Real-time Session Performance
Overview

This article provides information about real-time session performance and throughput. It also provides recommendations on how you can optimize a real-time session. The recommendations are based upon performance tests that Informatica ran on real-time sessions. It includes the following topics:

- Key factors that influence performance
- Performance test
- Conclusion

Key Factors that Influence Performance

Real-time sessions involve a complicated set of components, such as source message-oriented middleware system, target message-oriented middleware system, source database, and target database. All of these components affect the overall session performance. Performance is usually a combination of factors based on the scenario.

The following key factors can influence throughput:

- **Message size.** Message size influences the time it takes for data to get in and out of PowerCenter. Larger messages increase performance.

- **Message complexity.** Complex messages increase performance. The more complex the message, the longer processing takes.

- **Flush latency and commit interval.** Smaller values of flush latency or commit interval lead to more frequent commits on the source and target systems and more activity not related to processing. A smaller commit interval is configured through a smaller source-based commit.

- **Connectivity.** (For WebSphere MQ, both native and through JMS). The use of client or server connectivity library and whether PowerCenter runs on the same machine as the message-oriented middleware server can influence performance.

  Generally, if PowerCenter runs on the same machine as the message-oriented middleware server, performance is optimized when you use the server connectivity library. For more information about how to enable the server library, see the [PowerExchange for WebSphere MQ User Guide](#). By default, the client connectivity library is used.

- **Partitions.** Performance depends on the number of partitions. Generally, performance increases with use of partitions up to a certain point. Then, performance may flatten. Tune the number of partitions to find the optimized session performance. If you configure multiple partitions, PowerCenter cannot guarantee message order.

- **Target type.** The type of target, such as flat file, relational database, or WebSphere MQ, and the target configuration can affect session performance. For example, performance can decrease if you use a shared storage directory for a flat file target. There are a number of tasks you can perform on each target type to optimize the performance. For more information, see the [PowerCenter Performance Tuning Guide](#).

- **Message-oriented middleware system configuration and tuning.** The configuration of the message-oriented middleware system can negatively impact performance. To improve performance, configure tuning parameters.

- **Mapping complexity.** Performance can decrease depending on how complex the mapping is. You can optimize the mapping if you reduce the number of transformations in the mapping and delete unnecessary links between transformations.

Performance Test

The following section describes sizing and tuning considerations based on the outcome of performance tests that Informatica ran with real-time sessions.
Environment

The following table describes the hardware and software used for the performance tests:

<table>
<thead>
<tr>
<th>Hardware or Software Detail</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PowerCenter version</td>
<td>PowerCenter Real Time Edition Version 8.6 AIX 64-bit</td>
</tr>
<tr>
<td>Operation System</td>
<td>AIX 5.3</td>
</tr>
<tr>
<td>Number of CPUs</td>
<td>16, Clock frequency: 1.9 GHz</td>
</tr>
<tr>
<td>Memory</td>
<td>32 GB</td>
</tr>
<tr>
<td>WebSphere MQ Server Version</td>
<td>6.0.0.0 and 6.0.2.3</td>
</tr>
<tr>
<td>Database</td>
<td>Oracle</td>
</tr>
</tbody>
</table>

PowerCenter and WebSphere MQ were on the same machine. There was no network overhead.

The test team used two different types of mappings. Both sessions had flush latency set to 1000 ms and source-based commit set to 50K.

The following table describes the mappings that were used for the performance tests:

<table>
<thead>
<tr>
<th>Type of Mapping</th>
<th>Mapping Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple mapping</td>
<td>The simple mapping included a WebSphere MQ source and relational target.</td>
</tr>
<tr>
<td>Complex mapping</td>
<td>The complex mapping included a WebSphere MQ source, XML Parser transformation, Lookup transformation, Expression transformation, and relational target.</td>
</tr>
</tbody>
</table>

Scalability

The test team observed non-linear scalability as they added partitions, but performance also depends on factors such as the message size and mapping complexity. There is a scalability issue with WebSphere MQ where WebSphere MQ does not scale when the session included more than three partitions. (The test team verified with standalone Java/C++ programs too.)

For larger messages of size 512 KB or above, the test team observed flat scalability.

The following figure shows how throughput is affected by increases in message size and number of partitions:

The following key factors affected scalability in this test:

- **Message size.** Larger messages increased performance and may have caused flat scalability.
• **Partitions.** Generally, performance increased with number of partitions. However, message size and mapping complexity can decrease performance even with partitions.

**Throughput**

The following observations were seen for WebSphere MQ reader and writer throughput:

- Throughput increased with increase in number of partitions. Throughput was flat beyond three partitions.
- Non-linear scalability with increase in number of partitions for WebSphere MQ and JMS readers.
- PowerCenter matched standalone Java/C++ apps throughput.
- In the performance setup, there was a sample Java program injecting messages of 4 KB size at the rate of 1213 msgs/sec into the WebSphere MQ system.
- PowerCenter matched the input rate and could have read and processed messages at the rate of 1208 msgs/sec.
- Messages were read in destructive mode from the source, meaning the messages were removed from the message queue source after reading and processing.

The following key factors affected throughput in this test:

- **Partitions.** Generally, performance increased with number of partitions. However, it leveled off beyond three partitions.
- **Connectivity.** If PowerCenter and WebSphere MQ Server were on separate machines, then throughput dropped.

**Throughput and Flush Latency**

For the simple mapping, throughput was around 4500+ messages with 1 KB messages. This was the best performance observed.

For the complex mapping, throughput was above 1000+ with 1 KB messages.

The best throughput was observed with a combination of a high source-based commit and a flush latency of 1000 ms. Throughput increased with increase in flush latency, but leveled off beyond 2000 ms.

The following key factors affected throughput and flush latency in this test:

- **Flush latency and commit interval.** Throughput increased with increase in flush latency and commit interval up to a certain threshold.
- **Mapping complexity.** Throughput decreased with the complexity of the mapping.

**Throughput and Source-Based Commit**

Commit type and commit interval played a significant role in determining throughput. With lower source-based commits, there were more frequent commits and hence lower throughputs were observed. For the simple mapping, throughput went as low as 212 msgs/sec when source-based commit is set to 1 and flush latency is set to 1000 ms.

With a higher value of source-based commit, such as 1 K, the test team saw higher throughputs, about 3500 messages.

To achieve the best possible latency, set source-based commit to 1.

The following table describes results of the performance tests for the complex mapping:

<table>
<thead>
<tr>
<th>Message Size</th>
<th>Source-Based Commit</th>
<th>Throughput (Msg/Sec)</th>
<th>Latency (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1KB</td>
<td>1</td>
<td>78</td>
<td>0.66</td>
</tr>
<tr>
<td>1KB</td>
<td>10</td>
<td>431</td>
<td>7.66</td>
</tr>
<tr>
<td>1KB</td>
<td>100</td>
<td>833</td>
<td>81.87</td>
</tr>
<tr>
<td>Message Size</td>
<td>Source-Based Commit</td>
<td>Throughput (Msg/Sec)</td>
<td>Latency (ms)</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------</td>
<td>----------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>1KB</td>
<td>500</td>
<td>1064</td>
<td>193.19</td>
</tr>
<tr>
<td>1KB</td>
<td>1000</td>
<td>1111</td>
<td>225.21</td>
</tr>
</tbody>
</table>

The following table describes the results of the performance tests for the simple mapping:

<table>
<thead>
<tr>
<th>Message Size</th>
<th>Source-Based Commit</th>
<th>Throughput (Msg/Sec)</th>
<th>Latency (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1KB</td>
<td>1</td>
<td>212</td>
<td>0.24</td>
</tr>
<tr>
<td>1KB</td>
<td>10</td>
<td>588</td>
<td>1.19</td>
</tr>
<tr>
<td>1KB</td>
<td>100</td>
<td>1818</td>
<td>4.38</td>
</tr>
<tr>
<td>1KB</td>
<td>500</td>
<td>3333</td>
<td>5.90</td>
</tr>
<tr>
<td>1KB</td>
<td>1000</td>
<td>3529</td>
<td>6.42</td>
</tr>
</tbody>
</table>

The following key factor affected throughput and source-based commit in this test:
- **Commit interval.** Throughput increased with increase in source-based commit.

**Throughput and Message Size**

For the simple and complex mappings, there was roughly one third reduction in throughput. This was observed as the message size increased from 1 KB to 4 KB and from 4 KB to 16 KB.

The following key factor affected throughput and message size in this test:
- **Message size.** Throughput increased as message size decreased.

**Conclusion**

Throughput is mainly affected by message size, number of partitions, flush latency, and commit interval. The following guidelines summarize how throughput is affected by these key factors in real-time processing:
- Throughput increases with increase in partitions, but throughput can flatten beyond three partitions. However, if the session contains large messages or complex mappings, then you may not see an increase in throughput as you increase partitions.
- Throughput increases as message size decreases.
- Throughput increases as message complexity decreases.
- Throughput increases as flush latency increases.
- Throughput increases as commit interval increases. Latency is optimized when source-based commit is set to 1.

**Authors**

Shankar Subramanian  
Director of Development

Jennifer Posivak  
Senior Technical Writer

Nagesh Anepally  
Quality Assurance Manager