Sizing Guide – RELEASED
Document Version: 1.3 – 2014-12-01

CUSTOMER

Sizing SAP NetWeaver Process Orchestration

Disclaimer

Some components of this product are based on Java™. Any code change in these components may cause unpredictable and severe malfunctions and is therefore expressly prohibited, as is any decompilation of these components.

Documentation in the SAP Service Marketplace

You can find this documentation at the following address: http://service.sap.com/sizing.

Typographic Conventions

Type Style	Description
Example	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Textual cross-references to other documents.
Example	Emphasized words or expressions.
`EXAMPLE`	Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE.
`Example`	Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
`Example`	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
`<Example>`	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
`EXAMPLE`	Keys on the keyboard, for example, `F2` or `ENTER`.

Document History

Version	Date	Change
1.0	2014-06-10	Released
1.1	2014-06-11	Minor corrections
1.2	2014-07-12	Minor corrections
1.3	2014-12-01	Updated

Table of Contents

1 Introduction

SAP NetWeaver® is the platform that provides the shared technology foundation for SAP business applications. In addition, SAP NetWeaver provides optional technology components that allow customers:

To extend the reach of SAP applications to more people, also via new devices and consumption models

To integrate SAP applications and processes into heterogeneous landscapes

To extend existing SAP processes

To compose new processes that span SAP and non-SAP applications

To securely manage and deliver information

To holistically manage the lifecycle and infrastructure of SAP applications

Based on the principles of a service-oriented architecture (SOA), SAP NetWeaver helps organizations optimize the return of their investment in SAP and non-SAP applications by allowing customers to evolve their current IT landscapes into flexible business process platforms that close the gap between insight and action.

SAP NetWeaver 7.4 is a robust and lean standard-based platform that enables you to develop Java and composite applications from scratch and on top of existing services. With the activated usage types of AS Java, Composition Environment Platform, Business Process Management, and Business Rules Management you can realize development projects that correspond to the use case Building Composite Applications of SAP NetWeaver. The activated usage type of Process Integration - Advanced Adapter Engine Extended facilitates the integration of business processes that span different departments, organizations, or companies. An integration scenario is used to model the process flow and the separation of a business application into its application components. The focus of SAP NetWeaver PI AEX is not on the “inner-life” of the individual application components, or how the business logic is implemented within an application component, but rather on how the application components exchange data with each other.

The SAP NetWeaver 7.4 Installation and Documentation Information Center is the central starting point for the technical implementation of SAP NetWeaver® 7.4. It provides you with information about the use cases for SAP NetWeaver 7.4 as well as the installable software units, and refers you to the detailed documentation (such as the Installation Guides).

Composite applications use a service-oriented architecture to fill the gaps that standard solutions cannot cover. They sit on top of existing enterprise solutions and reuse their functionality to shape and add new innovative business processes. Composite applications are based on services that are already provided by other applications or components. They combine available service operations with new application logic, user interfaces, and business process orchestration. These services can be provided by SAP's Business Process Platform (BPP) with SAP Business Suite, as well as by service-enabled third-party solutions.

Process Orchestration (SAP Help Documentation) provides a tool infrastructure to model and design business processes - from the overall process flow and the sequence of activities up to the interfaces and data types that are needed to integrate SAP systems and external systems.

It combines the integration capabilities of SAP NetWeaver Process Integration Advanced Adapter Engine Extended (AEX) with the process modeling, operation, and monitoring capabilities of SAP NetWeaver Business Process Management (BPM).

Note Like AEX, Process Orchestration is based on AS Java only.

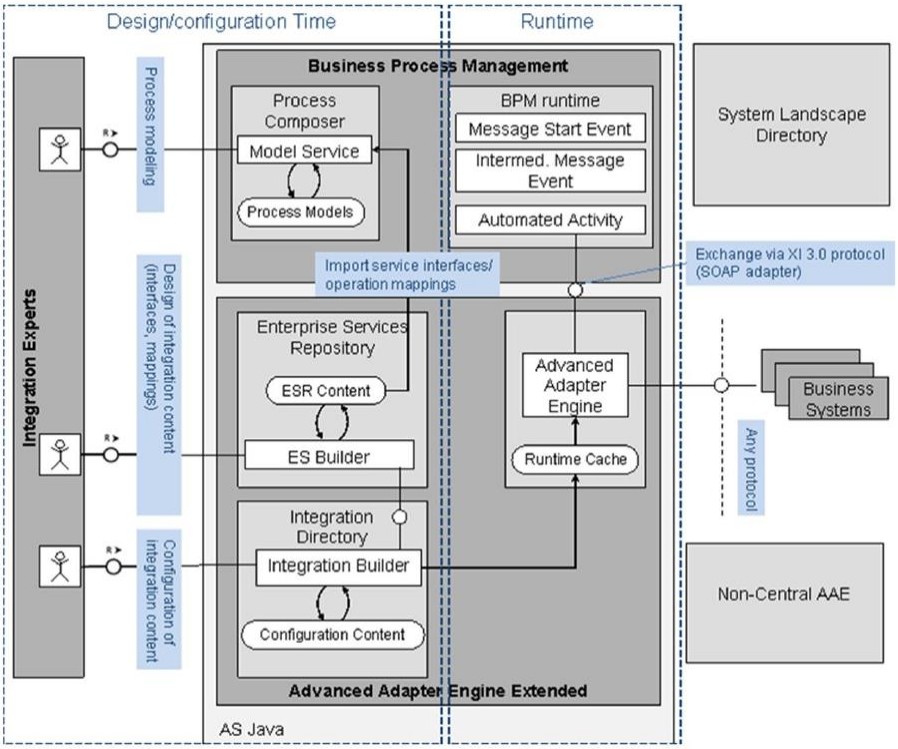


Figure 1 Components of Process Orchestration

Constraints

- Process Orchestration runs on one system. The deployment on more than one system is not supported
- Process Orchestration can be installed in a cluster. Every node would have an AEX and a BPM instance running

The package is based on the following components:

Advanced Adapter Engine Extended

AEX provides the connectivity capabilities of the Advanced Adapter Engine (AAE) as well as design and configuration tools (ES Repository and the Integration Directory) to set up integration scenarios.

More information: Advanced Adapter Engine Extended

Business Process Management and Business Rules Management

BPM and Business Rules Management allow you to design, execute, and monitor business processes.

More information: Business Process Management

You can obtain Process Orchestration by installation or by adding the corresponding usage types to an existing SAP NetWeaver system (as of release 7.4). For the latter, both are supported, deployment of BPM on an existing AEX system and deployment of an AEX on an existing BPM system.

For more information about features and constraints for the AEX and BPM runtime integration, see AEX and BPM Runtime Integration

1.1 Functions of Business Process Management

SAP NetWeaver Business Process Management (BPM) lets your business and IT professionals jointly compose executable processes using standardized notation.

BPM helps you rapidly tailor your business processes to changing business needs by enabling the following:

- Joint modeling of processes by business and IT specialists
- Central process execution via a Java-based process engine
- Provision of intuitive interfaces for business users
- Integration of business rules into your processes

Support your business process improvement initiatives with these features and functions of SAP NetWeaver Business Process Management:

Collaboratively model your processes – Provide a shared environment for co-designing new processes and adapting existing ones – based on the user-friendly business process model and notation standard (BPMN). You can also use SAP StreamWork, SAP's on-demand application for collaborative decision-making, to help your distributed teams of process experts to collaboratively and simultaneously design process models.

Execute what you model – Efficiently manage process models from initial high-level definition, to the specification of all technical implementation details, right through to process deployment and execution.

Reuse business functionality from service-enabled applications – Tap into the benefits of service-oriented architecture (SOA) by reusing best-practices-based functionality from software, such as SAP Business Suite, in the form of reusable services.

Provide user-friendly interfaces – Give business users intuitive access to applications and information through various channels, including mobile phones and tablets.

Analyze your processes – Identify potential bottlenecks and gain transparency into processes through powerful monitoring and analytical capabilities.

1.2 Functions of Process Integration Advanced Adapter Engine Extended

The installation option Advanced Adapter Engine Extended (AEX) provides the connectivity capabilities of the Advanced Adapter Engine (AAE) as well as the design and configuration tools (ES Repository and the Integration Directory) to set up scenarios based on the AAE. For a comprehensive overview of Process Integration and the Advanced Adapter Engine see Concepts of Process Integration

For a detailed view of the components of the AEX, see Architecture (Advanced Adapter Engine Extended)

With the Advanced Adapter Engine you can connect SAP systems and non-SAP systems together. You use the various adapters in the Advanced Adapter Engine to convert XML and HTTP-based messages to the specific protocol and format required by these systems, and the other way around.

The Advanced Adapter Engine is based on the Adapter Framework.

For more information, see Adapter Framework

Adapters enable the runtime engines of SAP NetWeaver PI to communicate with different applications. The Advanced Adapter Engine (AEE) is based on AS Java.

AEX supports the mediation capabilities of the AAE. In particular, you can use the following adapters:

- RFC Adapter
- SAP Business Connector Adapter
- File/FTP Adapter
- JDBC Adapter
- JMS Adapter
- SOAP Adapter
- Marketplace Adapter
- Mail Adapter
- RNIF Adapter
- CDIX Adapter
- IDoc Adapter (AAE) (adapter type IDOC_AAE)

HTTP Adapter (AAE) (adapter type HTTP_AAE) You only require an adapter to communicate with SAP systems older than Release 6.20 and with external systems. A direct system connection using proxies and without additional adapters is supported for SAP systems that are based on Application Server 6.20 or higher.

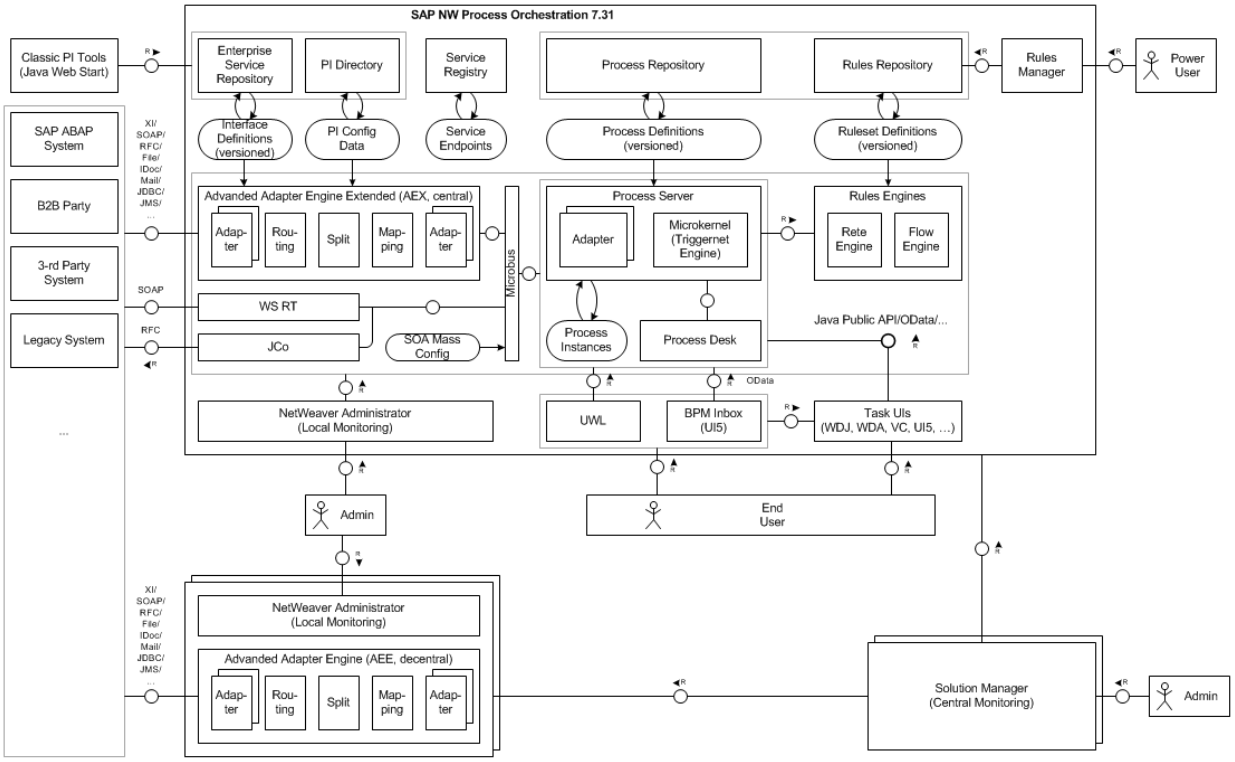
1.3 Architecture of SAP Process Orchestration Package

SAP Process Orchestration is a combination of the usage types Process Integration - Advanced Adapter Engine Extended and Business Process Management.

For a complete description of use-cases and functional units please refer to the SAP NetWeaver 7.4 Master Guide.

The primary runtime components of SAP NetWeaver Process Orchestration are the Advanced Adapter Engine Extended, Process Server, Rules Engine, Business Log and Runtime Repository, Process Desk, Rules Manager, BPM Inbox, and the Universal Worklist (UWL) or BPM Inbox. Sizing considers normal usage across all these components. Special consideration is required for certain use-cases – see section 1.4 Factors that Influence the Performance

Figure 2 Architecture of SAP NetWeaver Process Orchestration



1.4 Factors that Influence the Performance

A brief overview of the key performance influencing factors is provided in this section. A comprehensive Composite Development Architecture Guideline is published on SAP Community Network (http://scn.sap.com). The guideline provides detailed coverage of these factors, including both design time and runtime recommendations for optimizing performance and reliability.

Two worklist implementations are available for task provisioning to end-users. The Universal Worklist is a standard worklist provided by the light-Portal and included with the BPM system. The BPM Inbox is a SAPUI5 based worklist with a modern user experience. The BPM Inbox provides advanced search and filter capabilities. Custom attributes and actions are available for configured tasks. The Inbox also has an alternative stream view which is optimized for occasional users.

1.4.1 Task Provisioning

The UWL and the BPM Inbox are the primary task management interfaces for BPM users. The sort and filter functions of the UWL and BPM Inbox rely on SQL queries. A BPM user may sort and/or filter the worklist frequently, especially when the number of items in the worklist is large. Regardless of which worklist is used the key performance considerations are the same.

Number of 'Ready' tasks in the system

As the total number of tasks increases the SQL queries to return the tasks for any user will need to process more data, thereby increasing the response time and the performance cost

- in general, the higher the total number of ready tasks in the system the greater the performance impact
- more total tasks will increase the query response time for each user, this is a linear increase

Number of potential owners for a task

When there are many potential owners for a task then the task will appear in many users' worklist. Since the task might be claimed by another user at any time all users will tend to refresh their task list more frequently, adding additional load to the system.

- < 10 users as potential owner is termed 'single-approver'
- > 25 users as potential owner is termed 'broadcast'
- > 50 users is not recommended due to high performance cost (SAPS)

Number of tasks for a potential owner

As the number of tasks in the users' worklist increases it is more difficult for the user to manage the task list.

- >50 tasks in the worklist typically cause the user to make extra requests for sorting, filtering and refreshing.
- >50 tasks in the worklist increases the data returned to the user on each refresh, impacting response time and network performance

Number of tasks for a group of potential owners

Groups of users who see the same tasks in the worklist are more likely to attempt to access tasks already claimed or completed by other users. To avoid this, the users will typically refresh the worklist more often.

- >50 tasks in the worklist may lead to extra requests for sorting, filtering and refreshing.
- >50 tasks in the worklist increases the data returned to the user on each refresh, impacting response time and network performance
- when a group of users all see the same >50 tasks in their worklist the negative performance impact may be significantly increased, this depends on how many users are in the group and how many are logged in concurrently
- as the size of the group and the number of tasks visible to the group increases the performance impact may trend in a non-linear manner – this should be avoided !

The standard sizing is based on fewer than 10 users as potential owners for tasks and fewer than 20 tasks in each user's worklist. Deviations from the standard may significantly increase the sizing. Applicable factors are discussed in the sizing section.

Performance Impact: The number of concurrent users accessing the worklist and the number of common tasks among those users must be considered when sizing a BPM system. The numbers of tasks available to the user in

their worklist influences the response time of a request to refresh, filter, or sort the task list. When many users share common tasks there is an increased likelihood that users will attempt to access the same task which can lead to locking. Although the BPM Inbox is highly optimized the same factors which negatively influence the performance of the UWL will also negatively influence the BPM Inbox performance. These factors are at the database level. Queries which return a large number of rows (many tasks for an owner) or which might be executed many times (such as when many users refresh the task list because it changes frequently) cause an increased use of system resources (CPU, memory, and network) to achieve the same work throughput.

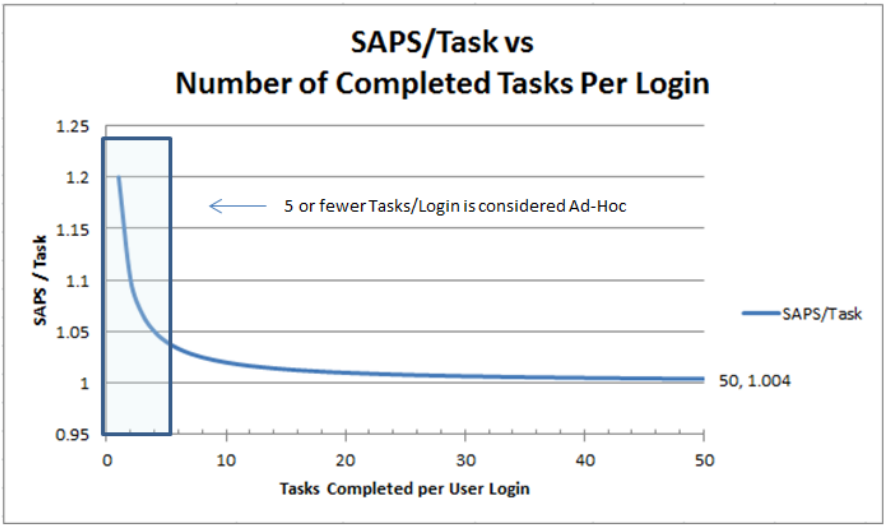
1.4.2 User interaction model (ad-hoc vs. business day)

Number of tasks completed per user login

The standard sizing table assumes that a user completes at least 10 tasks in a single login. The ad-hoc interaction model is applicable when a user completes fewer than ten tasks per login. The business day interaction model is applicable when a user completes ten or more tasks per login.

Performance Impact: A user login is approximately the same performance cost as a completed task. As the number of completed tasks per user login increases the performance cost per task (in SAPS) decreases. When a user logs in once and completes a significant number of tasks during that session this is considered as 'business day' usage. When a user logs in and completes one or only a few tasks per session this is considered as 'ad-hoc' usage. Ad-hoc SAPS cost is approximately 20% higher than the SAPS cost of business day usage. The sizing tables in Section 3.3.1, Tables 2 and 3 assume business day interaction model. For ad-hoc interaction model you must multiply the estimated SAPS by a factor of 1.2. Figure 2 demonstrates the relationship between number of tasks completed per login and the corresponding performance cost per task.

Figure 3 Relationship of SAPS/Task and Tasks Completed per User Login



The standard sizing assumes the typical user completes 20 tasks per hour (an average of 3 minutes per task) with a single login. Login is a significant system operation in terms of both CPU and memory.

1.4.3 Number of process instances in the system (active and completed)

Number of active process instances (consumes memory and execution threads)

A process which is currently running (such as executing a human task or an automated activity) consumes Java Virtual Machine (JVM) heap memory and one or more execution threads. The amount of memory and execution threads available must be sufficient for the expected concurrency. The amount of memory consumed per session is primarily related to the size of the process context.

Number of active process instances (persisted to database)

Processes which are currently active but not running will have their data persisted to the database. This data consumes database space. If you have many long-running tasks which are periodically activated this can impact the performance. When a hibernated process instance becomes active in the JVM then the process data must be serialized from the database into memory.

Number of completed process instances prior to archiving

The total number of completed process instances in the system must be considered when sizing the database. Each process instance and its related data require space in the database until they are archived. The amount of database space needed for each process instance varies; depending on the amount of data persisted in the process.

Performance Impact: Process context is held in heap memory for process that are currently running (i.e. executing an activity or transition) or processes that recently ran but which have not yet been persisted to the database (hibernated). The process context remains in the database until a process instance and the related task instances are archived. The number of active processes can therefor influence heap memory usage and database storage. The amount of memory and disk per process instance depends on the size of the process context.

1.4.4 Number of inter-system or intra-system interactions

Automated activities, synchronous or asynchronous

Intermediate message events

Performance Impact: Outbound system interactions hold a process server execution thread for the entire request and response cycle. For synchronous calls this can lead to thread exhaustion at relatively low concurrency if the other system's response time is long. Synchronous outbound service calls can also return large data sets. Care should be taken to avoid returning unneeded data in synchronous responses as this impacts system memory. Complex data mappings into the process context should also be carefully considered as these consume both memory and CPU. Inbound system interactions include both start events and intermediate message events (both asynchronous). Care should be taken to optimize the data transferred and mapped into the process context.

The Composite Architecture Guideline provides a thorough discussion of the various ways that data handling in the process context impacts system performance. The standard sizing considers average context object size of 10 KB and basic flat structures, Deep data object structures which require complex mappings should be avoided.

1.4.5 Use of Conditional Start

The conditional start is a special case of the start event which ensures only a single process instance runs for a specific unique matching condition. This allows for messages to be either collected by an already running process instance or in the case when no matching process is running a message will trigger the start of a new process instance. The conditional start uses enqueue locks to ensure that two messages with the same condition cannot both start process instances. During the condition check the entire service interface is locked. This means the processing of messages on a conditional start interface is serialized. When a message matches an already existing process the lock duration is typically only a few milliseconds. When a new process instance is started by an incoming message the lock may be held for several hundred milliseconds. Based on the CPU capacity available to the enqueue server (part of the SCS instance) the throughput for conditional start processes should be carefully monitored to prevent an overload of the enqueue server.

1.4.6 Size of data transferred in system interactions

Data mapped into inputs of automated activities

Data mapped from the outputs of start events, intermediate message events, and synchronous automated activities

Performance Impact: The size of the data objects transferred in system interactions should be optimized to improve system performance. As well; the structure of the data should be optimized to minimize the necessity for complex data mappings or XML parsing.

1.4.7 Number and size of data persistence actions (memory and database)

Data persisted into process context objects

Repetitive persistence to process context objects (locking potential)

Performance Impact: Process context data objects should contain only the minimum data necessary for the process flow. Writing to the objects should be carefully managed to avoid potential deadlocks such as might occur if two parallel branches of a process were to write to an object at the same time. Sizing is based on up to 50 data elements in a context object with a flat data structure.

1.4.8 UI technology (local or remote, chosen technology)

Web Dynpro UIs hosted on the local system
SAPUI5 UIs hosted on the local system using BPM APIs or BPM OData services
Web Dynpro UIs hosted on other systems (ABAP/Java) using CHIPs
Remote UIs based on SAPUI5, Mobile, etc.., using API integration

Performance Impact: SAP NetWeaver BPM sizing considers only the case of locally hosted UIs. This is typically the highest performance cost as the local runtime must also process the UIs of the application. When using remote UIs or UIs built against the BPM APIs you should carefully test the performance impact on the hosting system and size that landscape accordingly. Sizing for remotely hosted UIs is beyond the scope of this sizing guideline.

The standard sizing considers Web Dynpro or SAPUI5 UIs of moderate complexity, similar to the UIs generated by the wizards based on the task data context objects. This implies that the UIs work only against the process context data (no external queries) and do not require navigation between tabs (not multi-tabbed). The standard sizing considers up to 50 data fields are available for read/write operations within the UI, although the recommendation is that the number of data elements be optimized and unnecessary elements removed.

1.4.9 Message Size

Message size is a key performance factor in AEX and Process Orchestration Scenarios. The standard message size considered for AEX sizing is 10-100 KB. Larger message sizes incur a significant performance penalty in terms of both CPU and memory. Each additional 100KB of message size above 100KB can increase the CPU and memory required by up to 50%. For each additional 100KB increment of message size the estimated SAPS must be multiplied by a factor of 1.5. For very large messages (above 2MB) the sizing tables should be used with caution as the sizing factor of 1.5 no longer scales linearly. When very large messages are required the sizing should be based on testing of the actual message interfaces and using messages which very closely resemble those that will be passed in production.

Within the AEX, asynchronous messages are persisted at least twice during the processing. The persistence of very large messages consumes significant CPU and memory resources and increases the total time for message processing. Message concurrency (and throughput) is also reduced as message size increases.

2 Sizing Fundamentals and Terminology

SAP provides general sizing information on the SAP Service Marketplace. For the purpose of this guide, we assume that you are familiar with sizing fundamentals. You can find more information at service.sap.com/sizing → *Sizing Guidelines* → *General Sizing Procedures*.

This section explains the most important sizing terms, as these terms are used extensively in this document.

2.1 Sizing

Sizing means determining the hardware requirements of an SAP application, such as the network bandwidth, physical memory, CPU processing power, and I/O capacity. The size of the hardware and database is influenced by both business aspects and technological aspects. This means that the number of users using the various application components and the data load they put on the server must be taken into account.

2.2 Benchmarking

Sizing information can be determined using SAP Standard Application Benchmarks and scalability tests (www.sap.com/benchmark). Released for technology partners, benchmarks provide basic sizing recommendations to customers by placing a substantial load upon a system during the testing of new hardware, system software components, and relational database management systems (RDBMS). All performance data relevant to the system, user, and business applications are monitored during a benchmark run and can be used to compare platforms.

2.3 SAPS

The SAP Application Performance Standard (SAPS) is a hardware-independent unit that describes the performance of a system configuration in the SAP environment. It is derived from the Sales and Distribution (SD) Benchmark, where 100 SAPS is defined as the computing power to handle 2,000 fully business processed order line items per hour. (For more information about SAPS, see www.sap.com/benchmark → *Measuring in SAPS*).

2.4 Initial Sizing

Initial sizing refers to the sizing approach that provides statements about platform-independent requirements of the hardware resources necessary for representative, standard delivery SAP applications. The initial sizing guidelines assume optimal system parameter settings, standard business scenarios, and so on.

2.5 Expert Sizing

This term refers to a sizing exercise where customer-specific data is being analyzed and used to put more detail on the sizing result. The main objective is to determine the resource consumption of customized content and applications (not SAP standard delivery) by comprehensive measurements. For more information, see service.sap.com/sizing → *Sizing Guidelines* → *General Sizing Procedures* → *Expert Sizing*.

2.6 Configuration and System Landscaping

Hardware resource and optimal system configuration greatly depend on the requirements of the customer-specific project. This includes the implementation of distribution, security, and high availability solutions by different approaches using various third-party tools. In the case of high availability through redundant resources, for example, the final resource requirements must be adjusted accordingly.

There are some “best practices” which may be valid for a specific combination of operating system and database. To provide guidance, SAP created the NetWeaver configuration guides (service.sap.com/instguides → *SAP NetWeaver*).

2.7 System Landscape Governance

Missing guidance and lack of clear recommendations about optimal landscape setups for SAP products belong to the prominent challenges often addressed by customers in the past.

SAP products like SAP NetWeaver Process Orchestration support a high level of flexibility for different deployment options with respect to the mentioned architectural decisions. However, the optimal landscape layout is mostly a trade-off between flexibility and simplicity as pointed out before. Dependent on the specific situation and the kind of capabilities that should be deployed, SAP System Landscape Governance provides clear recommendations, how to deploy SAP Process Integration Advanced Adapter Engine, SAP NetWeaver Business Process Management, and SAP NetWeaver Process Orchestration components within your solution landscape.

The current System Landscape Governance recommendations are maintained at the Landscape Governance wiki.

3 Sizing SAP NetWeaver Process Orchestration

The purpose of initial sizing is to provide a coarse grained estimate of resource requirement suitable for high-level planning of system landscapes and product licensing. User sizing is based on the number of concurrent users accessing the system at peak. Task based sizing is based on the peak throughput of human tasks. Task based sizing is the preferred approach for initial sizing as it is more accurate than user based sizing.

For initial sizing it is sufficient to know either the number of concurrent users who will access the BPM application or the peak throughput in terms of the number of completed processes and human tasks per hour.

3.1 Assumptions

- The initial sizing approach detailed in this guide is based on the SAP NetWeaver BPM reference business processes (section 3.1.1).
- Task provisioning (access to tasks) is provided via the BPM Inbox of the SAP Enterprise Portal.
- Task UIs are implemented in Web Dynpro Java using the UI generator of NetWeaver Developer Studio (NWDS) .
- System parameters and tuning are 'as-is' provided by the SAP delivered *Performance_Template_BPM_High_Load_Scenarios* zero-admin template. Any other applicable tuning or configurations changes are documented in SAP notes.

3.1.1 SAP NetWeaver Process Orchestration Reference Business Processes

In order to validate the performance of SAP NetWeaver Process Orchestration several reference business process are tested using automated tools. These reference scenarios are purposefully designed to use a comprehensive set of product features and capabilities. The test scenarios are run with a variety of inputs and conditions which provide excellent coverage of factors such as user concurrency, message size, number of processes in the system (active and inactive). Taken together, the results of the tests provide the basis for the sizing tables presented in this guide.

The test scenario evaluates both ad-hoc and business day usages for the process. The ad-hoc scenario is the highest resource requirement due to the fact the login is itself an expensive operation. The business-day scenario is a best-case requirement as the login occurs only once per user and each user completes more than twenty tasks per login. The BPM Inbox is the configured task provider and offers improved performance when compared with the UWL.

Two system-centric reference business processes are used for performance validation and sizing. These test scenarios are used to derive the unit-cost for automated activities (synchronous and asynchronous) and intermediate message events and the related data object mappings. The scenarios are tested using both plain BPM connectivity and stateful mediated connectivity using the standard Java Proxy Runtime connectivity of the AEX.

The reference processes are used to determine the performance characteristics (CPU and memory consumption) of commonly used artifacts such as BPM tasks, Intermediate Message Events, Automated Activities, Integration Flows, Message Mapping, and Message Routing. Analysis of the system and database after test completion provides the basis for sizing of the data within the database and on the file system.

3.2 Sizing Guideline –Sizing Tables

Sizing tables are provided for usage based on the use of the BPM Inbox for user task provisioning and for the Universal Worklist (UWL). The UWL has a higher performance cost in comparison, due to the overhead of the Portal framework.

All sizing tables use the following standard values for SAPS. The standard values are derived from actual results achieved in performance tests and are normalized for the recommended 65% maximum CPU utilization.

- Human Task = 1.0 SAPS each (using BPM Inbox)
- Human Task = 1.6 SAPS each (using UWL)
- Non-mediated (stateless) Message = 0.04 SAPS each or 40 SAPS / 1000 messages
- Mediated (stateful) Message = 0.08 SAPS each or 80 SAPS / 1000 messages

All sizing tables are based on the following assumptions

- A user is assumed to complete 20 tasks/login (business day scenario)
- Message size is 100KB or less

The memory sizing is the needed JAVA heap memory which is in addition to the base heap used by the running application server.

3.2.1 Simple User Based Sizing with BPM Inbox

- A user is assumed to complete 20 tasks/hour

User Based (no messaging in process)				
Size		Users	SAPS	MEM (MB)
X-Small	up to…	100	2,000	1,600
Small	up to…	200	4,000	3,200
Medium	up to…	400	8,000	6,400
Large	up to…	800	16,000	12,800

Sizing using the User Based table is appropriate for basic estimates of potential resource usage. It is possible to use the initial sizing tables when only an estimate of the number of BPM users is known. The user based table assumes that a user logs in once and completes twenty tasks per hour. User based sizing is applicable only to pure BPM scenarios and not for Process Orchestration scenarios. The table is appropriate only for rough sizing.

3.2.2 Simple User Based Sizing with Universal Worklist

- A user is assumed to complete 20 tasks/hour

User Based (no messaging in process)				
Size		Users	SAPS	MEM (MB)
X-Small	up to…	100	3,200	1,600
Small	up to…	200	6,400	3,200
Medium	up to…	400	12,800	6,400
Large	up to…	800	25,600	12,800

3.2.3 Simple Task Based Sizing (BPM only, no AEX and no messaging) using BPM Inbox

- Task based sizing does not assume any per user throughput, it based only on task throughput

SIMPLE TASK BASED (no messaging)				
Size		Task/Hr	SAPS	MEM (MB)
X-Small	up to…	2,000	2,000	1,600
Small	Up to…	4,000	4,000	3,200
Medium	up to…	8,000	8,000	6,400
Large	up to…	16,000	16,000	12,800
X-Large	up to…	32,000	32,000	25,600
XX-Large	up to…	64,000	64,000	51,200

3.2.4 Simple Task Based BPM Sizing (BPM with related Messaging, no use of AEX) using BPM Inbox

- Message throughput is based on an assumed 2 messages per task
- Task based sizing does not assume any per user throughput, it is only based on task throughput
- Each task is assumed to have two related messages such as an automated activity and an intermediate message event
- All messages are assumed to be sent directly to or from BPM and do not use AEX

BPM ONLY (basic messaging without AEX)					
Size		Task/Hr	Msg/Hr	SAPS	MEM (MB)
Small	up to…	4,000	8,000	4,000	3,200
Medium	up to…	8,000	16,000	8,000	6,400
Large	up to…	16,000	32,000	15,000	12,800
X-Large	up to…	32,000	64,000	29,000	25,600
XX-Large	up to…	64,000	128,000	57,000	51,200

3.2.5 Basic Process Orchestration Sizing

- Message throughput is based on an assumed 3 messages per task
- Task based sizing does not assume any per user throughput, it is only based on task throughput
- Task provisioning is provided by the BPM Inbox
- Each task is assumed to have three related messages such as automated activities or intermediate message events
- All messages are assumed to be stateful using AEX for mediation

Basic Process Orchestration (Stateful messaging, 3 msg/Task)					
Size		Task/Hr	Msg/Hr	SAPS	MEM (MB)
Small	up to…	4,000	12,000	5,000	4,000
Medium	up to…	8,000	24,000	10,000	8,000
Large	up to…	16,000	48,000	20,000	16,000
X-Large	up to…	32,000	96,000	40,000	32,000
XX-Large	up to…	64,000	192,000	80,000	64,000

3.2.6 Moderate Process Orchestration Sizing

- Message sizing is based on an assumed 6 messages per task
- Task based sizing does not assume any per user throughput, it is only based on task throughput
- Task provisioning is provided by the BPM Inbox

- Each task is assumed to have six related messages such as automated activities or intermediate message events
- All messages are assumed to be stateful using AEX for mediation

Moderate Process Orchestration (Stateful messaging, 6 msg/Task)					
Size		Task/Hr	Msg/Hr	SAPS	MEM (MB)
X-Small	up to…	2,000	12,000	3,000	2,800
Small	up to…	4,000	24,000	6,000	5,600
Medium	up to…	8,000	48,000	11,000	11,200
Large	up to…	16,000	96,000	21,000	22,400
X-Large	up to…	32,000	192,000	41,000	44,800
XX-Large	up to…	64,000	384,000	82,000	89,600

3.2.7 Complex Process Orchestration Sizing

- Message sizing is based on an assumed 12 messages per task
- Task based sizing does not assume any per user throughput, it is only based on task throughput
- Task provisioning is provided by the BPM Inbox
- Each task is assumed to have twelve related messages such as automated activities or intermediate message events
- All messages are assumed to be stateful using AEX for mediation

Complex Process Orchestration (Stateful messaging, 12 msg/Task)					
Size		Task/Hr	Msg/Hr	SAPS	MEM (MB)
X-Small	up to…	2,000	24,000	4,000	2,800
Small	up to…	4,000	48,000	8,000	5,600
Medium	up to…	8,000	96,000	16,000	11,200
Large	up to…	16,000	192,000	32,000	22,400
X-Large	up to…	32,000	384,000	64,000	44,800
XX-Large	up to…	48,000	576,000	95,000	89,600

3.2.8 AEX Only Sizing (Stateless Messaging)

- Stateless messaging is sized as standard PI-AEX using quicksizer
  - https://service.sap.com/quicksizer

3.2.9 Advanced Process Orchestration Sizing (considers message size)

- Stateful messaging with large messages must consider the additional CPU and memory consumption of managing the large messages through both the AEX and BPM infrastructure as well as the potential impact of large messages on the process context
- For messages larger than 100KB you should increase the CPU by a factor of 1.5 for EACH INCREMENT of 100KB size.
- No tables are provided for advanced process Orchestration sizing as the estimation of SAPS can only accurately be estimated using test results based on scenarios which closely match the specific implementation requirements of the project
- It is possible to use a basic sizing factor to roughly estimate the additional SAPS required for advanced scenarios – but the resulting values should not be considered as accurate as the base tables. Such estimates should be used for rough sizing purposes only.

3.3 Sizing Factors

BPM Related Factors

UWL vs. Inbox

The Inbox is a stand-along Task Worklist and is highly optimized. The UWL requires the Portal framework and has resource overhead related to that. The Inbox has lower CPU and Memory requirements. Standard sizing is based on the use of the BPM Inbox. When using the UWL a factor of 1.6 should be applied to the estimated SAPS requirement.

Ad-Hoc vs. Business Day Simulation

A user login is approximately 1/5 the cost of completing a BPM task. Standard sizing is based on the assumption that a user completes 20 tasks per login. When a user completes only 1-5 tasks per login a factor of 1.2 should be applied, between 5-20 tasks use a factor of 1.1.

AEX Related Factors

Message Size

The standard message size assumed in the tables is < 100KB. For large messages you must increase the estimated SAPS by 50% for each 100KB increment in message size beyond 100KB. The required memory must be increased by 40% for each 100KB increment in message size beyond 100KB.

3.4 Database Sizing

Initial database size and space requirements are given in the Master Guide. You can size the SAPS and CPU requirement of the database server by using standard ratios for the split of the workload between the application and the database. The database server typically consumes 35% of the total estimated SAPS. The application

server(s) consume the other 65%. When using the sizing tables you can size for distributed landscapes using these ratios.

In order to size the space required for database tables you must consider both the concurrent (running or in-progress) workload as well as the completed workload which has not yet been archived. You can estimate the size of your process context by considering the total size of all the XSDs for you context data objects when they are filled with typical data. For in-progress process instances the process context will take 50% of the size of the filled XSDs. Once a process instance is completed the transactional tables no longer contain the process context. The context objects are then only stored in the business logs. A completed process instance will still consume 50% of the context objects size and this will be stored only in the business logs. When a process instance is archived the data is removed entirely from the database and only a small amount of metadata remains to facilitate the access to the archived process details.

The reference business process has 100 KB of context data when considering the size of the filled XSD. AN in-progress instance will consume approximately 50KB of database space. Once the process is archived it consumes approximately 50KB of database space.

The data retention strategy is a key factor in determining the database space requirements. To estimate the total space requirement for process data you must consider both the size of the concurrent in-progress instances as well as the total number of instances which will be held in a completed state on the system. An example is helpful.

- The required business throughput is 5,000 process instances per day.
- The typical process has a lifespan of one week
- The process context size is 300KB
- Completed process instances are archived once they are six months old.

Given these inputs we can estimate the required DB size, assuming a 5 day work week for 50 weeks annually

Your high-water mark is based on the number of In-Progress and Completed Process Instances which are less than six months old.

Total concurrent in-progress instances: 5,000 x 5 = 25,000 (weekly)

Total in-progress context size: 25,000 x 400 KB = 10 GB

Total database size of in-progress instances: 10 GB x 50% = 5 GB

Total completed process instances: 5,000 x 5 (days) x 25 (weeks) = 625,000 completed process instances

Total completed process context size: 625,000 x 400 KB = 250 GB

Total database size of completed process instances = 250 GB x 50% = 125 GB

The combined total database space required is 130 GB, which is the sum of the in-progress instances (5 GB) and the completed but not yet archived instances (125 GB). If you maintain the archiving of all process data over 6 months old then this level should remain fairly constant.

For Process Orchestration on SAP HANA there is also the consideration of the database in-memory. Please refer to the SAP Note “Sizing for SAP Process Integration - Advanced Adapter Engine (PI-AEX) , Business Process Management (BPM), and Process Orchestration on SAP In-Memory Database (HANA)” for more details.

3.5 Sizing Guideline – Advanced Sizing

Advanced sizing requires that the process is tested on production-like hardware using realistic inputs. The most accurate sizing is achieved when a complete scenario is fully implemented and it is tested at anticipated concurrency and throughput levels using data which closely matches the production data. Advanced sizing is beyond the scope of this document.

4 Sizing SAP Advanced Adapter Engine Extended

Sizing for PI-AEX usage is provided in the Quicksizer – http://service.sap.com/sizing

In order to use the quicksizer you need to know certain information about your busness requirements.

4.1 Basic Sizing information needed for PI-AEX

Message size

The average size for XML representation of messages in a scenario is a key input. If the size for XML representation is unknown, you can use the following guidelines to estimate the XML size:

- For flat files, calculate a factor of 10 for conversion to XML (multiply flat file size by 10).

- For IDocs, you can use the following approach: In a sender or receiver system, use the IDoc test tool (transaction WE19) to write a sample IDoc to a file using an XML File port (or a File port if an XML File port is not available); then use the file size as XML message size.

It is advisable to use quick sizing results for scenarios that include messages larger than 10 MB with great care. These scenarios may not work in the default configuration and especially the memory requirements depend significantly on several configuration and scenario details that cannot be considered in the Quick Sizer tool. Therefore the QuickSizer tool may overestimate memory requirements for these scenarios.

Number of messages

Another required input is the peak number of outbound messages to be processed in a given timeframe for a scenario. For an Adapter Engine, AEX or using integrated configurations in general, all processed messages must be counted. For peak load calculation it might be helpful to distribute processing over time (e.g. by using batch functionality).

Processing mode

Synchronous or asynchronous processing mode determines how messages are handled in PI.

- Synchronous messaging means that a sender application has to wait until a message is delivered to a receiver application, and that the receiver sends an immediate response like in a synchronous remote function call. This applies to Quality of Service Best Effort.

- Asynchronous messaging means that a sender application sends a message to the Adapter Engine and receives a technical OK that the message has been received; the sender does not wait for immediate response by the receiver application. With this processing mode, the Adapter Engine needs to store data and deliver the message later (asynchronously) to the receiver application. Asynchronous processing is standard in PI and applies to Quality of Service Exactly Once and Exactly Once In Order. If the mode is in order processing (serialized), the Adapter Engine cannot make use of parallel processing effectively.

Memory sizing:

Memory sizing is based on the number of required CPU resources (SAPS value). Memory requirements increase as additional Application Server instances (ABAP or Java) need to be installed. Additionally, individual memory

requirements by single messages due to their message size are considered. As result, the maximum for both memory requirements is calculated. The number of parallel messages to be processed is derived from the given throughput value. Depending on the type of adapter at least six to eight times the message size is required as memory for processing. Therefore, available OS and Java heap limits must be taken into consideration.

Please note that memory sizing might overestimate memory requirements for several scenarios with large message sizes (e.g. 50 MB) and low message volume, if it is possible to reach an equal or almost equal distribution of the different scenario loads over the peak load time, because the Quick Sizer tool will sum up memory requirements for all scenarios which is wrong if messages from several scenarios are processed sequentially (but is correct if they are processed in parallel).

4.2 Advanced Adapter Engine Extended Sizing

Definition

An Adapter Engine or an AEX runs on the J2EE Engine. Process Orchestration scenarios require both an inbound adapter and an outbound adapter and make use of XI 3.0 connectivity and Java Proxy Runtime. An integrated scenario refers to the AEX using both an inbound and outbound adapter. The term 'integrated' is this sense, as in 'integrated scenario' or 'integrated configuration' does not necessarily imply the scenario is a process orchestration scenario. The term can be confusing though as a Process Orchestration scenario, which is based on an integration flow, will use an integrated configuration at runtime. In the case of Process Orchestration however the integrated configuration is a runtime artifact generated on deployment of the integration flow, whereas in the classical scenario the integrated configuration is created in the Integration Directory as a design-time artifact and then deployed as a runtime component.

The sizing information needed for advanced sizing of an AEX is the following:

Message size, Number of messages, Processing mode

Inbound Adapter

- J2EE: J2EE-based adapters (File, FTP, JDBC, JMS, RFC, SOAP, Mail, IDoc, HTTP, XI).
- Industry Speak: B2B adapters (RNIF, CIDX).
- 3rdParty: Certified 3rd party adapters from partners.

Outbound Adapter

Outbound adapters send requests to a receiver application and are therefore also called receiver adapters. The available adapter types are the same as for the inbound adapters.

Integrated Scenario

Integrated scenarios, also called integrated configurations are executed locally on the Adapter Engine. Note that integrated configurations require the selection of an inbound and an outbound adapter.

Adapters are divided into different categories referring to the adapter type. Several adapters are grouped together in subcategories as the sizing estimate is quite comparable (for example, J2EE-based adapters like File, FTP, JDBC). Only adapters implemented in the J2EE Adapter Framework can be selected.

All Process Orchestration scenarios are Integrated Scenarios, but not all Integrated Scenarios are Process Orchestration scenarios.

5 Sizing Process Orchestration and AEX

Sizing for a Process Orchestration system is simply a combination of the sizing for the three distinct scenario types that can be implemented.

a) Advanced Adapter Engine only scenarios (stateless message mediation)
b) Business Process Management only scenarios (no message mediation)
c) Integrated AEX and BPM Process Orchestration scenarios (stateful message mediation)

AEX only sizing is classical message based sizing. AEX sizing is available in the Quicksizer, https://service.sap.com/quicksizer . BPM only sizing is based on task and message throughput – where the messages are not mediated by AEX. Integrated AEX and BPM sizing is for scenarios when BPM processes use AEX mediated message exchange for inter-system and or intra-system messaging.

A Process Orchestration system is likely to satisfy business requirements that span the classical PI-AEX message scenarios, plain BPM scenarios, and combined PI-AEX and BPM scenarios. A complete system sizing should include all the implemented scenarios.

5.1 Size for AEX

Use the quicksizer for AEX to estimate the SAPS requirement for the messaging throughput requirement.

5.2 Size for BPM

Use plain BPM sizing by consulting the BPM sizing table provided in section 3.2.3 of this guide.

5.3 Size for Combined AEX and Process Orchestration

You can use the basic tables in sections 3.2.5 (Basic), 3.2.6 (Moderate), and 3.2.7 (Advanced) for estimates where the exact task and/or message throughput may not be well known. These tables are valid for rough or initial sizing.

You can use the Simple BPM sizing table in section 3.2.3 to estimate a known BPM task throughput and then use the AEX quicksizer for AEX messaging.

A full system sizing is simply the sum of the separate sizing tables or results. You can, for instance, combine the result taken from the BPM sizing table 3.2.3 with the AEX sizing result obtained from the quicksizer.

6 SAP HANA Platform Sizing

Process Orchestration sizing is provided for all supported operating system and database platforms. The sizing for the application servers (75% of total estimated SAPS) is the same regardless of the database platform. As of SAP NetWeaver 7.40 the use of SAP HANA is supported as a database for SAP BPM, SAP Process Integration AEX, and SAP Process Orchestration. There are special sizing considerations when using SAP HANA as the database. The typical SAPS estimated for the database server (25% of total estimated SAPS) is slightly higher for SAP HANA. SAP HANA has its own specific sizing guidelines and there are special considerations for running SAP PI-AEX, SAP BPM, or Process Orchestration on SAP HANA. In particular, when there is a high volume of logged PI messages the SAPS requirement for SAP HANA DB may be higher than for other DB platforms. In Process Orchestration Scenarios there is also a higher load placed on the HANA database which is in relation to the number of messages flowing (and logged) between BPM and PI. With SAP BPM you must consider the business log-level when logging is set to higher than 'Standard' as this places a higher load on the database due to increased data write operations.

You must follow the HANA specific guides and recommendations to assure that your DB load is within the expected ratio of total SAPS estimated. When using the sizing tables you can split the application and database workload for all DB platforms using 75% Application and 25% Database. With SAP HANA as the database you should increase the SAPS estimated for the database by 40%. This ensures the recommended increased CPU capacity for HANA systems optimized to share OLTP and OLAP workloads. For SAP HANA you must also consider the database size as the database will reside fully in memory of the HANA system. The result for a Process Orchestration system based on SAP HANA is a total sizing which is approximately 10% higher than for other database platforms.

6.1 Recommendations for Process Orchestration on HANA Sizing

For sizing of Process Orchestration on HANA the standard value in the tables may be used as a basis for the estimation of the total SAPS requirement. Use the ratio of 75% SAPS to application server and 35% SAPS to the SAP HANA database server and size the database server accordingly.

Example Sizing for Process Orchestration on HANA			
	STANDARD	On HANA Sizing	Net SAPS on HANA
Total Estimated SAPS Sizing	20,000 (100%)	21,500 (108%)	+1,500 SAPS (8%)
Application Server SAPS	15,000 (75%)	15,000 (75% of standard)	Same
Database Server SAPS	5,000 (25%)	7,000 (30% of standard)	+1,500 SAPS (30%)

The HANA database will use additional CPU and memory resources to ensure that both OLTP and OLAP workloads perform well.

6.1.1 Sizing HANA Memory for SAP BPM, PI-AEX, and Process Orchestration

The system and application database schemas reside in memory with SAP HANA. You must ensure that the available memory for your SAP HANA instance is sufficient to meet immediate and future needs. The memory requirement for the database can be estimated based on the business throughput and the data retention strategy. For purposes of initial sizing you can refer to 3.4 Database Sizing, of this document. For SAP Hana Memory use 65% of the estimated disk size plus 20% safety factor. The typical compression rate for Process Orchestration data on SAP HANA 35%, but this can vary significantly depending on the scenario and some configuration and optimization options.

6.1.2 Data Retention Strategies for SAP BPM, PI-AEX, and Process Orchestration

It is critical that you manage your application data carefully to ensure the system performs optimally. The standard recommendation for Business Process Data is that it be retained for two-months on the system and data over two-months is archived. Archived business process data is still available in the system for audit and review purposes. By reducing the data in the transactional tables and shifting it to the business logs your system will perform better and require less computing resources.

6.1.3 Performance Optimization Notes for SAP BPM, PI-AEX, and Process Orchestration on SAP HANA In-Memory Database

 The sizing values provided in this document are valid for SAP HANA when the applicable performance optimization notes have been applied to the system.

 SAP Note 2039873

Table classification, index optimizations and Memory LOBS for SAP Process Integration Advanced Adapter Engine (PI-AEX) , Business Process Management (BPM), and Process Orchestration

The note 2039873 describes optimizations to the table definitions and storage for key transactional tables of SAP PI-AEX and SAP BPM. These optimizations are implemented using SQL script provided as an attachment to the note.

SAP Note 2039874

Performance Characteristics of SAP Process Integration Advanced Adapter Engine (PI-AEX) , Business Process Management (BPM), and Process Orchestration on SAP HANA Database

The note 2039874 discusses key performance factors for HANA with respect to its' operation as a database platform for high-throughput OLTP application(s). The note references key HANA documentation topics related to Technical Operations, SQL and System Views Reference, Memory Usage, and Performance Analysis.

SAP Note 2039820

Sizing for SAP Process Integration - Advanced Adapter Engine (PI-AEX) , Business Process Management (BPM), and Process Orchestration on SAP In-Memory Database (HANA)

The note 2039820 provides a detailed method for sizing the CPU and memory of SAP HANA when used as the database platform for SAP Process Orchestration. The note includes further detail regarding the sizing of memory and disk space. Managing data retention per business and regulatory requirement is a key operations topic.

7 Comments and Feedback

Please direct comments, questions, and feedback to the document author.

Nicholas Holshouser
SAP HANA Platform, Integration and Orchestration
nicholas.holshouser@sap.com

8 Glossary

Ad-Hoc – Users access a very low number of tasks per system login. Ad-hoc usage of BPM increases the required SAPS by as much as 50% when a user completes only one task on each login.

Business Day – Users ten or more tasks per login. Business day usage of BPM is the most efficient use of system resources as the overhead for logins is only a small percentage of the total workload for task execution.

Stateless messaging – message exchange with external systems using only the PI Advanced Adapter Engine

Stateful messaging – message exchange with external systems using the orchestration capabilities of the Business Process Management system

www.sap.com/contactsap

Material Number