	12.4 GB	**171.19 GB**	8 cores	**4 GB**
8	8	100 Million	12.4 GB	278.58 GB	16 cores	8 GB
16	16	100 Million	12.4 GB	**315.86 GB**	20 cores	**12 GB**
4	4	500 Million	62.3 GB	162.39 GB	8 cores	4 GB
8	8	500 Million	62.3 GB	282.09 GB	16 cores	8 GB
16	16	500 Million	62.3 GB	355.97 GB	28 cores	12 GB

Evaluation:

- **# of File threads = 4** results in **Throughput (GB/hour) = 171.19 GB**
- **# of File threads = 16** results in **Throughput (GB/hour) = 315.86 GB**
- Memory increased from 4 GB to 12 GB when we increase the number of threads from 4 to 16. The memory increased because the more loaders you use, the more memory Data Services uses.

Parent topic: Batch process sizing [page 12]

Related Information

Scenario 1: Loading data from Oracle to SAP HANA [page 14]
Scenario 3: loading dimension table from Oracle to SAP HANA [page 16]
Scenario 4: Loading initial and delta with table comparison [page 17]
Scenario 5: Joining tables to load a fact table [page 18]

6.3 Scenario 3: loading dimension table from Oracle to SAP HANA

See how a validation transform affects performance when we load a customer dimension table from Oracle to SAP HANA.

In this scenario, we define three business rules:

- Customer first name can't be NULL
- Customer last name can't be NULL
- E-mail address is in a format that contains a local part, an @ symbol, and a domain. For example, john.smith@ example.com

To get an idea about how the validation transform affects performance, we execute the following tests:

1. In test 1, we load the customer dimension table directly from the Oracle source database to the HANA target database.
2. In test 2, we loaded the same customer dimension table but included a validation transform that validated the three business rules before loading the data into the target.

Operations	Execution time (sec)	Throughput (GB/hour)	CPUs	Memory
Test 1: No validation during loading	22	156 GB	2 cores	4 GB
Test 2: Validation during loading	32	108 GB	2 cores	4 GB

Evaluation: Even though Test 2 included the validation transform, which produced a more complex data load and took slightly longer to execute, SAP Data Services still performed 108 GB/hour and didn't introduce a significant bottleneck.

Parent topic: Batch process sizing [page 12]

Related Information

Scenario 1: Loading data from Oracle to SAP HANA [page 14]
Scenario 2: Loading data from flat file to SAP HANA [page 15]
Scenario 4: Loading initial and delta with table comparison [page 17]
Scenario 5: Joining tables to load a fact table [page 18]

6.4 Scenario 4: Loading initial and delta with table comparison

In this scenario, we find out which tuning technique is more efficient based on throughput.

We configure a batch data flow to use CDC functionality from a supporting RDBMS for processing delta loads. If your source database doesn't support CDC, SAP Data Services uses a built-in Table Comparison transform, which detects changes by comparing the source with the target table.

For each job execution, loading progressed from initial load to delta loads. Before each delta load, we implemented one of the following tuning techniques:

- **Row-by-row select:** Selects each row from the source, and calls a query on the target to verify if the row is new, existing, or changed. Row-by-row select is usually the slowest method.
- **Cache comparison table:** Reads columns into memory and compares columns from the target table in memory time. This technique is fast for smaller tables, but can be slower for larger tables (> 2 GB) because paging can occur. If the memory-consuming operations in your data flow exceed the available memory, use pageable cache to complete the operation.
- **Sorted input:** Sorts input data in both the source and the target tables.

For more information about these tuning techniques, see the *Performance Optimization Guide*.

The following table shows the processing statistics beginning with the initial load. Notice the change to **Throughput (GB/hour)** for each delta load.

Load	Parameters Tuned	Execution Time (Seconds)	Throughput (GB/hour)	CPUs	Memory
Initial load		22	156 GB	2 cores	4 GB

Load	Parameters Tuned	Execution Time (Seconds)	Throughput (GB/hour)	CPUs	Memory
Delta load	row-by-row DOP = 2	248	14 GB	2 cores	4 GB
Delta load	row-by-row DOP = 4	118	30 GB	2 cores	4 GB
Delta load	Cache DOP = 4	40	86 GB	2 cores	4 GB
Delta load	Sorted input DOP = 2 or 4	25	137 GB	2 cores	4 GB

Evaluation:

In our 2,000,000 record customer table, we achieved a throughput of 14 GB/hour for the delta load using the row-by-row comparison. Based on execution time, the cache comparison table and the sorted input were the more effective tuning techniques.

- Cache comparison table: 86 GB/hour
- Sorted input: 137 GB/hour

Parent topic: Batch process sizing [page 12]

Related Information

Scenario 1: Loading data from Oracle to SAP HANA [page 14]
Scenario 2: Loading data from flat file to SAP HANA [page 15]
Scenario 3: loading dimension table from Oracle to SAP HANA [page 16]
Scenario 5: Joining tables to load a fact table [page 18]

6.5 Scenario 5: Joining tables to load a fact table

Pushing a join operation to the database server optimizes processing speed in SAP Data Services batch jobs.

In a typical data warehouse project, a retailer pulls a sales report for a specific store during a specified period of time. The retailer retrieves information from the fact tables that relate to the transactions that meet the specific criteria. For example, to accomplish the task, the retailer obtains the data from three tables:

- **Store_Sales**

- **Date**
- **Customer**

Further, the retailer configures the data flow to:

- Generate a data warehouse key for each new record.
- Load the joined fact table into the target database

The following table shows the number of rows in each dimension table, and the total table size in Gigabytes (GB).

Source Table	Store_Sales	Date	Customer
# of rows	287,997,204	73,049	2,000,000
Total Table Size (GB)	37.88	0.009	0.249

SAP Data Services performs the join either in the database server, or in the Data Services engine. A common trend is to push down as much processing as possible to the database server for performance optimization. Pushing down processes to the database server, however, isn't always possible.

In the first table, all dimension tables are from the same database type. Therefore, SAP Data Services configures the data flow to push the join process to the source database server.

Push down to source database

# of Rows	**Size of Data**	**Job Completion**	**CPUs**	**Memory**
5,244,914	313.65 GB	90 seconds	2 cores	4 GB

However, not all tables are stored in the same database. If the tables are from various types of databases, the Data Services engine performs the join operation. The following table contains the processing statistics.

No push down possible

# of rows	**Size of data**	**Job Completion**	**CPUs**	**Memory**
5,244,914	313.65 GB	30 minutes	2 cores	4 GB

Evaluation: The difference between pushing down the join to the database server versus the join being performed in the Data Services engine is evident when you look at the **Job Completion** column:

- Push down to database server: 90 seconds
- Join processed in Data Services engine: 30 minutes

For complete information about implementing push down operations, see the *Performance Optimization Guide*.

Parent topic: Batch process sizing [page 12]

Related Information

Scenario 1: Loading data from Oracle to SAP HANA [page 14]

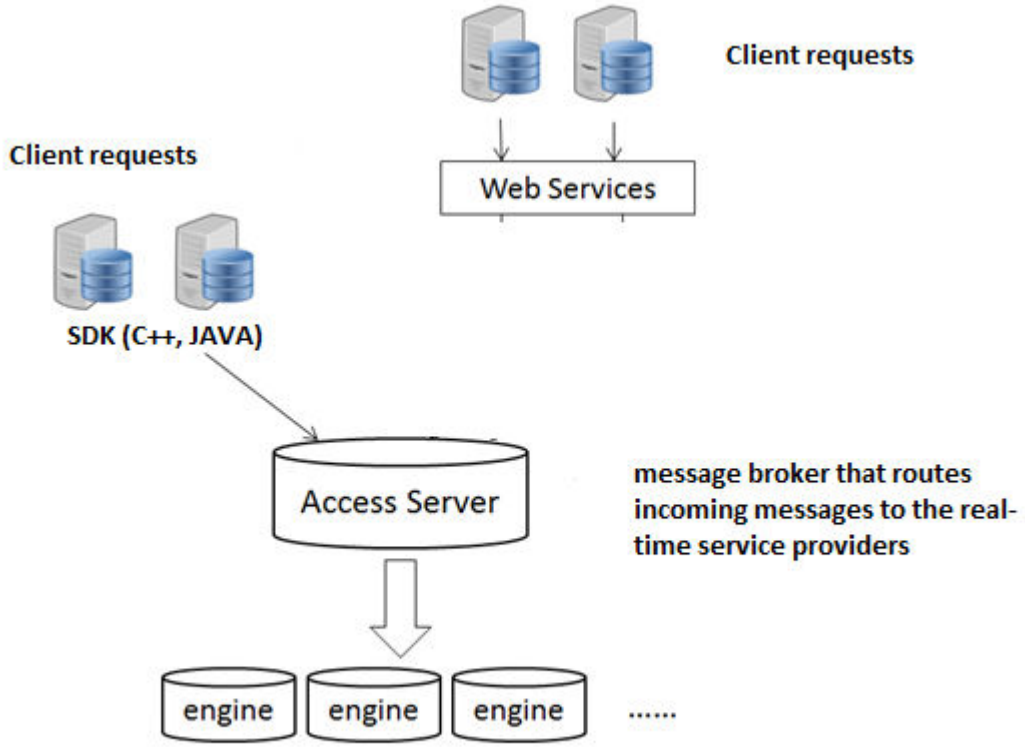
7 Real-Time process sizing

When sizing for transactional processing, it's important to consider response time requirements, estimated peak transactional throughput needs, and the number of potential concurrent client requests to support.

Transactional process

SAP Data Services provides Web Services, Java Messaging Service (JMS), and SDK library (C++ and Java APIs) to programmatically pass data to Data Services for real-time data movement, transformation, and retrieval. Real-time processing has slightly more overhead than batch processing because of the work required for processing client requests and distributing them to the job servers via the access servers.

The following figure illustrates a typical route that messages take starting with a client request.



After the message goes to the Data Services engine, it undergoes processing through a real-time data flow, and Data Services sends the message back as a response in a corrected format.

Concept for scenarios

A typical real-time message includes information that implements a business operation, for example, cleanse and standardize name and address input. It also requires an appropriate response, a corrected and standardized record.

> ### Example
>
> You have operators in a call center who receive customer data either online or over the phone. The system is integrated with Data Services through Web Services. Your system takes the incoming data and sends it to the Data Services engine through a web service. Data Services receives the message, runs it through a data flow that is designed to cleanse and standardize data, and the data flow outputs the generated reply to the message.

Before the process can work efficiently, you determine the hardware that you require and the system landscape, including the number of servers, amount of memory, and so on. To help you with this process, we present three scenarios: Simple, Moderate, and Complex.

Sizing calculation elements [page 22]
The elements to use when you calculate system and hardware size for transactional processes include engine process time, system resources, and hardware requirements.

Simple scenario: Cleansing US addresses [page 24]
Configure a real-time data flow to parse, standardize, and correct U.S. address data.

Moderate scenario: Cleansing world-wide addresses [page 25]
Configure an SAP Data Services real-time data flow to parse, standardize, and correct world-wide address data.

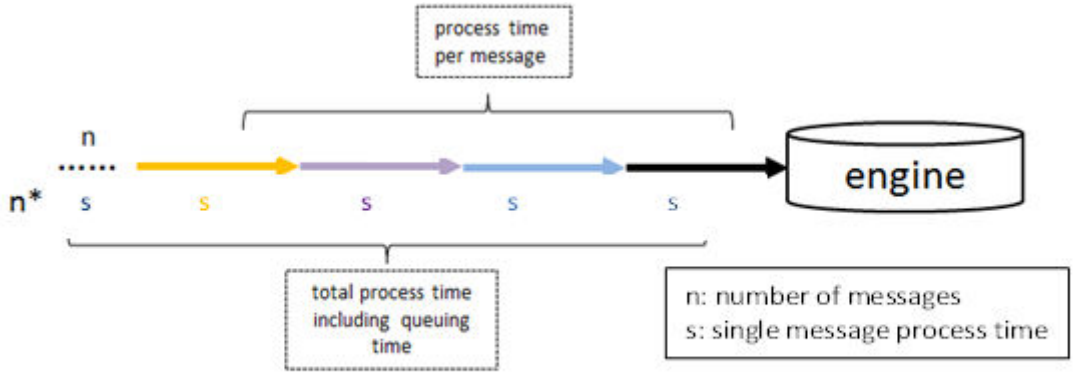
Complex scenario: identifying duplicates [page 26]
Configure a real-time data flow to identify duplicate records in addition to parsing, standardizing, and updating addresses.

7.1 Sizing calculation elements

The elements to use when you calculate system and hardware size for transactional processes include engine process time, system resources, and hardware requirements.

Follow these general guidelines when you calculate the system size for transactional processing:

1. Identify the engine processing time by sending a single message to Data Services. Check the maximum processing time. The queuing time should be zero.
   SAP Data Services uses a round-robin scheduling algorithm to split the incoming messages to each process engine in equal portions and in circular order. The following illustration shows the process with one engine.



2. Identify the number of required engines based on the concurrent message number. As the messages continue to grow, configure Data Services with additional engines for optimized processing time. Keep track of your average processing time and compare it to your desired average performance time.
3. For performance optimization, consider increasing the number of service providers as the number of concurrent messages increase.
4. Identify the minimum system resource requirement. Consider other processes including Job Server, Web server, engine processes recycling, and BOE/IPS Central Management Server (CMC) processes. Also allocate enough system resources for your Data Services installation. To find out the minimum hardware requirements to run Data Services products in one or multiple landscapes, see the Product Availability Matrix (PAM).
5. Plan for more system resources when you add an additional engine to your system.
   To calculate the total memory and CPU consumption, multiple the memory and CPU per engine by the number of engines configured. For example, if **m** = Number of engines:
   - **m*memory consumption per engine** = total RAM consumption
   - **m*CPU utilization per engine** = total CPU consumption

   Minimum system resources [page 24]
   Allocate enough system resources for your SAP Data Services installation when performing your sizing exercise.

Parent topic: Real-Time process sizing [page 21]

Related Information

Simple scenario: Cleansing US addresses [page 24]
Moderate scenario: Cleansing world-wide addresses [page 25]
Complex scenario: identifying duplicates [page 26]

7.1.1 Minimum system resources

Allocate enough system resources for your SAP Data Services installation when performing your sizing exercise.

By default, the size of an empty SAP Data Services and SAP Information Steward repository is insignificant. However, the size of the repository expands based on the increase of the number of objects within the repository. Each job and datastore that you add, plus the runtime statistics captured within statistic files such as `al_statistics`, `al_history`, increase the size of the repository. The number of objects stored, and the longer you retain the runtime statistics, the bigger the size of the repositories.

The following table provides a snapshot of the minimum disk space to install Data Services with Information Platform Services (IPS), which is a scaled-back version of SAP BusinessObjects Enterprise (BOE) platform, with an empty repository.

Disk space in empty DS/IS repository

Component	Disk space
SAP BusinessObjects folder, which includes applications such as: • IPS • Tomcat • Data Services • SQL Anywhere	~16-18 GB
Default installation with all languages	~23 GB
Empty Data Services (DS) Repository	~25 MB
Empty Information Steward (IS) Repository	~65 MB

Parent topic: Sizing calculation elements [page 22]

7.2 Simple scenario: Cleansing US addresses

Configure a real-time data flow to parse, standardize, and correct U.S. address data.

In this scenario, an incoming message contains information required to create a new customer account. The message content is simple. The message contains the customer name along with the address.

First, the process sends a message to the SAP Data Services real-time data flow. The data flow cleanses the customer data by parsing and standardizing the content, and correcting and updating the address. The data flow sends the results back to the client application.

The following table shows the processing statistics based on the number of concurrent client requests and the number of engines.

Number of Concurrent Client Requests	Number of Engines	Average Engine Processing Time (Milliseconds)	# of Messages Processed (Seconds)	Memory Consumption For DS Engines	Core
1	1	< 10 ms	103.55	1 GB	2
5	1	< 20 ms	383.05	1 GB	4
5	5	< 10 ms	607.90	5 GB	6
10	1	< 20 ms	741.93	1 GB	6
10	5	< 10 ms	1856.98	5 GB	10
50	5	< 30 ms	2176.17	5 GB	14
50	10	<20 ms	3188.93	10 GB	16

As a general rule, to minimize the queuing time for better performance, add more engines as the number of concurrent client requests increase.

> ### Example
>
> For the rows where the **Number Of Concurrent Client Requests** = 5, increasing the engines from 1 to 5 changes the **Average Engine Processing Time** from less than 20 milliseconds to less than 10 milliseconds. The **# Of Messages Processed (in seconds)** increases from 383.05 to 607.90 per second.

Parent topic: Real-Time process sizing [page 21]

Related Information

Sizing calculation elements [page 22]
Moderate scenario: Cleansing world-wide addresses [page 25]
Complex scenario: identifying duplicates [page 26]

7.3 Moderate scenario: Cleansing world-wide addresses

Configure an SAP Data Services real-time data flow to parse, standardize, and correct world-wide address data.

Similar to the simple scenario, this scenario also has an incoming message that contains information required to create a new customer account, but the addresses are from various countries. As with the simple example, the client application sends the message to a SAP Data Services data flow. The processes is a bit more involved, however, because Data Services uses reference data from multiple countries, and the internal processes are more complex.

The data flow cleanses the customer data by parsing and standardizing the content, and correcting and updating the address. The data flow sends the results back to the client application.

The following table shows the processing statistics based on the number of concurrent client requests and the number of engines.

Number of Concurrent Client Requests	Number of Engines	Avg Processing Time (Milliseconds)	# of Messages Processed (Seconds)	Memory Consumption For DS Engines	Core
1	1	< 10 ms	88.53	1 GB	2
5	1	< 20 ms	191.73	1 GB	4
5	5	< 10 ms	572.40	5 GB	6
10	1	< 40 ms	201.20	1 GB	6
10	5	< 10 ms	936.27	5 GB	10
50	5	< 30 ms	1148.93	5 GB	14
50	10	< 20 ms	1521.73	10 GB	16

Examine the differences in the **Average Engine Processing Time** and the **# of Messages Processed** columns when the **Number of Concurrent Processes** and **Number of Engines** change.

Parent topic: Real-Time process sizing [page 21]

Related Information

Sizing calculation elements [page 22]
Simple scenario: Cleansing US addresses [page 24]
Complex scenario: identifying duplicates [page 26]

7.4 Complex scenario: identifying duplicates

Configure a real-time data flow to identify duplicate records in addition to parsing, standardizing, and updating addresses.

To ensure your customer database doesn't contain duplicate records, add a Match transform to the SAP Data Services data flow. The Match transform checks that each incoming message contains a unique customer. If the customer is unique, Data Services adds the customer to the database. If the customer is not unique, Data Services finds the current record and updates it with any changed or additional data. You control all actions in the data flow configuration. If the data flow also has the U.S. or global data cleansing transforms, this scenario is more complex than the previous two scenarios.

The following table shows the processing statistics for processing messages with the complex real-time data flow. The statistics are based on the number of concurrent requests and the number of engines.

Number of Concurrent Client Requests	Number of Engines	Avg Processing Time (Milliseconds)	# of Messages Processed (Seconds)	Memory Consumption For DS Engines	Core
1	1	< 10 ms	75.60	1 GB	2
5	1	< 20 ms	201.45	1 GB	4
5	5	< 10 ms	544.05	5 GB	6
10	1	< 40 ms	194.55	1 GB	6
10	5	< 10 ms	1074.45	5 GB	10
50	5	< 30 ms	970.35	10 GB	16
50	10	< 20 ms	1340.25	10 GB	16

Based on the statistics, it seems that the processing overhead for handling requests in Data Services outweighs the variances in performance of the Data Quality transforms. Therefore, regardless of the complexity of the data flow configuration, the overall sizing requirements are similar.

Parent topic: Real-Time process sizing [page 21]

Related Information

Sizing calculation elements [page 22]
Simple scenario: Cleansing US addresses [page 24]
Moderate scenario: Cleansing world-wide addresses [page 25]

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.
About the icons:

- Links with the icon : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering an SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.
The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

www.sap.com/contactsap

THE BEST RUN SAP