PI Best Practices: Sizing & Performance Tuning

Applicable Releases:
SAP Exchange Infrastructure 3.0
SAP NetWeaver Process Integration 7.0
SAP NetWeaver Process Integration 7.1 (Including Enhancement Package 1)

Topic Area:
SOA Middleware

Capability:
Service Bus

Version 1.1
November 2009
These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

These materials are provided "as is" without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials. SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.

SAP NetWeaver “How-to” Guides are intended to simplify the product implementation. While specific product features and procedures typically are explained in a practical business context, it is not implied that those features and procedures are the only approach in solving a specific business problem using SAP NetWeaver. Should you wish to receive additional information, clarification or support, please refer to SAP Consulting.

Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.

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 expressively prohibited, as is any decompilation of these components.

Any Java™ Source Code delivered with this product is only to be used by SAP’s Support Services and may not be modified or altered in any way.
## Document History

<table>
<thead>
<tr>
<th>Document Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>First official release of this guide</td>
</tr>
<tr>
<td>1.10</td>
<td>Updates for Enhancement Package 1 (EHP 1) of SAP NetWeaver Process Integration 7.1</td>
</tr>
</tbody>
</table>
## Typographic Conventions

<table>
<thead>
<tr>
<th>Type Style</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example Text</strong></td>
<td>Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation</td>
</tr>
<tr>
<td><strong>Example text</strong></td>
<td>Emphasized words or phrases in body text, graphic titles, and table titles</td>
</tr>
<tr>
<td><strong>Example text</strong></td>
<td>File and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.</td>
</tr>
<tr>
<td><strong>Example text</strong></td>
<td>User entry texts. These are words or characters that you enter in the system exactly as they appear in the documentation.</td>
</tr>
<tr>
<td><strong>&lt;Example text&gt;</strong></td>
<td>Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.</td>
</tr>
<tr>
<td><strong>EXAMPLE TEXT</strong></td>
<td>Keys on the keyboard, for example, F2 or ENTER.</td>
</tr>
</tbody>
</table>

## Icons

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🔴</td>
<td>Caution</td>
</tr>
<tr>
<td>📜</td>
<td>Note or Important</td>
</tr>
<tr>
<td>🌐</td>
<td>Example</td>
</tr>
<tr>
<td>🧵</td>
<td>Recommendation or Tip</td>
</tr>
</tbody>
</table>
# Table of Contents

1. **Background Information** ........................................................................................................... 1

2. **Sizing Guidelines** ...................................................................................................................... 2

3. **Performance Tuning** ................................................................................................................ 3
   3.1 Preliminary Considerations ...................................................................................................... 3  
   3.2 Tuning Mechanisms .................................................................................................................. 6  
       3.2.1 Scaling & Tuning of IS/AE ............................................................................................... 6  
       3.2.2 Point-to-Point ................................................................................................................. 7  
       3.2.3 Local Processing in AAE ............................................................................................... 8  
       3.2.4 Message Packaging ........................................................................................................ 9  
       3.2.5 Message Packaging versus Local Processing in AAE and Point-to-Point .................... 15  
       3.2.6 Message Prioritization ................................................................................................... 16  
       3.2.7 Time-controlled message processing ............................................................................. 16  
       3.2.8 Tuning of BPE .................................................................................................................. 17  
       3.2.9 Special Cases .................................................................................................................. 17
1. Background Information

This guide is part of a how-to guide series providing best practices and guidelines for SAP NetWeaver Process Integration (PI) and SOA processes. The current guide deals with best practices in terms of sizing and performance. Where information is available otherwise, the guide will refer to the same rather than to repeat it.
2. Sizing Guidelines

For information about sizing SAP solutions in general, please refer to SAP Service Marketplace (SMP), alias sizing: http://service.sap.com/sizing (SMP logon required).

Initial hardware sizing can be done using the Quicksizer tool, accessible at the SAP Service Marketplace, alias quicksizing: http://service.sap.com/quicksizing (SMP logon required).

For SAP NetWeaver PI, the Quicksizer tool calculates CPU, memory, and disk space resources which depend on the message throughput, the message size, and your specific business scenarios. The goal of using the tool is rather to get a rough idea about the size of the system required to run the estimated workload than providing exact numbers. Sizing is considered to be an iterative process. Having the Quicksizer results as a starting point, you have to continuously run benchmarks, and monitor your system to guarantee a healthy system.

With Quicksizer Version 12, as of June 2008, the PI sizing questionnaire has been revised in order to improve usability. Furthermore, the calculation has been adapted taking into account new features that were introduced with latest Support Packages of SAP NetWeaver PI 7.0 and with SAP NetWeaver PI 7.1, e.g., packaging, local processing in advanced adapter engine (latter applies for SAP NetWeaver PI 7.1 and beyond only), etc.


For guidelines about how to use the Quicksizer tool to size SAP XI 3.0 and SAP NetWeaver PI 7.0 systems, please refer to the Quicksizing Exchange Infrastructure 3.0 webinar on SDN: https://www.sdn.sap.com/irj/sdn/webinar?rid=/library/uuid/910148f1-0901-0010-ba91-b89aead41e81.

For guidelines about how to use the Quicksizer tool to size SAP NetWeaver PI 7.1 (including EHP 1) systems, please refer to the Quicksizing SAP NetWeaver PI 7.1 webinar on SDN: https://www.sdn.sap.com/irj/sdn/webinar?rid=/library/uuid/70e8b684-9f23-2b10-74bc-fdbc84ead706.

Once sizing has been completed, you can access SAP Standard Application Benchmarks results at www.sap.com/benchmark to find the appropriate hardware configuration provided by your hardware partner.
3. Performance Tuning

3.1 Preliminary Considerations

Tuning Impact
The tuning of SAP NetWeaver PI can lead to
- a decrease of the overall message processing time or/and
- an increase of the message throughput, i.e. the number of messages processed within a specific time frame

Depending on your business requirement, you may tackle either one of the factors or both.

For synchronous scenarios, usually it is more important to reduce the message response time. Especially in case of queries, end users expect fast request/response cycles. But even for synchronous scenarios where end users are not directly involved, such as lookups, a fast message processing time might be crucial, e.g. to avoid running into time-outs.

An increase of the message throughput usually is more related to asynchronous scenarios. Since messages are processed asynchronously, the processing time is less important. The main goal is to process as many messages as possible within a given time frame.

Note
Increasing the message throughput not necessarily directly leads to an increase of the message processing time, however for a high load situation a more stable system supposed to speed up processing time in the end.

Best Message Size / Message Size Impact on Performance

The figure below shows the impact of the message size on the message throughput. The shape of the graph is what you typically observe when measuring the message throughput for different message sizes. The actual number for the throughput highly depends on your hardware specifications. However it has been empirically observed that the message size range is more or less the same independent of the hardware used.

For smaller messages, the throughput is low. With increasing message size, the throughput increases as well until it reaches an upper limit where no more throughput improvements can be gained, it even starts to decrease when message size becomes too large.

So, we can distinguish 3 areas where message processing behaves differently:
- Small message sizes: Each message requires an overhead which is spent for context switches, header processing, DB operations, loading the respective program into main memory, etc. Since this overhead is more or less constant, the smaller the message, the larger the percentage time spent for this part of the overall message processing time.
- Best performing message sizes: It has been observed that the best choice is an average message size in the range of 1 MB to 5 MB, even slightly lower message sizes still lead to an acceptable performance.
- Large message sizes: The larger the message, the larger is the main memory resource consumption. Since multiple messages are processed in parallel, for larger messages you may reach the limits of your total main memory leading to swapping, or even memory overflow. Furthermore, Java virtual machine garbage collection times may also impact the overall performance even leading to a decline of the message throughput.
**Recommendation**

- Use reasonable message sizes to improve performance, to avoid memory overflows and to increase overall system stability.
- At design time, you should consider that the message throughput is much higher for larger messages due to the necessary processing overhead for a single message. On the other side, the memory consumption is higher for processing larger messages.
- The best practice is to keep the average message size in the range of 1 MB to 5 MB, see performance improvement mechanisms below.

**Performance Improvement**

As seen before, tuning can have an impact on message throughput or message processing time. Usually, the tuning mechanisms for those both factors differ.

Speeding up message processing can be achieved by switching to a more powerful hardware, and faster CPU. Another possibility would be to omit intermediaries if possible at all, and by doing so reduce the number of processing steps and persistency steps.

Increasing the message throughput usually can be achieved by increasing the degree of parallel processing, let it be by scaling up your system while adding additional Application Servers or Java Engines, or by increasing the number of work processes and threads, respectively.

As discussed in chapter 3.1, the optimum throughput is given if the average message size lies within a specific range. So, an approach to improve the message throughput would be to shift the actual message size into the best performing message size range. This can be done either by packaging or splitting mechanisms depending on whether your original average message size is too small or too large.
To sum up, the following mechanisms to improve system performance can be distinguished:

- Move to a more powerful hardware, and faster CPU
- Scaling (parallel processing), see chapter 3.2.1:
  - Add further Application Servers
  - Add further Java Engine nodes
- Tuning of Integration Server (ABAP), see chapter 3.2.1:
  - Increase number of dialog work processes (parallel processing)
  - ICM tuning
  - Increase number of queues (parallel processing)
- Tuning of Adapter Engine/Messaging System (Java), see chapter 3.2.1:
  - Increase number of threads for each adapter type (parallel processing)
  - Use parallel connections depending on adapter type (e.g., maximum connections for RFC adapter, maximum concurrency for File and JDBC receiver adapter)
  - Use multiple communication channels (e.g., parallel polling for file adapter)
- Packaging
  - Packaging on Sender side
  - Message Packaging in general, see chapter 3.2.4
  - IDoc Sender Packaging, see chapter 3.2.9.1
- Message splitting, e.g., for file sender, see chapter 3.2.9.2
• Omit intermediaries
  o Move from mediated scenario to Point-to-Point, see chapter 3.2.2
  o Local processing in Advanced Adapter Engine bypassing Integration Server, see chapter 3.2.3
  o For SOAP: bypass Adapter Engine by directly sending request to Integration Server pipeline, see chapter 3.2.9.3
  o For ABAP proxy connectivity via local AAE processing, see chapter 3.2.9.4
• Increase throughput for core scenarios only via message prioritization, see chapter 3.2.6
• Reduce message load during peak hours by moving non-time critical scenarios to off-peak hours via time-controlled message processing, see chapter 3.2.7

3.2 Tuning Mechanisms

3.2.1 Scaling & Tuning of IS/AE

Best practices and guidelines for scaling and tuning SAP NetWeaver PI in general is covered in several documents available on SAP Service Marketplace, alias instguides, navigate to Operations: http://service.sap.com/instguides (SMP logon required).

  • PI Performance Check, see also SAP Note 894509
  • PI Tuning Guide
  • PI Troubleshooting Guide, see also SAP Note 806546

SAP Safeguarding offers service packages and tools to stabilize, optimize, and improve the overall performance of your system landscape. For details, please refer to the SAP Service Marketplace, alias safeguarding: http://service.sap.com/safeguarding (SMP logon required).

For guidelines about how to analyze the overall performance of your SAP NetWeaver PI system and how to tackle performance bottlenecks, please refer to the SAP NetWeaver XI Performance Analysis webinar on SDN: https://www.sdn.sap.com/irj/sdn/webinar?rid=/library/uuid/defd5544-0c01-0010-ba88-fd38caee02f7.

⚠️ Note

Tuning guides are only provided for SA PXI 3.0 and SAP NetWeaver PI 7.0. For SAP NetWeaver PI 7.1 (including EHP 1), they have been replaced by the zero administration concept providing dynamic system configuration and built-in configuration templates. For more details, please refer to SAP Help Portal http://help.sap.com, and navigate to SAP NetWeaver → SAP NetWeaver PI/Mobile/IdM 7.1 → SAP NetWeaver Process Integration 7.1 Including Enhancement Package 1 → SAP NetWeaver Process Integration Library → Administrator's Guide → Technical Operations for SAP NetWeaver → Administration of SAP NetWeaver Systems → AS Java (Application Server Java) → Technical System Landscape → Architecture of AS Java → Zero Administration.

⚠️ Note

When tuning your system by increasing the number of dialog processes, queues, or threads, please ensure that enough hardware resources in terms of CPU and memory are available. Tune carefully, i.e., always keep in mind that tuning one stack might affect the other stack.
3.2.2 Point-to-Point

In a mediated scenario messages are sent from sender systems to receiver systems via SAP NetWeaver PI as middleware software providing process integration services like routing, mapping, and a business process engine. The systems that exchange messages using SAP NetWeaver PI are separated from each other which makes it more flexible to integrate systems of different technologies. Furthermore, using SAP NetWeaver PI allows you to centrally configure, administer, and monitor your communications.

For Web Services/Enterprise Services scenarios you may switch from a mediated scenario to a Point-to-Point connection, depending on your business requirements. In a Point-to-Point scenario, messages are sent directly from a sender system to a receiver system. If you move from a mediated scenario to a Point-to-Point connection, you will lose the advantages introduced with the loose coupling of the middleware solution. Furthermore, by bypassing SAP NetWeaver PI during runtime, you can not use the functionalities and services provided by SAP NetWeaver PI, such as mappings, dynamic receiver determinations, ccBPM, sending one message to multiple systems, etc.

However, the performance will increase due to the fact that you eliminate the message processing within the middleware software. This approach mainly leads to a speed-up of the message processing, however since less resources are required it can also improve the message throughput.

Although SAP NetWeaver PI is bypassed during runtime, you still can use SAP NetWeaver PI to design your service interfaces, and to configure and monitor the Point-to-Point communications. Even though this tuning approach does not directly impact SAP NetWeaver PI's runtime behavior, it exploits all SAP NetWeaver PI capabilities at design time, at configuration time, and for central monitoring. The message processing is optimized through de-centralized runtime but you keep control through centralized configuration.

For SAP NetWeaver PI 7.1 (including EHP 1), the configuration of Point-to-Point communications has been made easier with the so called Direct Connection object. Direct Connection applies to ABAP Web Services using WS-RM protocol. Configuration is done centrally in the Integration Directory.
Recommendation

For Web Service scenarios it makes sense from a performance point of view to switch from a mediated scenario to a Point-to-Point connection if middleware functionalities are not required. So, this approach applies to the following implementation scenario:

- Web service
- No mapping
- No dynamic, content-based routing
- One receiver only

3.2.3 Local Processing in AAE

For SAP NetWeaver PI 7.1 and beyond, the Adapter Engine is deployed as a so-called Advanced Adapter Engine (AAE). In addition to Adapter Engine capabilities of previous releases, the AAE supports message processing without invoking PI’s Integration Engine if both sender and receiver adapter are located on the same Adapter Engine. In this case, mapping and routing are carried out on the Adapter Engine. You configure the same by maintaining the so-called Integrated Configuration Object in the Integration Directory.


Note

With SAP NetWeaver PI 7.1, local AAE processing functionality using the Integrated Configuration Object is limited, e.g., no content-based routing, no receiver splits, no 1:n message mapping, etc.
Note
Starting with SAP NetWeaver PI 7.1 EHP 1, local AAE processing is enhanced to include support for content-based routing and receiver splits. Further enhancements such as interface and mapping splits are planned for the next major release of SAP NetWeaver PI.

Since the Integration Engine is bypassed for local message processing in the AAE, the resource consumption both in memory and CPU is lower. This leads to higher message throughput, and faster response times which especially is important for synchronous scenarios.

Tip
ABAP based adapters such as the plain-http adapter, IDoc adapter, and the XI adapter (ABAP proxies) are not supported when choosing local message processing since they reside on the Integration Engine. However, plain-http adapter can be implemented using SOAP adapter instead; XI adapter can be implemented on project basis with SOAP adapter in Axis mode.

Starting with SAP NetWeaver PI 7.1 EHP 1, the SOAP adapter supports communication with ABAP proxies (i.e., via the XI 3.0 protocol). However, some limitations may exist for specific proxy based scenarios. For the latest status on these limitations, please check SAP Note 1247043.

Recommendation
Use Local Processing in the AAE in the following cases:
- Restricted to adapter to adapter based scenarios
- No integration processes required
- Both supported, asynchronous and synchronous communication
- Leads to message processing speed and throughput improvements
- Especially suited for synchronous calls

3.2.4 Message Packaging
As of SAP NetWeaver PI 7.0 SPS13 and SAP NetWeaver PI 7.1, a generic message packaging functionality has been introduced. Several messages within a message queue are grouped together in packages, and processed within one LUW. By processing multiple messages in packages instead of individually, the overall overhead can be reduced leading to less hardware resources consumption and an increased message throughput:
- The respective programs required for message processing are only loaded once into main memory for each package
- Multiple messages are processed in one dialog work process
- Only one DB commit is carried out for the complete package
- Only one logon is required when switching from ABAP stack to Java stack or vice versa

Both are supported, asynchronous messages with quality of service exactly once and exactly once in order for following steps:
- Sender and receiver proxies

November 2009
• Pipeline steps within Integration Engine, e.g., receiver determination, mapping, etc
• IDoc adapter on receiver side
• Adapter-Engine-based adapters on receiver side
• Integration processes within the Business Process Engine (BPE)

For more details about the message packaging functionality, please refer to SAP Help Portal http://help.sap.com, also check SAP Notes 1037176 and 1058623.

For SAP NetWeaver PI 7.0, navigate to SAP NetWeaver 7.0 → SAP NetWeaver 7.0 Library → SAP NetWeaver Library → SAP NetWeaver by Key Capability → Process Integration by Key Capability → SAP NetWeaver Exchange Infrastructure → Runtime → Integration Engine → Message Packaging.

For SAP NetWeaver PI 7.1 (including EHP 1), navigate to SAP NetWeaver → SAP NetWeaver PI/Mobile/IdM 7.1 → SAP NetWeaver Process Integration 7.1 Including Enhancement Package 1 → SAP NetWeaver Process Integration Library → Function-Oriented View → Process Integration → Integration Engine → Message Packaging.

For an introduction to message packaging, please refer to the Message Packaging with Process Integration component of SAP NetWeaver 7.0 webinar on SDN: https://www.sdn.sap.com/irj/sdn/webinar?rid=/library/uuid/90926057-cd00-2a10-c1b0-ef8474aad03b0.

Note
For IDoc scenarios, use the new message packaging instead of the IDoc message packages (transaction code IDXPW) from previous releases since the new message packaging approach won't lead to any message processing delay. Message packaging with IDocs in this context is not to be confused with the new IDoc package processing capability available in the IDoc sender adapter starting from SAP NetWeaver PI 7.0 EHP 1 and 7.1 EHP 1 (see section 3.2.9.1).

Recommendation
Use message Packaging in the Integration Server in the following cases:
• Intended for small asynchronous messages
• Best results when using proxies
• Leads to throughput improvements

Package Configuration Types and Parameters
Package Configuration Types define the way how packages are created either in general or for specific service interfaces. You can maintain them running transaction code SXMS_BCM.

Change View "Package Configuration Types": Overview

<table>
<thead>
<tr>
<th>Package Configuration Type</th>
<th>Description</th>
<th>Wait Time</th>
<th>Number</th>
<th>Maximum Package Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUSTOM</td>
<td>customized</td>
<td>10</td>
<td>1000</td>
<td>2,000</td>
</tr>
<tr>
<td>DEFAULT</td>
<td>Default Package Configuration Type</td>
<td>100</td>
<td>1,000</td>
<td></td>
</tr>
<tr>
<td>N0</td>
<td>No Packages</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
By default, a standard configuration type is delivered which applies to all scenarios once message packaging has been activated unless you define and assign your own configuration type.

**Note**

For SAP NetWeaver PI 7.0, you have to explicitly activate message packaging by calling transaction code SXMB_ADM, Integration Engine Configuration. Maintain parameter PACKAGING of category RUNTIME. For SAP NetWeaver PI 7.1 (including EHP 1), message packaging is activated by default.

For each Package Configuration Type, you have to maintain three parameters:

- **Number**: maximum number of messages per package (default: 100)
- **Maximum package Size in KB**: (default: 1000)
- **Wait Time**: specifies how long the message processing should wait if the maximum number of messages criteria has not met (default: 0)

Calling transaction code SXMS_BCONF, you can change either global or specific package configuration, i.e. you can assign your own package configuration type to the respective pipeline steps, overwriting the default setting.

If your packaging parameters should apply to specific scenarios only, you can maintain the specific package configuration. Here, you have to filter on your specific scenario by maintaining the Sender/Receiver ID that you have defined beforehand in transaction code SXMSIF.

To be able to monitor your packages, you first have to switch on the package statistics running transaction code XMSPKSTATCNF.
Once done, you can display the package information in transaction code \textit{XMSPKSTATMON}. For each queue, you get provided the number of packages created, the average number of messages per package, the average package size in KB, and the number of timeouts. Latter refers to the wait time, i.e. it indicates that the maximum number of messages criteria has not met, and hence the message processing ran into a halt.

Packaging for the Business Process Engine has to be first activated globally in transaction code \textit{SXMB_ADM}. Maintain parameter \textit{PACKAGING} of category \textit{BPE}. Once done, packages are sent to and from the BPE. Packaging within the BPE has to be activated for each integration process individually because the optimum choice of packaging parameters highly depends on the specific integration process. This is done in the \textit{BPE Inbound Processing}, transaction code \textit{SWF_INB_CONF}.
**Recommendation**

Usually, the default settings are a good choice for most scenarios. You can adapt them to fit best your specific message profile. As a rule of thumb, choose your parameters in such a way that the resulting packages shift into the best performing message size area, as discussed in chapter 3.1.

**Example**

If your messages are 10KB in average size, a maximum number of messages per package of 100 is a good choice resulting in packages of approx. 1MB size. If the average message size is even lower, you should increase the number accordingly.

**Recommendation**

Monitor your packages regularly to optimize message packaging and hence overall performance. Usually, you rely on the number of packages, the average number of messages per package as well as the average package size.

**Example**

If you have too many small packages where both criteria are not met, the maximum number of messages and the maximum package size, you should increase the wait time to collect more messages within each package. This further lessens the hardware resources consumption, and increases the overall throughput. However, it could also be that the small packages are a result of too many parallel queues. So, in this case you should reduce the number of parallel queues, see also below.

If you have too many small packages in terms of number of messages however the maximum package size has been exceeded, you should increase the latter to collect more messages within each package.

**Recommendation**

The creation of packages is lot easier if more messages of the same size are processed in one queue. If possible, try to configure your queues accordingly. For each queue or scenario you can maintain interface-specific package configuration with an optimum set of parameters for your different scenarios to optimize message packaging.

**Note**

Please note that for principal propagation, messages with different user credentials cannot be processed in the same package. This might impact the message packaging efficiency.
Benchmark

The benchmark results listed in table below were taken on hardware with roughly 25,000 SAPS (SAP Application Performance Standard is a hardware-independent performance unit, see http://www.sap.com/solutions/benchmark/measuring). For this test case, messages were sent in IDoc format from an SAP ECC system to SAP NetWeaver PI, converted into XML format by the IDoc adapter, and finally stored as comma-separated files (csv) using the file adapter of the Adapter Engine. Measurements with different message sizes were carried out, and compared to estimations derived from Quicksizer tool.

<table>
<thead>
<tr>
<th>Message Size</th>
<th>11 KB</th>
<th>32 KB</th>
<th>245 KB</th>
<th>2.37 MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corresponding number of line items</td>
<td>2</td>
<td>10</td>
<td>100</td>
<td>1000</td>
</tr>
<tr>
<td>Expected number of messages with 66% system usage (as calculated by Quicksizer)</td>
<td>200,000 m/h</td>
<td>180,000 m/h</td>
<td>110,000 m/h</td>
<td>21,500 m/h</td>
</tr>
<tr>
<td>~ 56 m/s</td>
<td>~ 50 m/s</td>
<td>~ 31 m/s</td>
<td>~ 6 m/s</td>
<td></td>
</tr>
<tr>
<td>Extrapolated estimated number of messages with 100% system usage (without message packaging)</td>
<td>300,000 m/h</td>
<td>270,000 m/h</td>
<td>155,000 m/h</td>
<td>32,000 m/h</td>
</tr>
<tr>
<td>~84 m/s</td>
<td>~75 m/s</td>
<td>~66 m/s</td>
<td>~9 m/s</td>
<td></td>
</tr>
<tr>
<td>Number of messages with message packaging (max. 100 messages, max. package size of 1MB, wait time of 10 sec.)</td>
<td>790,000 m/h</td>
<td>630,000 m/h</td>
<td>207,000 m/h</td>
<td></td>
</tr>
<tr>
<td>~220 m/s</td>
<td>~175 m/s</td>
<td>~58 m/s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>~8.7 GB/h</td>
<td>~20.2 GB/h</td>
<td>~50.7 GB/h</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of messages with message packaging (max. 100 messages, max. package size of 5MB, wait time of 10 sec.)</td>
<td>33,000 m/h</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>~9 m/s</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Improvement factor: number with message packaging compared to estimated values from Quicksizer</td>
<td>2.6</td>
<td>2.33</td>
<td>1.34</td>
<td></td>
</tr>
</tbody>
</table>

The following recommendations are more or less a result of the benchmarks carried out.

**Recommendation**

Packages are created for messages within the same queue. If the backlog of messages within each queue is high enough, sufficiently large packages are created to keep performance high. If you use too many parallel queues so that each queue only contains a few number of messages, it might be that the packages are too small resulting in limited performance improvement. It is recommended to use less parallel queues to guarantee a sufficiently high number of messages within each queue. Note that scheduler runs more efficient having fewer queues which are evenly and fully used.

**Recommendation**

For large hardware resources where queues usually are processed quickly, it might be that the packages become too small. In this case, increase the wait time of message
packaging, e.g. from 0 to 10 seconds, in order to obtain larger packages. This was actually the case in the benchmark, and hence the wait time was set to 10 seconds.

The same is valid for low load case scenarios (load in terms of number of messages in the queues within a specific time frame). Every time the scheduler reads the messages from a queue, there are not enough messages available to create an optimum sized package. Here, you can also introduce a wait time to allow more messages to pile up in the queue, so that larger packages are created. By doing so, HW resources can be freed up for other scenarios resulting in a better usage of your overall system resources.

This increases the overall throughput, and decreases CPU and DB usage. However, keep in mind that the average runtime of each single message will increase.

**Recommendation**

As can be seen from table above, best performance improvements are gained for small messages. However, the highest throughput in terms of message size per time is gained for larger messages. So, message packaging is especially intended to improve performance for small asynchronous messages. However, if possible it is recommended to package already on sender side since this leads to highest throughput numbers.

**Recommendation**

For larger message sizes, the maximum package size should be increased to ensure that the created packages contain a sufficient number of messages. As a prerequisite, your hardware resources should be large enough to be able to handle such large packages.

### 3.2.5 Message Packaging versus Local Processing in AAE and Point-to-Point

The performance Improvement methods which were introduced above are mutually exclusive, i.e. we can not apply all techniques at the same time. Packaging runs in ABAP, while Local Processing purely runs in the AAE.

Furthermore, the different approaches also apply to different scenarios. Packaging is mainly intended for small asynchronous messages. Local Processing in the AAE is used for scenarios where only AE based adapters are required, and which do not need any integration process functionality. Although it supports both asynchronous and synchronous communication, it is best suited for synchronous calls since it mainly improves message processing time. Finally, Point-to-Point connections are restricted to Web Services, i.e. the SAP NetWeaver PI 7.1 introduced object Direct Connection is restricted to ABAP Web Services using WS-RM protocol.

**Recommendation**

If both options, packaging and local processing in AAE, apply to your particular scenario, from a performance point of view you should opt for the latter one. From the benchmarks carried out so far, the local processing in AAE led to a higher performance improvement compared to packaging. With packaging, a maximum performance boost of factor 3 could be achieved, whereas for local processing in AAE the factor was up to 10. However, please take into account that actual performance improvement numbers do depend on your specific scenario and protocols used.
3.2.6 Message Prioritization

You can increase the message throughput for your core scenarios by maintaining message prioritization, i.e. you can assign more system resources to those specific scenarios only. This can be done either in the Integration Engine or the Advanced Adapter Engine. Latter is supported as of SAP NetWeaver PI 7.1 only.


For SAP NetWeaver PI 7.0, navigate to SAP NetWeaver 7.0 → SAP NetWeaver 7.0 Library → SAP NetWeaver Library → SAP NetWeaver by Key Capability → Process Integration by Key Capability → SAP NetWeaver Exchange Infrastructure → Runtime → Integration Engine → Prioritized Message Processing.

For SAP NetWeaver PI 7.1 (including EHP 1), navigate to SAP NetWeaver → SAP NetWeaver PI/Mobile/IdM 7.1 → SAP NetWeaver Process Integration 7.1 Including Enhancement Package 1 → SAP NetWeaver Process Integration Library → Function-Oriented View → Process Integration → Integration Engine → Prioritized Message Processing.


3.2.7 Time-controlled message processing

Another means to increase your overall throughput, especially for your core scenarios, is to better balance message load across the day. This can be achieved by event-driven message processing where you stop message processing of scenarios of lower priority, and postpone processing of the same to non-peak time frame, hence you only process time-critical scenarios during peak hours.


For SAP NetWeaver PI 7.0, navigate to SAP NetWeaver 7.0 → SAP NetWeaver 7.0 Library → SAP NetWeaver Library → SAP NetWeaver by Key Capability → Process Integration by Key Capability → SAP NetWeaver Exchange Infrastructure → Runtime → Integration Engine → Event-Driven Message Processing.

For SAP NetWeaver PI 7.1 (including EHP 1), navigate to SAP NetWeaver → SAP NetWeaver PI/Mobile/IdM 7.1 → SAP NetWeaver Process Integration 7.1 Including Enhancement Package 1 → SAP NetWeaver Process Integration Library → Function-Oriented View → Process Integration → Integration Engine → Event-Driven Message Processing.
3.2.8 Tuning of BPE

Besides the above described message packaging, for integration processes, the following tuning mechanism can be applied:

- Inbound processing without buffering
- Use of one configurable queue which can be prioritized or assigned to a dedicated server
- Use of multiple inbound queues
- Configure transactional behavior of ccBPM process steps to reduce number of persistency steps

The mechanisms are described in a how-to guide series, accessible on SDN, alias howtoguides: https://www.sdn.sap.com/irj/sdn/howtoguides. Navigate to SAP NetWeaver 7.0 How-to Guides → End-to-End Process Integration → Business Process Management:

- How to Configure Inbound Processing in ccBPM Part I: Delivery Mode
- How to Configure Inbound Processing in ccBPM Part II: Queue Assignment
- How to Configure ccBPM Runtime Part III: Transactional Behavior of an Integration Process


3.2.9 Special Cases

3.2.9.1 IDoc

IDoc Packages on PI Inbound (Sender side)

Instead of immediately sending each IDoc message, you can collect them in the sender system and then send them as a package to SAP NetWeaver PI. Only one tRFC call is needed to transfer multiple IDocs leading to an improved performance.

Note

In releases prior to SAP NetWeaver PI 7.0 EHP 1 and SAP NetWeaver PI 7.1 EHP 1, if IDocs are sent as packages to SAP NetWeaver PI, the packages are not kept but split into individual messages by the IDoc adapter.

Another option is that the sending system collects IDoc messages into one file using the XML Port. Instead of using the IDoc adapter, the file is sent as a single message via the File adapter to SAP NetWeaver PI. In this case, the IDoc package is not split into individual messages at PI inbound. As a prerequisite, the IDoc schema has to be adapted in the ESR, see also SAP Note 814393.

For more details, please refer to SAP Help Portal http://help.sap.com, and navigate to SAP NetWeaver 7.0 → SAP NetWeaver 7.0 Library → SAP NetWeaver Library → SAP NetWeaver by Key Capability → Application Platform by Key Capability → Platform-Wide Services → Connectivity → Components of SAP Communication Technology → Classical SAP Technologies (ABAP) → IDoc Interface/ALE.
IDoc Package Processing via IDoc Sender Adapter (from 7.0 EHP 1 & 7.1 EHP 1)

Starting with SAP NetWeaver PI release 7.1 EHP 1, IDoc packages sent to the IDoc adapter at the Integration Server inbound channel (sender IDoc adapter) can be processed as a complete package. In previous releases, packages were broken up into individual PI messages according to the number of IDocs contained in the IDoc package. So, if the scenario called for the receiver system to receive the IDocs as a package, some additional collection mechanism (e.g., using an Integration Process) within the Integration Server was necessary and this could add significant message processing overhead. The new IDoc package processing feature processes a package directly (or splits the package into smaller packages if so configured). It also eliminates the need for the sender SAP system to use the XML port and send the package as a file to the PI file adapter as described above.

Overall, the new IDoc package processing feature results in a reduced number of messages, less acknowledgements, better alignment with backends, simplified processing and monitoring, and, ultimately, improved performance.

Note

If mapping of a PI message containing an IDoc package is required, the imported IDoc schema must be adjusted as the IDoc tag in the schema only allows for one IDoc instance (see SAP Note 814393 for details). For more information on IDoc packaging using the sender IDoc adapter, please refer to the SAP Help Portal http://help.sap.com and navigate to SAP NetWeaver → SAP NetWeaver PI/Mobile/IdM 7.1 → SAP NetWeaver Process Integration 7.1 Including Enhancement Package 1 → SAP NetWeaver Process Integration Library → Function-Oriented View → Process Integration → Integration Directory → Defining Collaboration Profiles → Defining Communication Channels → Configuring the IDoc Adapter → Configuring the Sender IDoc Adapter.

IDoc Packages on PI Outbound (Receiver side)

You can collect IDocs in the Integration Engine outbound, and send to a receiver as a package, calling transaction code IDXPW. This functionality is more or less based on the event-driven message processing, i.e. the messages are stopped, collected, and sent with a delay. In this way, for all messages within the package only one RFC connection to the receiver system has to be established by the IDoc adapter. This saves system resource, and reduces overhead caused by setting up RFC connections.

For more details, please refer to SAP Help Portal http://help.sap.com:

For SAP NetWeaver PI 7.0, navigate to SAP NetWeaver 7.0 → SAP NetWeaver 7.0 Library → SAP NetWeaver Library → SAP NetWeaver by Key Capability → Process Integration by Key Capability → SAP NetWeaver Exchange Infrastructure → Runtime → Integration Engine → Event-Driven Message Processing → Activating IDoc Message Packages.

For SAP NetWeaver PI 7.1, navigate to SAP NetWeaver → SAP NetWeaver PI/Mobile/IdM 7.1 → SAP NetWeaver Process Integration 7.1 Including Enhancement Package 1 → SAP NetWeaver Process Integration Library → Function-Oriented View → Process Integration → Integration Engine → Event-Driven Message Processing → Activating IDoc Message Packages.

Recommendation

As mentioned above, rather use the generic message packaging functionality than the IDoc message packaging since former does not lead to a delay. Furthermore, IDoc message packaging only applies to the Integration Engine outbound whereas the generic message packaging applies to both inbound and outbound.
IDoc tunneling

The IDoc Adapter converts an IDoc message into XML format. Optionally, you can specify that IDocs are kept and transferred in table format. To change the default setting, call transaction code SXMB_ADM, Integration Engine Configuration, and maintain parameter XML_CONVERSION of category IDOC accordingly.

**Recommendation**

If you neither need mapping nor content-based routing, you can prevent the IDoc to be converted into IDoc-XML. Of course, this only makes sense if both your sender and receiver expect the message in IDoc format. This leads to an improved system performance.

### 3.2.9.2 Large File Handling

In transaction code SXMB_ADM, Integration Engine Configuration, you can maintain parameter EO_MSG_SIZE_LIMIT of category TUNING to process large messages in series. This applies to any kind of messages however especially files usually exceed the best performing message size, and hence this is mentioned here. Once set, all messages exceeding the specified value are processed in series in a separate message queue.

**Recommendation**

Maintain the parameter in order to avoid that parallel processing of multiple large messages exceeds main memory resources. Furthermore, since large messages are processed in a separate queue the processing of smaller messages won't be affected or even blocked.

For large text files containing multiple records, you can split the same into multiple messages in the file/ftp adapter. This applies when File Content Conversion mode is chosen. In the communication channel, you have to maintain parameter Recordsets per Message.


### 3.2.9.3 SOAP

Instead of using the SOAP sender adapter, you can directly send a SOAP request to the entry point of the Integration Engine pipeline bypassing the Adapter Engine. As a prerequisite, the URL/query string should include the XI parameters like in the example below:

```
```

**Note**

This option can not be applied if the message contains more than one attachment. Furthermore, you can not use modules.

**Recommendation**

You may apply this approach only if the receiver adapter resides on the Integration Server, e.g. for XI proxy. Otherwise if the receiver adapter resides on the same Adapter Engine, using the local processing of the AAE should lead to a higher performance improvement.
3.2.9.4 Proxy Connectivity via SOAP Adapter

Starting with SAP NetWeaver PI 7.1 EHP 1, the XI 3.0 message protocol is supported in the SOAP adapter for the communication between an Advanced Adapter Engine (AAE) version 7.1 EHP 1 and ABAP proxies generated in the SAP NetWeaver Process Integration versions 3.0, 7.0, 7.1, 7.1 EHP 1, Advanced Adapter Engine version 7.1 EHP 1, and SAP Partner Connectivity Kit version 7.1 EHP 1. This means that suitable proxy scenarios can take advantage of the significant increase in performance realized by using local processing of the AAE.

Note

Some limitations may exist for specific proxy based scenarios. For the latest status on these limitations, please check SAP Note 1247043.

For more information on this new feature, please check the SAP NetWeaver PI 7.1 EHP 1 Service Bus release notes on the SAP Help Portal http://help.sap.com.