WebSphere MQ File Transfer Edition for Linux V7.0.1

Performance Evaluations

Version 1.0

November 2009

Richard Cumbers

WebSphere MQ File Transfer
IBM UK Laboratories
Hursley Park
Winchester
Hampshire
SO21 2JN

Please take Note!

Before using this report, please be sure to read the paragraphs on “disclaimers”, “warranty and liability exclusion”, “errors and omissions”, and the other general information paragraphs in the "Notices" section below.

First Edition, November 2009.

This edition applies to WebSphere MQ File Transfer Edition for Linux V7.0.1 (and to all subsequent releases and modifications until otherwise indicated in new editions).

Note to U.S. Government Users
Documentation related to restricted rights.
Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule contract with IBM Corp

Notices

DISCLAIMERS

The performance data contained in this report were measured in a controlled environment. Results obtained in other environments may vary significantly.

You should not assume that the information contained in this report has been submitted to any formal testing by IBM.

Any use of this information and implementation of any of the techniques are the responsibility of the licensed user. Much depends on the ability of the licensed user to evaluate the data and to project the results into their own operational environment.

WARRANTY AND LIABILITY EXCLUSION

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore this statement may not apply to you.

In Germany and Austria, notwithstanding the above exclusions, IBM's warranty and liability are governed only by the respective terms applicable for Germany and Austria in the corresponding IBM program license agreement(s).

ERRORS AND OMISSIONS

The information set forth in this report could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; any such change will be incorporated in new editions of the information. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this information at any time and without notice.

INTENDED AUDIENCE

This report is intended for architects, systems programmers, analysts and programmers wanting to understand the performance characteristics of WebSphere MQ File Transfer Edition V7.0.1. The information is not intended as the specification of any programming interface that is provided by WebSphere. It is assumed that the reader is familiar with the concepts and operation of WebSphere MQ File Transfer Edition V7.

LOCAL AVAILABILITY

References in this report to IBM products or programs do not imply that IBM intends to make these available in all countries in which IBM operates. Consult your local IBM representative for information on the products and services currently available in your area.

ALTERNATIVE PRODUCTS AND SERVICES

Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user’s responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

USE OF INFORMATION PROVIDED BY YOU

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

TRADEMARKS AND SERVICE MARKS

The following terms used in this publication are trademarks of International Business Machines Corporation in the United States, other countries or both:

- IBM
- WebSphere
- DB2

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

EXPORT REGULATIONS

You agree to comply with all applicable export and import laws and regulations.

How this document is arranged

Performance Headlines

Pages: 2-21

Chapter 2 details the performance headlines for the two scenarios (client and bindings). Each scenario is detailed fully with diagrams in this section. The headline tests show how the Chunk Size property for an agent, and show the effect of transferring files as a group of transfers vs transferring files as a single transfer.

We detail the time taken for each transfer to complete, and the associated CPU utilisation for the hardware in use.

Tuning Recommendations

Pages: 22-24

Chapter 3 discusses the appropriate tuning that should be applied to both the WebSphere MQ network, and File Transfer Edition Agents.

Measurement Environment

Pages: 25-26

Chapter 4 gives an overview of the environment used to gather the performance results. This includes a detailed description of the hardware and software.

Contents

1 - Overview

WebSphere MQ File Transfer Edition is a managed file transfer product that uses WebSphere MQ as it's transport layer. This is the first performance report on Linux and so there is no comparison to make between versions.

This performance report details WebSphere MQ File Transfer Edition in a range of scenarios, giving the reader information on transfer times and CPU utilisation. The report is based on measurements taken from Intel hardware, running the Red Hat Enterprise Linux operating system.

2 - Performance Headlines

The measurements for the performance headlines are based on the time taken to transfer a set of files, and the associated CPU cost. A single performance measurement will use 1000MB worth of files, with the size of the files varying as follows:

- 1MB
- 10MB
- 100MB

To illustrate a typical test, if using a test is using a 1MB file then the test will transfer 1000 files in a single performance run. Varying the file size, but keeping the same overall MB transferred demonstrates the cost of the open and close file operations on transfer time and CPU usage.

The performance headlines demonstrate the effect of altering the agent's Chunk Size property. (See http://publib.boulder.ibm.com/infocenter/wmqfte/v7r0/index.jsp?topic=/com.ibm.wmqfte.admin.doc/properties.htm for more details on setting this property). The Chunk Size defines the size of the MQ message that the agent will use to transfer the files. The following Chunk Sizes (defined in bytes) have been used:

- 65536
- 131072
- 262144 (this is the agent's default value)
- 524288

To demonstrate the multi-threaded capability of the agent, a multiple transfer test was run and compared to a single transfer run. The multiple transfer test took divided the number of files transferred in the single transfer test by ten, and submitted them at the same time.

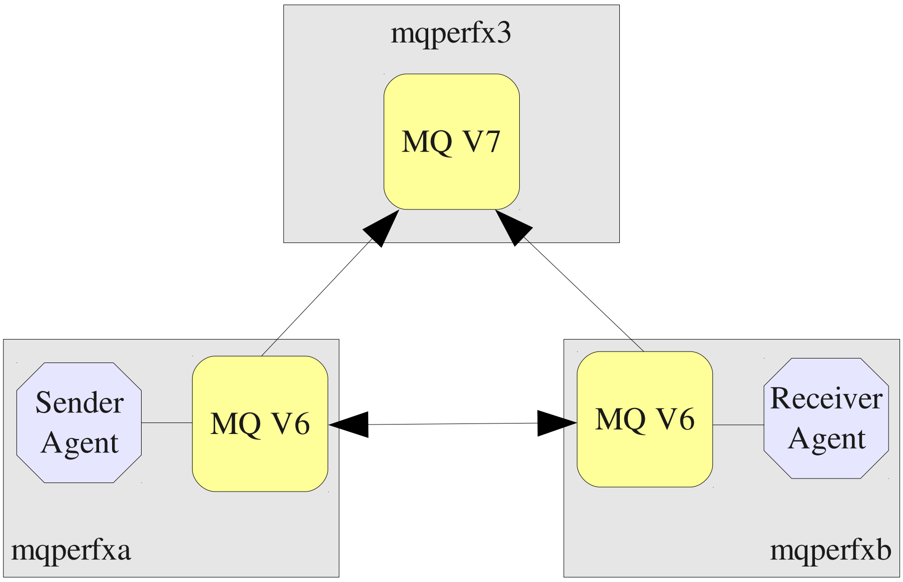
All files were transferred using text mode, as opposed to binary mode. Each file transferred was uniform in size for a given performance run, but contained random data. Transfers were submitted using the documented XML format which can be found in the samples directory of the WebSphere MQ File Transfer Edition Docs and Tools CD.

An agents queue manager was always WebSphere MQ Version 6, with the latest FixPack applied. For this performance report this equated to version 6.0.2.7. For the Coordination queue manager, the latest FixPack for Version 7.0.0 was used. For this performance report this equated to version 7.0.0.1

The results are laid out in the subsequent chapters. Each test case has its own results table, and associated graphs. The first set of tables and figures show the reader the effect that the chunk size (agentChunkSize) property has on the transfer time for a particular file size. These figures are then followed by a second set of tables and figures that compare the combinations of agent connectivity with the single/multiple transfer test at each of the Chunk Sizes. The second set of tables and figures serve to show the reader the difference between the transfer speeds and their associated CPU costs when using different agent connectivity options and single/multiple transfers.

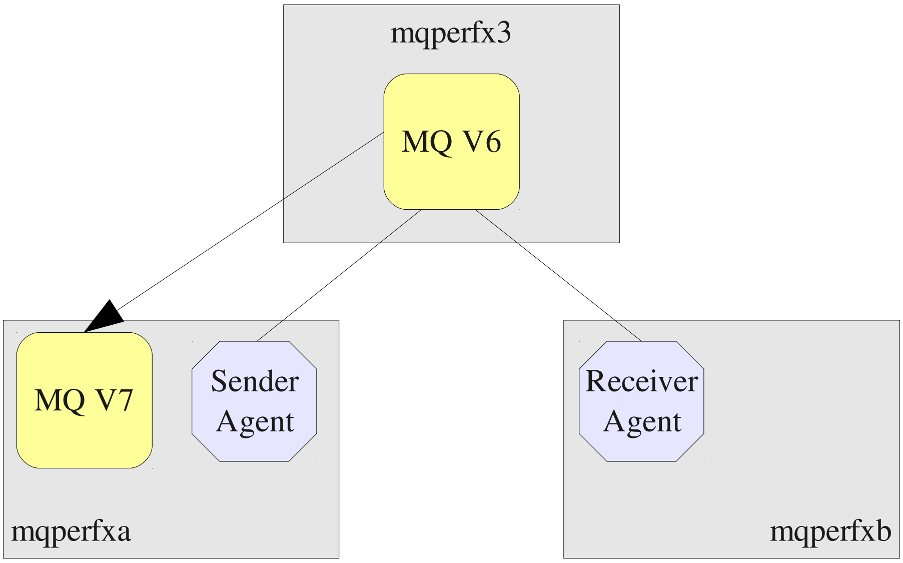
Agents Connecting in Bindings Mode

In this scenario, each agent is connected to a local queue manager in bindings mode. The two local queue managers are connected via Sender/Receiver channel pairs. A third queue manager is located on another machine, and is used as the Coordination Queue Manager. The following diagram details the exact scenario:



Agents Connecting in Client Mode

In this scenario each agent is connected to the same single remote queue manager in client mode. A second queue manager is placed on the Sender machine to act as the coordination queue manager. This coordination queue manager is not highly utilised as it is not directly involved in the transfers and so will have little or no effect on the Sender CPU values that are collected. The following diagram details the exact scenario:



2.1 - 1MB File Size

Table 1 shows the full list of results for 1MB file size. Charts showing the relevant times and CPU utilisation can be seen in Figures 1, 2, 3, 4.

Bindings/Client	Single/Multiple	ChunkSize	Transfer Time	Sender CPU	Receiver CPU	Server CPU
Bindings	Single	65536	108.94	11.15	16.64	N/A
Bindings	Single	131072	90.07	8.69	17.49	N/A
Bindings	Single	262144	82.5	7.11	17.13	N/A
Bindings	Single	524288	79.44	6.66	16.64	N/A
Bindings/Client	Single/Multiple	ChunkSize	Transfer Time	Sender CPU	Receiver CPU	Server CPU
Bindings	Multiple	65536	74.35	8.64	41.92	N/A
Bindings	Multiple	131072	69.6	9.48	45.26	N/A
Bindings	Multiple	262144	59.35	13.94	46.15	N/A
Bindings	Multiple	524288	54.25	13.77	44.93	N/A
Bindings/Client	Single/Multiple	ChunkSize	Transfer Time	Sender CPU	Receiver CPU	Server CPU
Client	Single	65536	127.28	5.47	13.37	13.22
Client	Single	131072	92.13	4.93	14.78	10.83
Client	Single	262144	86.87	4.1	14.97	10.83
Client	Single	524288	83.51	3.72	14.97	8.01
Bindings/Client	Single/Multiple	ChunkSize	Transfer Time	Sender CPU	Receiver CPU	Server CPU
Client	Multiple	65536	60.83	8.21	45.3	15.25
Client	Multiple	131072	52.19	8.87	43.52	16.22
Client	Multiple	262144	50.7	8.14	45.48	15.99
Client	Multiple	524288	60.75	14.41	44.03	20.92

Table 1

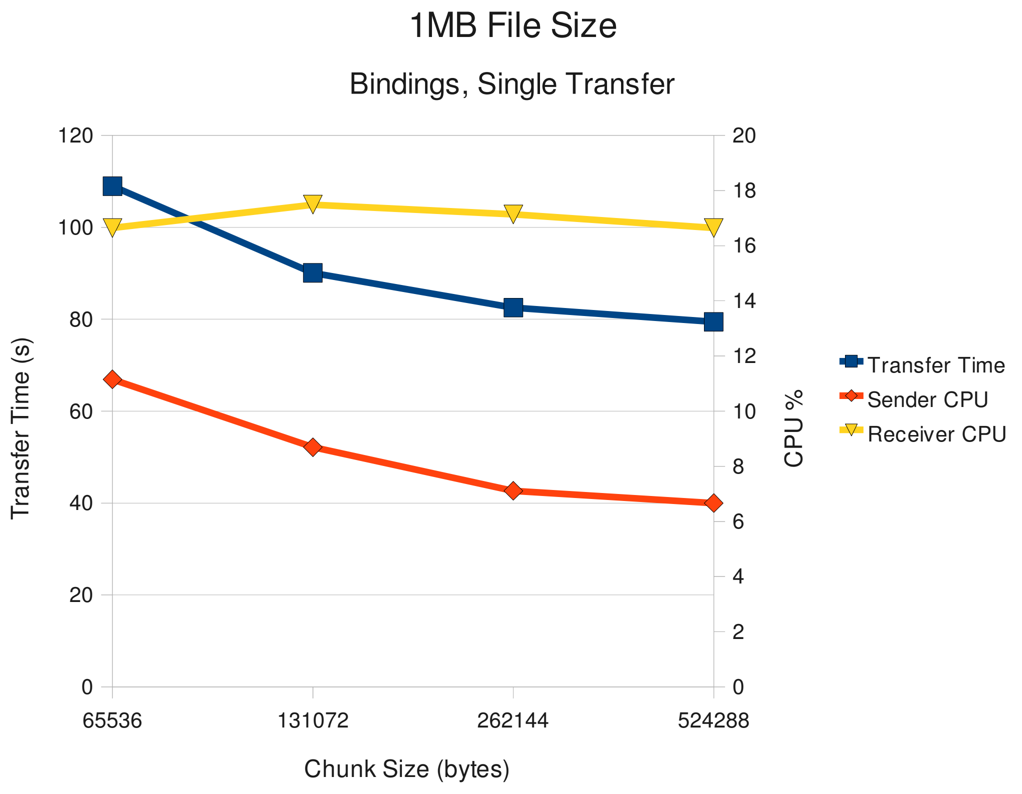


Figure 1

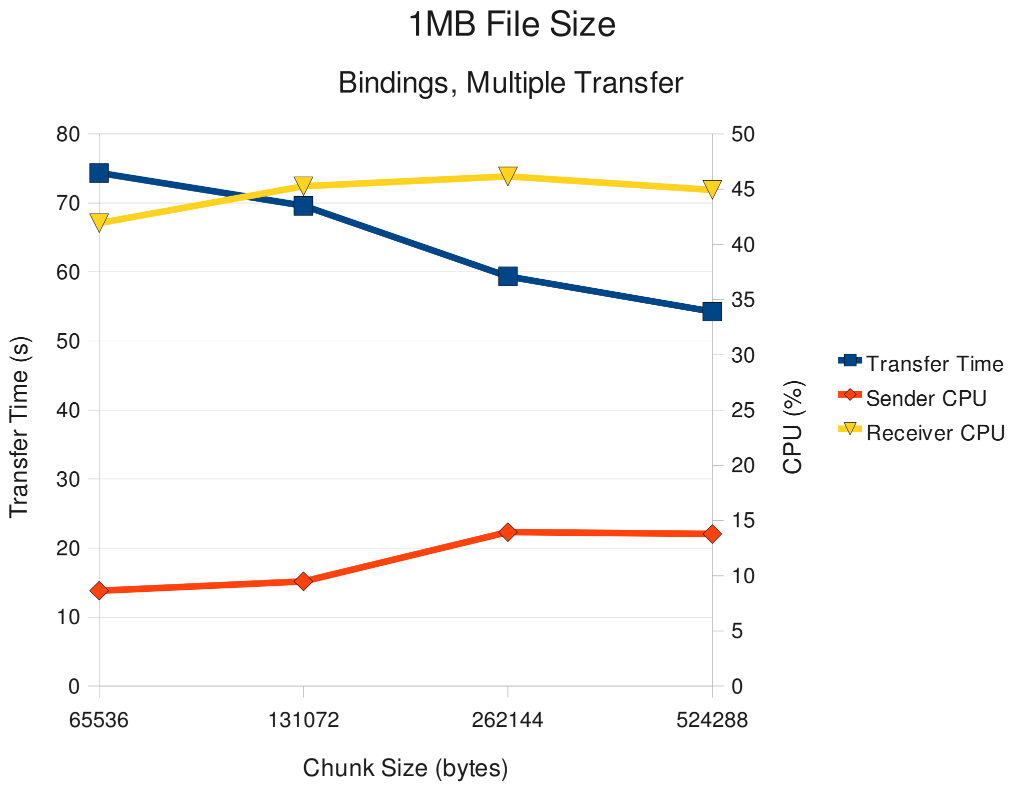


Figure 2

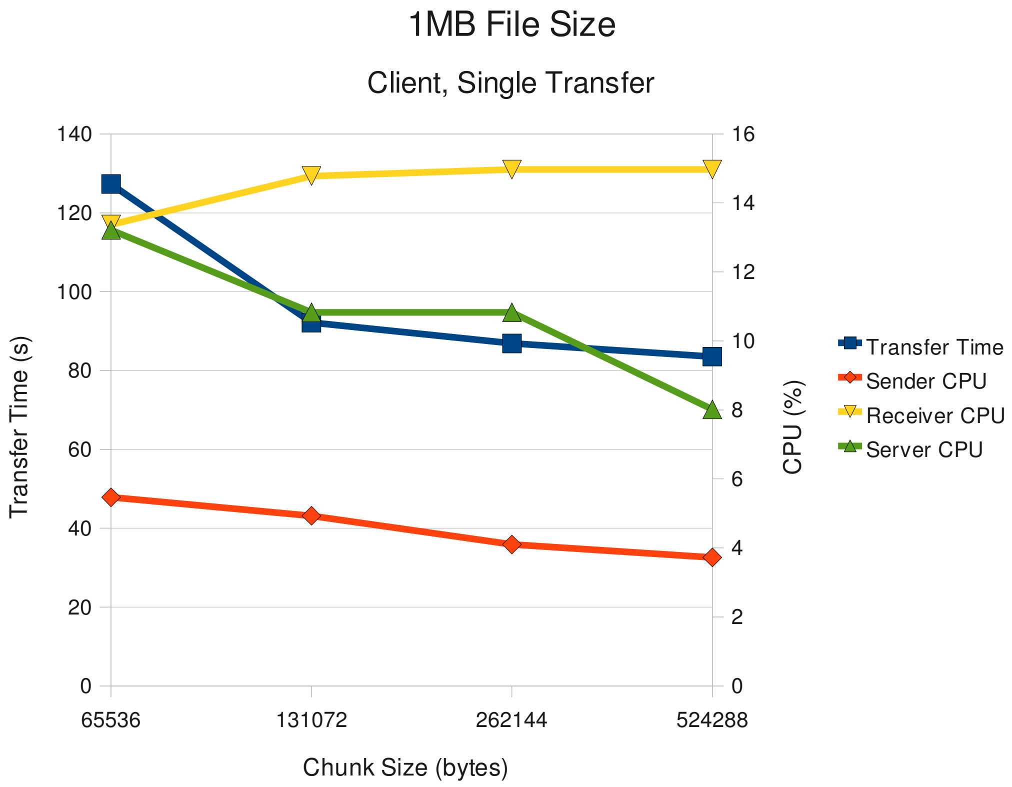


Figure 3

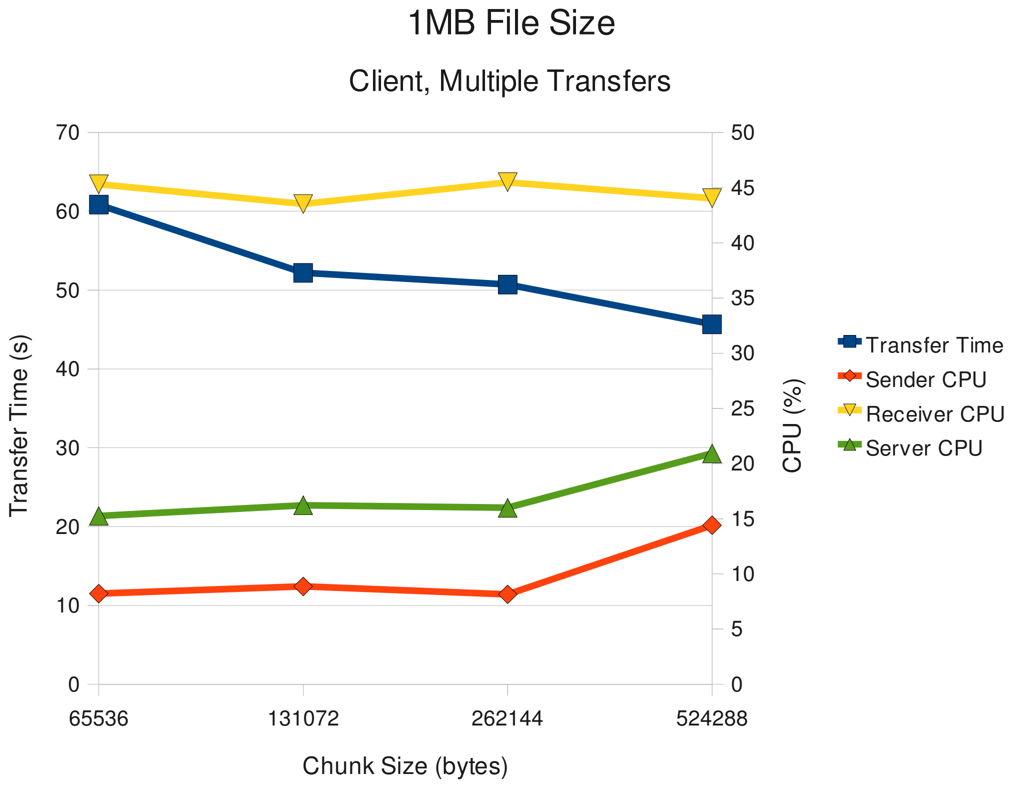
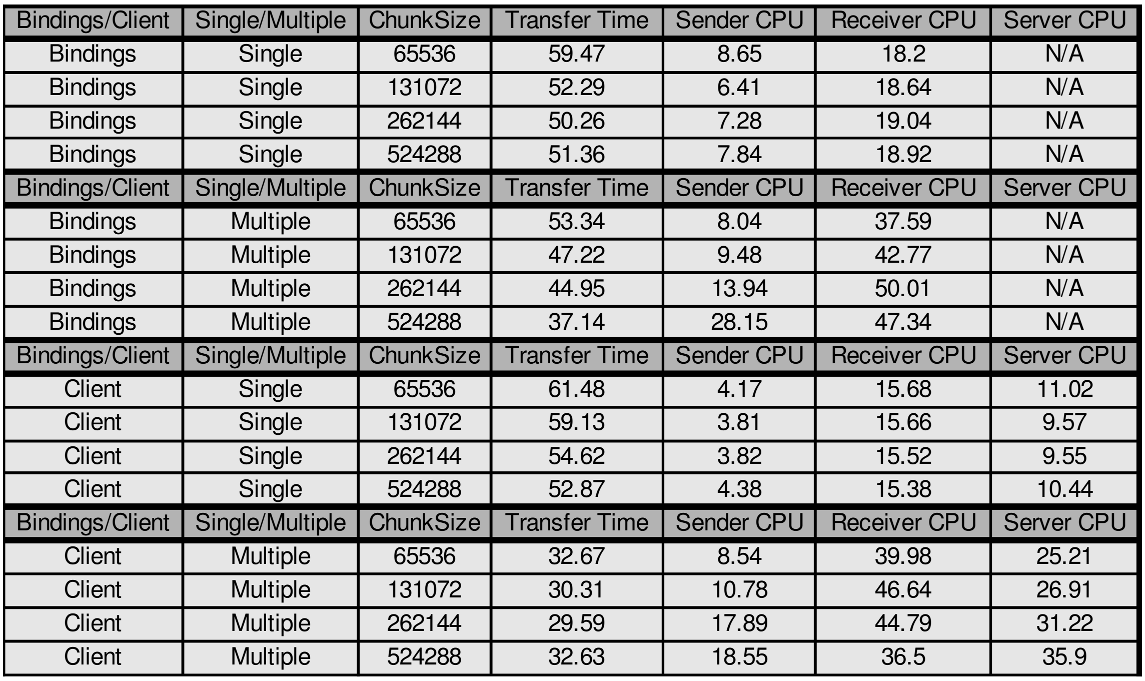


Figure 4

The results above show that for single transfers, the larger the chunk size, the quicker the test completed. This would not be expected to continue much beyond 524288. For multiple client transfers there is clearly an optimum chunk size in the range of 262144. Note the relatively high Receiver CPU compared to the Sender CPU. This is because MQGET calls are more CPU intensive then MQPUT calls. See the relevant Performance SupportPac for Linux for more information.

2.2 - 10MB File Size

Table 2 shows the full list of results for 10MB file size. Charts showing the relevant times and CPU utilisation can be seen in Figures 5, 6, 7 and 8.



Bindings/Client	Single/Multiple	ChunkSize	Transfer Time	Sender CPU	Receiver CPU	Server CPU
Bindings	Single	65536	59.47	8.65	18.2	N/A
Bindings	Single	131072	52.29	6.41	18.64	N/A
Bindings	Single	262144	50.26	7.28	19.04	N/A
Bindings	Single	524288	51.36	7.84	18.92	N/A
Bindings/Client	Single/Multiple	ChunkSize	Transfer Time	Sender CPU	Receiver CPU	Server CPU
Bindings	Multiple	65536	53.34	8.04	37.59	N/A
Bindings	Multiple	131072	47.22	9.48	42.77	N/A
Bindings	Multiple	262144	44.95	13.94	50.01	N/A
Bindings	Multiple	524288	37.14	28.15	47.34	N/A
Bindings/Client	Single/Multiple	ChunkSize	Transfer Time	Sender CPU	Receiver CPU	Server CPU
Client	Single	65536	61.48	4.17	15.68	11.02
Client	Single	131072	59.13	3.81	15.66	9.57
Client	Single	262144	54.62	3.82	15.52	9.55
Client	Single	524288	52.87	4.38	15.38	10.44
Bindings/Client	Single/Multiple	ChunkSize	Transfer Time	Sender CPU	Receiver CPU	Server CPU
Client	Multiple	65536	32.67	8.54	39.98	25.21
Client	Multiple	131072	30.31	10.78	46.64	26.91
Client	Multiple	262144	29.59	17.89	44.79	31.22
Client	Multiple	524288	32.63	18.55	36.5	35.9

Table 2

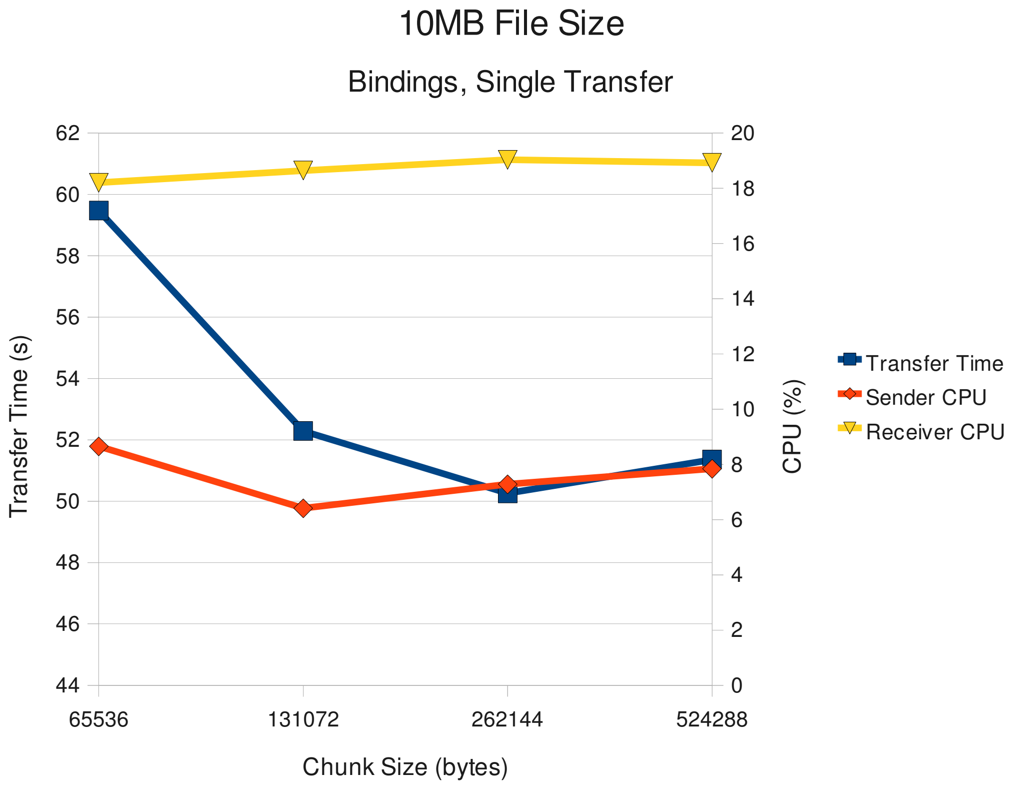

10MB File Size
Bindings, Single Transfer
Transfer Time (s)
62
60
58
56
54
52
50
48
46
44
CPU (%)
20
18
16
14
12
10
8
6
4
2
0
65536
131072
262144
524288
Chunk Size (bytes)
Transfer Time
Sender CPU
Receiver CPU

Figure 5

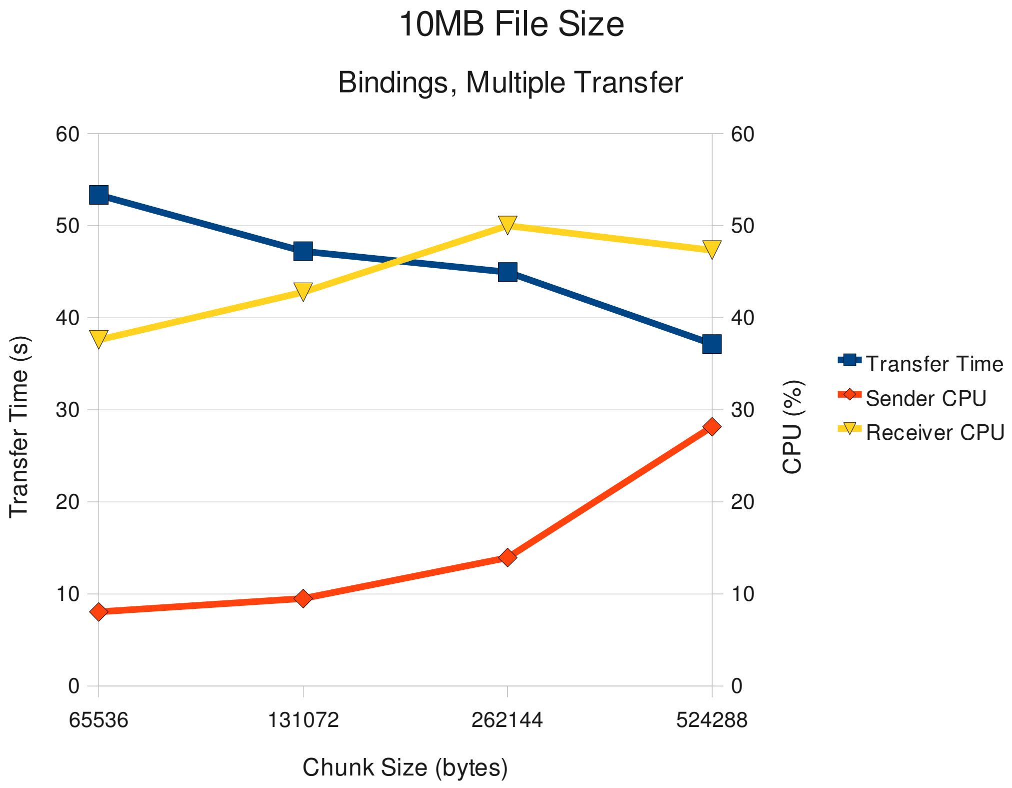


Figure 6

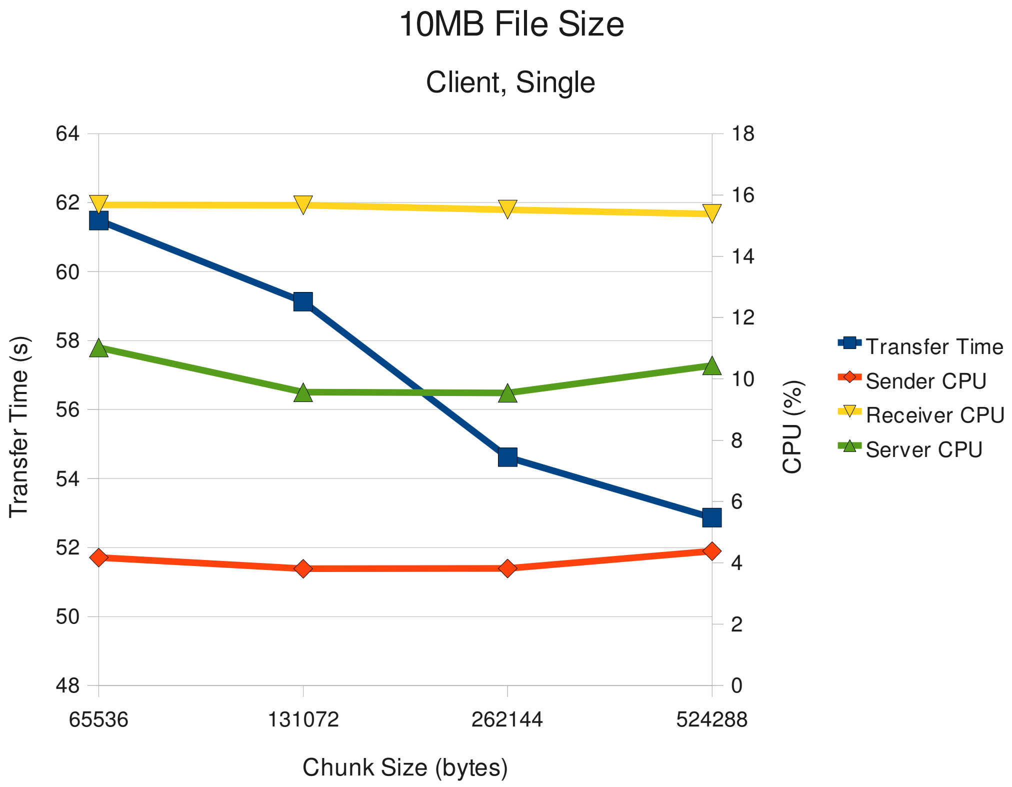


Figure 7

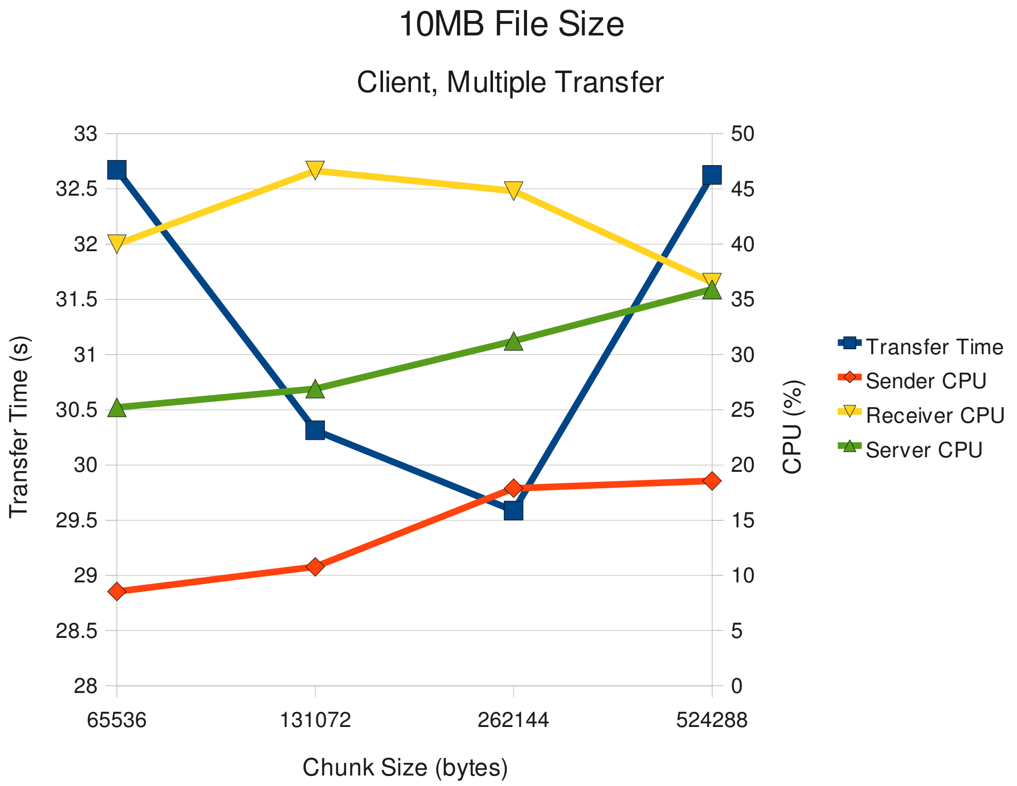


Figure 8

As with 1MB, there is a direct correlation between a larger chunk size and a lower transfer time. Both with the single bindings transfer, and the multiple client transfer you can see that peak performance is obtained when using 262144 as a chunk size value.

2.3 - 100MB File Size

Table 3 shows the full list of results for 100MB file size. Charts showing the relevant times and cpu utilisation can be seen in Figures 9, 10, 11 and 12.

Bindings/Client	Single/Multiple	ChunkSize	Transfer Time	Sender CPU	Receiver CPU	Server CPU
Bindings	Single	65536	55.99	6.53	18.42	N/A
Bindings	Single	131072	49.73	5.9	18.68	N/A
Bindings	Single	262144	46.69	7.67	18.6	N/A
Bindings	Single	524288	47.44	7.68	19.02	N/A
Bindings/Client	Single/Multiple	ChunkSize	Transfer Time	Sender CPU	Receiver CPU	Server CPU
Bindings	Multiple	65536	45.07	9.28	38.97	N/A
Bindings	Multiple	131072	42.24	10.88	33.62	N/A
Bindings	Multiple	262144	39.43	12.62	51.76	N/A
Bindings	Multiple	524288	31.11	28.07	51.38	N/A
Bindings/Client	Single/Multiple	ChunkSize	Transfer Time	Sender CPU	Receiver CPU	Server CPU
Client	Single	65536	56.86	3.93	15.76	10.89
Client	Single	131072	50.47	4.09	15.86	10.39
Client	Single	262144	48.85	4.18	15.84	10.31
Client	Single	524288	48.12	5.12	15.49	12.78
Bindings/Client	Single/Multiple	ChunkSize	Transfer Time	Sender CPU	Receiver CPU	Server CPU
Client	Multiple	65536	30.11	9.64	43.2	27.51
Client	Multiple	131072	29.09	12.92	47.16	28.17
Client	Multiple	262144	29.37	16.2	38.04	32
Client	Multiple	524288	31.49	25.73	35.17	36.39

Table 3

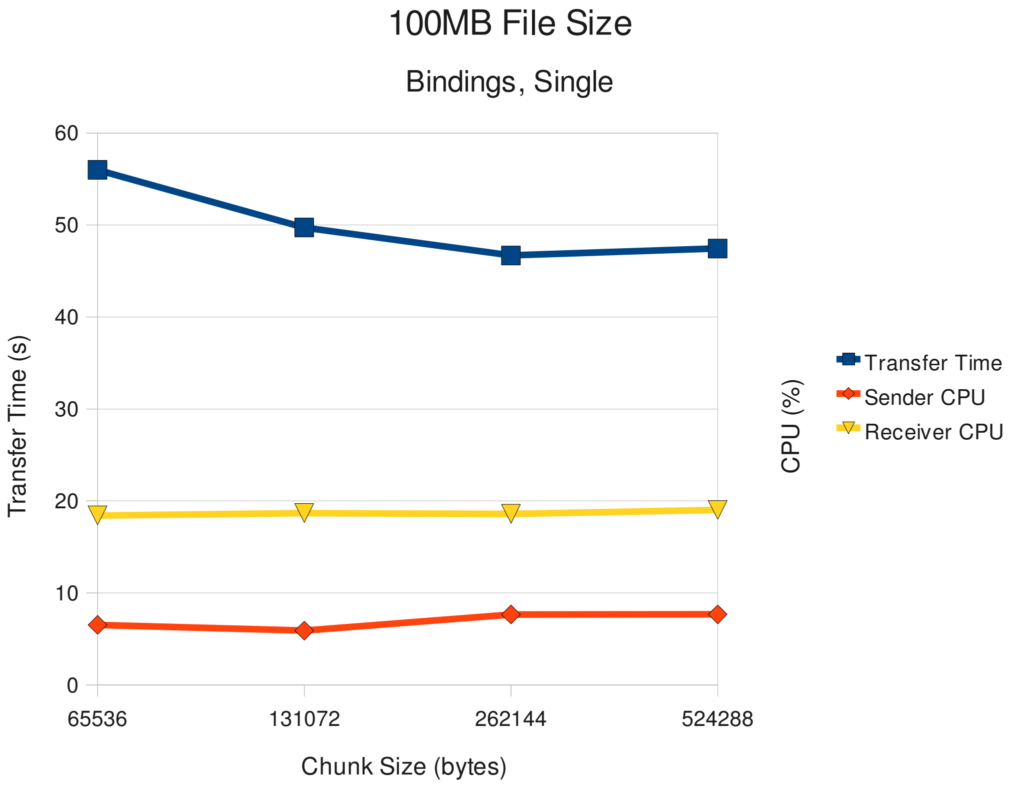


Figure 9

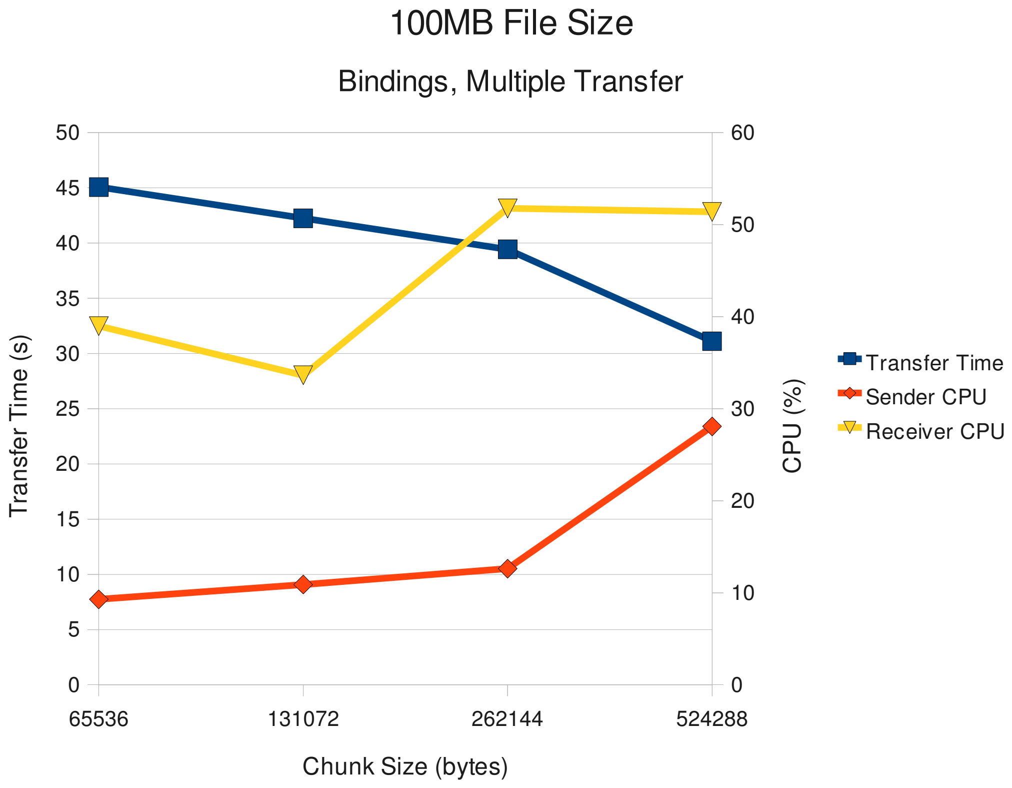


Figure 10

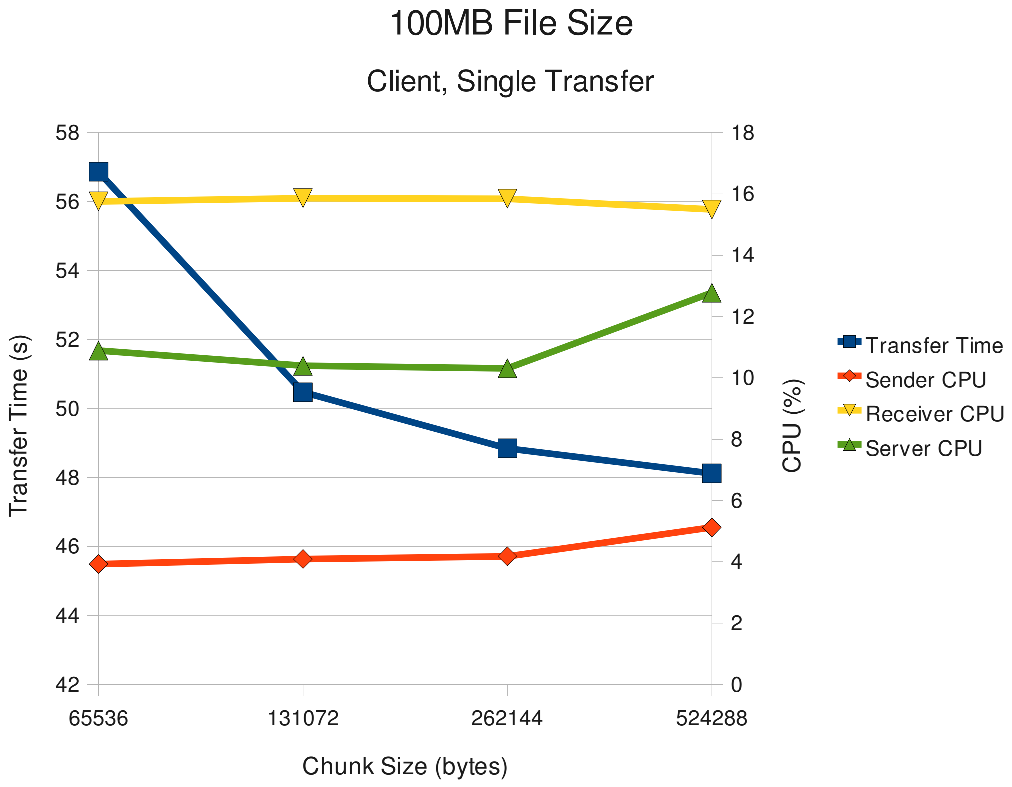


Figure 11

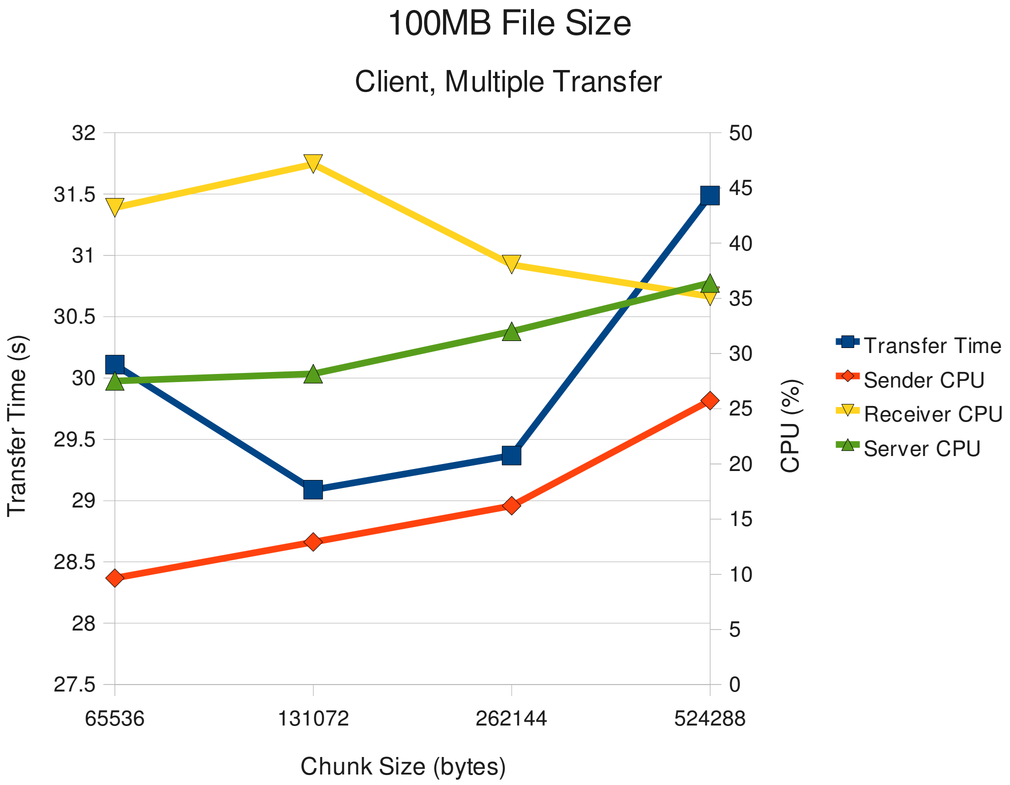


Illustration 1: Figure 12

Using 100MB files continues the trend of the smaller files sizes. A single bindings transfer, and multiple client transfers perform at their best when using 262144, or less as the chunk size value. It is interesting to note the high Server CPU value, does not correlate to a quicker transfer in the multiple client transfer scenario.

2.4 - 1MB Scenario Comparison

The following tables and figures show the difference in transfer time and CPU utilisation by scenario when using the 1MB file size.

Bindings/Client	Single/Multiple	ChunkSize	Scenario	Transfer Time	Sender CPU	Receiver CPU	Server CPU
Bindings	Single	65536	Bindings / Single Transfer	108.94	11.15	16.64	N/A
Bindings	Multiple	65536	Bindings / Multiple Transfer	74.35	8.64	41.92	N/A
Client	Single	65536	Client / Single Transfer	127.28	5.47	13.37	13.22
Client	Multiple	65536	Client / Multiple Transfer	60.83	8.21	45.3	15.25
Bindings/Client	Single/Multiple	ChunkSize	Scenario	Transfer Time	Sender CPU	Receiver CPU	Server CPU
Bindings	Single	131072	Bindings / Single Transfer	90.07	8.69	17.49	N/A
Bindings	Multiple	131072	Bindings / Multiple Transfer	69.6	9.48	45.26	N/A
Client	Single	131072	Client / Single Transfer	92.13	4.93	14.78	10.83
Client	Multiple	131072	Client / Multiple Transfer	52.19	8.87	43.52	16.22
Bindings/Client	Single/Multiple	ChunkSize	Scenario	Transfer Time	Sender CPU	Receiver CPU	Server CPU
Bindings	Single	262144	Bindings / Single Transfer	82.5	7.11	17.13	N/A
Bindings	Multiple	262144	Bindings / Multiple Transfer	59.35	13.94	46.15	N/A
Client	Single	262144	Client / Single Transfer	86.87	4.1	14.97	10.83
Client	Multiple	262144	Client / Multiple Transfer	50.7	8.14	45.48	15.99
Bindings/Client	Single/Multiple	ChunkSize	Scenario	Transfer Time	Sender CPU	Receiver CPU	Server CPU
Bindings	Single	524288	Bindings / Single Transfer	79.44	6.66	16.64	N/A
Bindings	Multiple	524288	Bindings / Multiple Transfer	54.25	13.77	44.93	N/A
Client	Single	524288	Client / Single Transfer	83.51	3.72	14.97	8.01
Client	Multiple	524288	Client / Multiple Transfer	60.75	14.41	44.03	20.92

Table 4

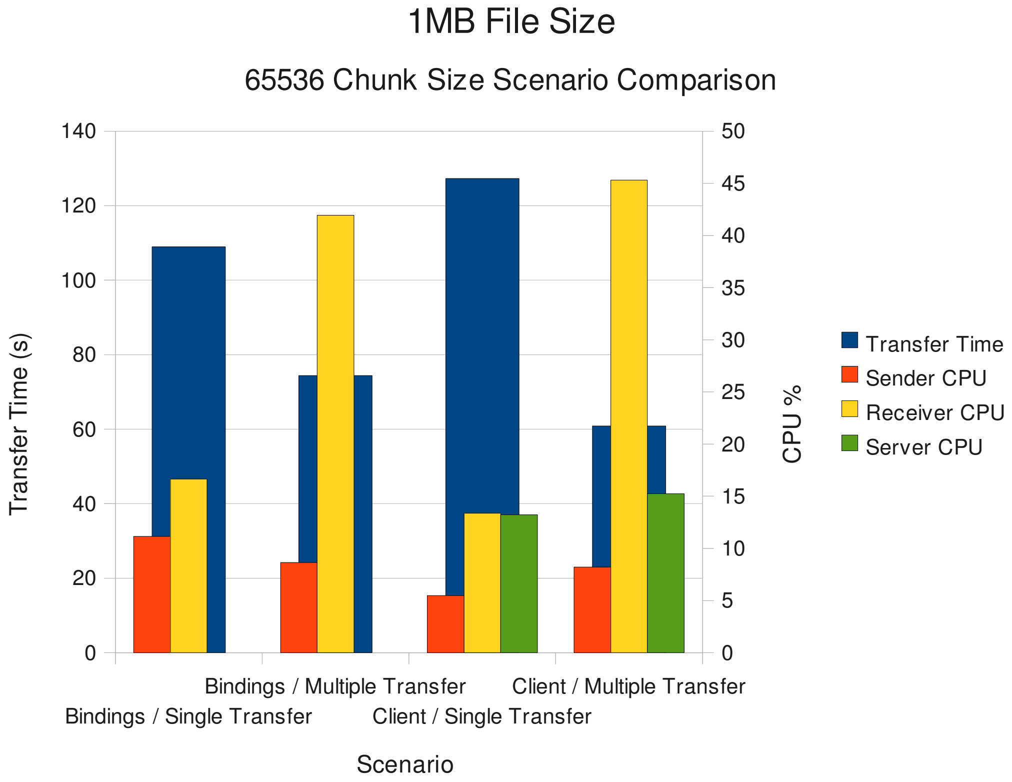


Figure 13

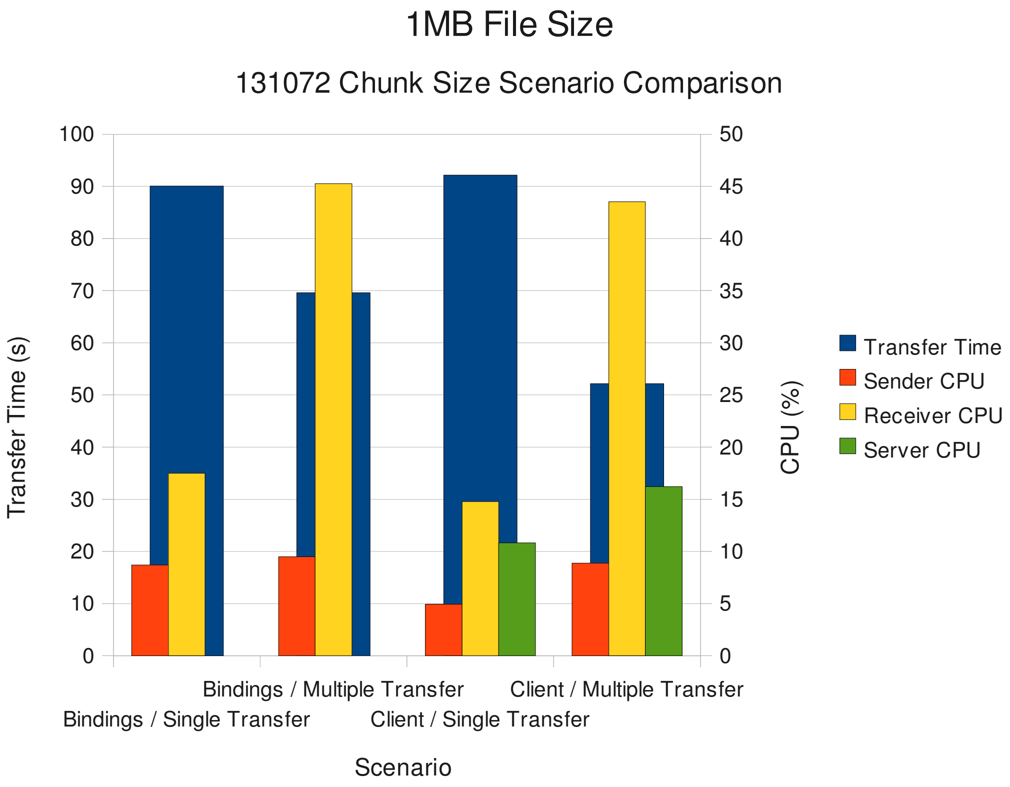


Figure 14

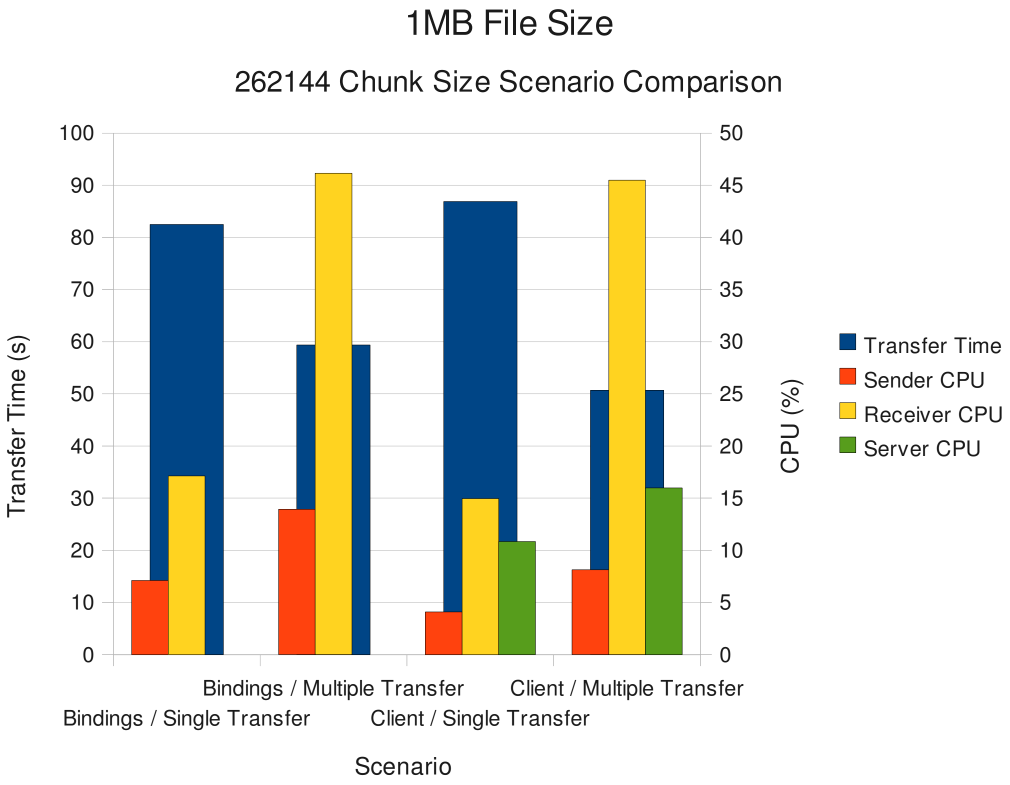


Figure 15

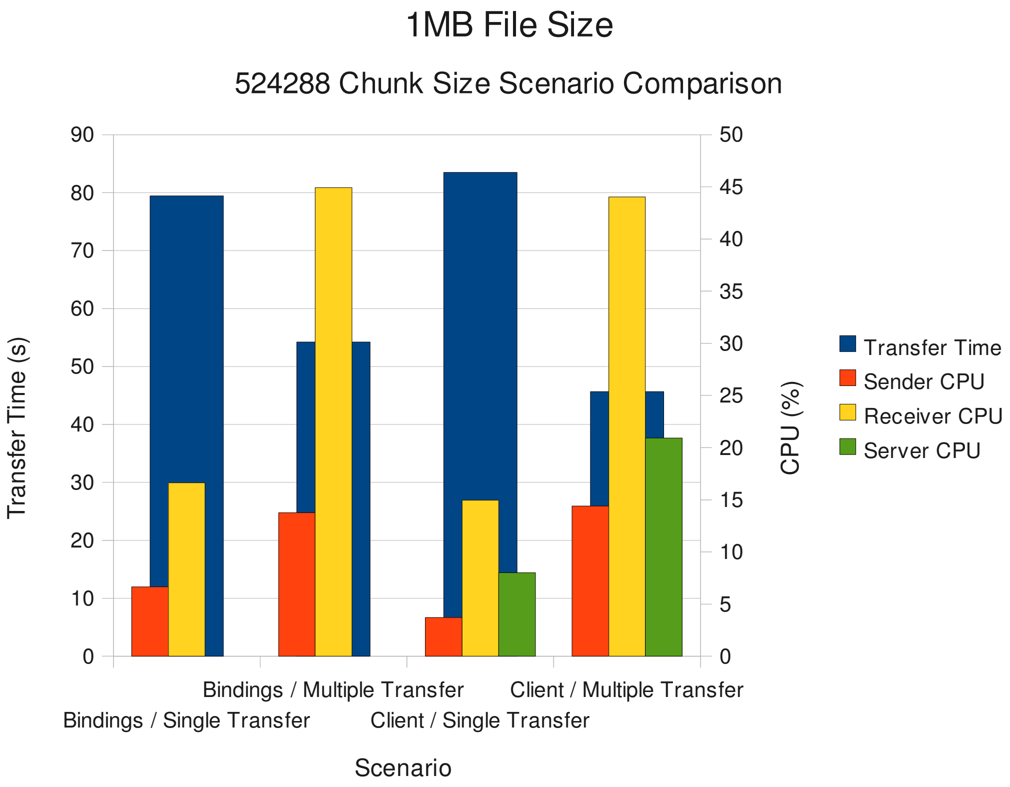


Figure 16

These graphs and results clearly show that due to the lack of high performance disks, client connectivity for agents running multiple transfers performs the best overall. This is due to WebSphere MQ and Managed File Transfer agents both writing to the same disks, causing an I/O bottleneck in the case of the bindings scenario. Note that for multiple transfers, both client and bindings show similarly high levels of Sender and Receiver CPU utilisation, which backs up the I/O bottleneck reasoning.

2.4 - 10MB Scenario Comparison

The following tables and figures show the difference in transfer time and CPU utilisation by scenario when using the 10MB file size.

Bindings/Client	Single/Multiple	ChunkSize	Scenario	Transfer Time	Sender CPU	Receiver CPU	Server CPU
Bindings	Single	65536	Bindings / Single Transfer	59.47	8.65	18.2	N/A
Bindings	Multiple	65536	Bindings / Multiple Transfer	53.34	8.04	37.59	N/A
Client	Single	65536	Client / Single Transfer	61.48	4.17	15.68	11.02
Client	Multiple	65536	Client / Multiple Transfer	32.67	8.54	39.98	25.21
Bindings/Client	Single/Multiple	ChunkSize	Scenario	Transfer Time	Sender CPU	Receiver CPU	Server CPU
Bindings	Single	131072	Bindings / Single Transfer	52.29	6.41	18.64	N/A
Bindings	Multiple	131072	Bindings / Multiple Transfer	47.22	9.48	42.77	N/A
Client	Single	131072	Client / Single Transfer	59.13	3.81	15.66	9.57
Client	Multiple	131072	Client / Multiple Transfer	30.31	10.78	46.64	26.91
Bindings/Client	Single/Multiple	ChunkSize	Scenario	Transfer Time	Sender CPU	Receiver CPU	Server CPU
Bindings	Single	262144	Bindings / Single Transfer	50.26	7.28	19.04	N/A
Bindings	Multiple	262144	Bindings / Multiple Transfer	44.95	13.94	50.01	N/A
Client	Single	262144	Client / Single Transfer	54.62	3.82	15.52	9.55
Client	Multiple	262144	Client / Multiple Transfer	29.59	17.89	44.79	31.22
Bindings/Client	Single/Multiple	ChunkSize	Scenario	Transfer Time	Sender CPU	Receiver CPU	Server CPU
Bindings	Single	524288	Bindings / Single Transfer	51.36	7.84	18.92	N/A
Bindings	Multiple	524288	Bindings / Multiple Transfer	37.14	28.15	47.34	N/A
Client	Single	524288	Client / Single Transfer	52.87	4.38	15.38	10.44
Client	Multiple	524288	Client / Multiple Transfer	32.63	18.55	36.5	35.9

Table 5

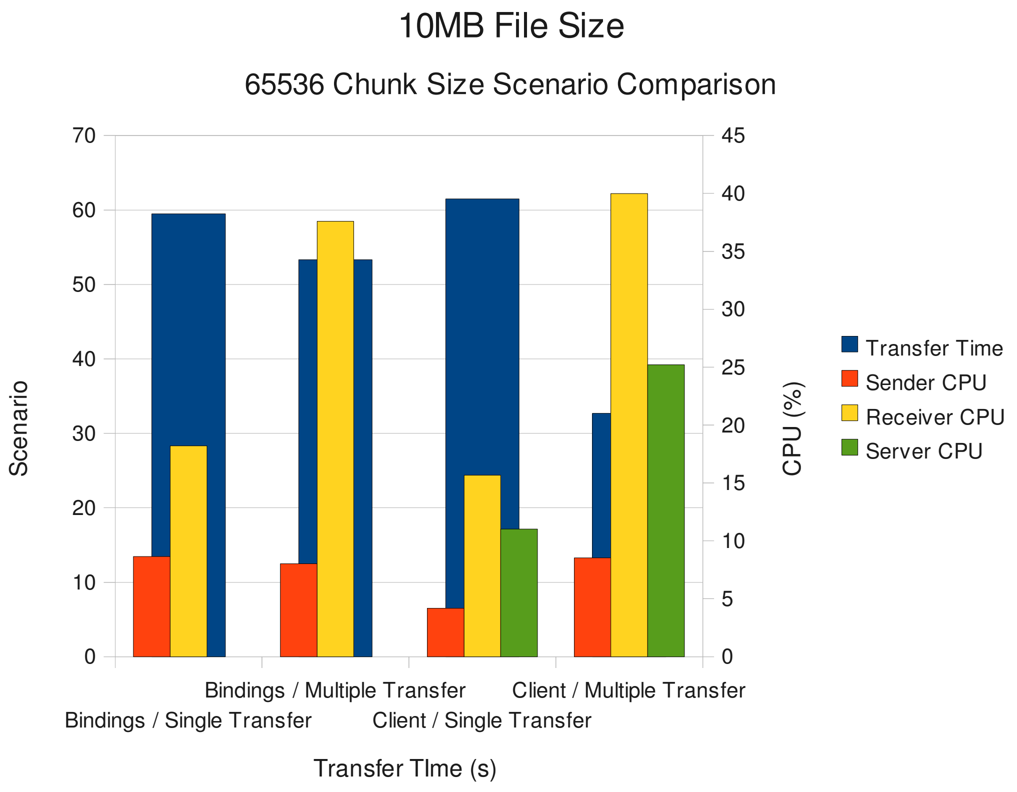


Figure 17

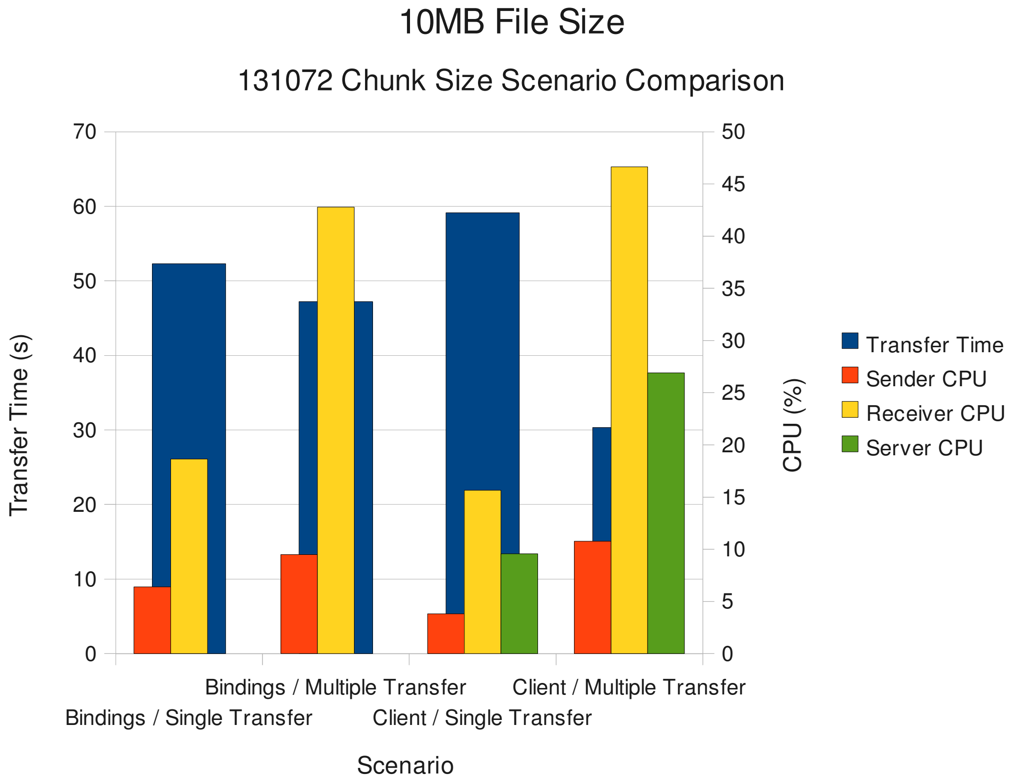


Figure 18

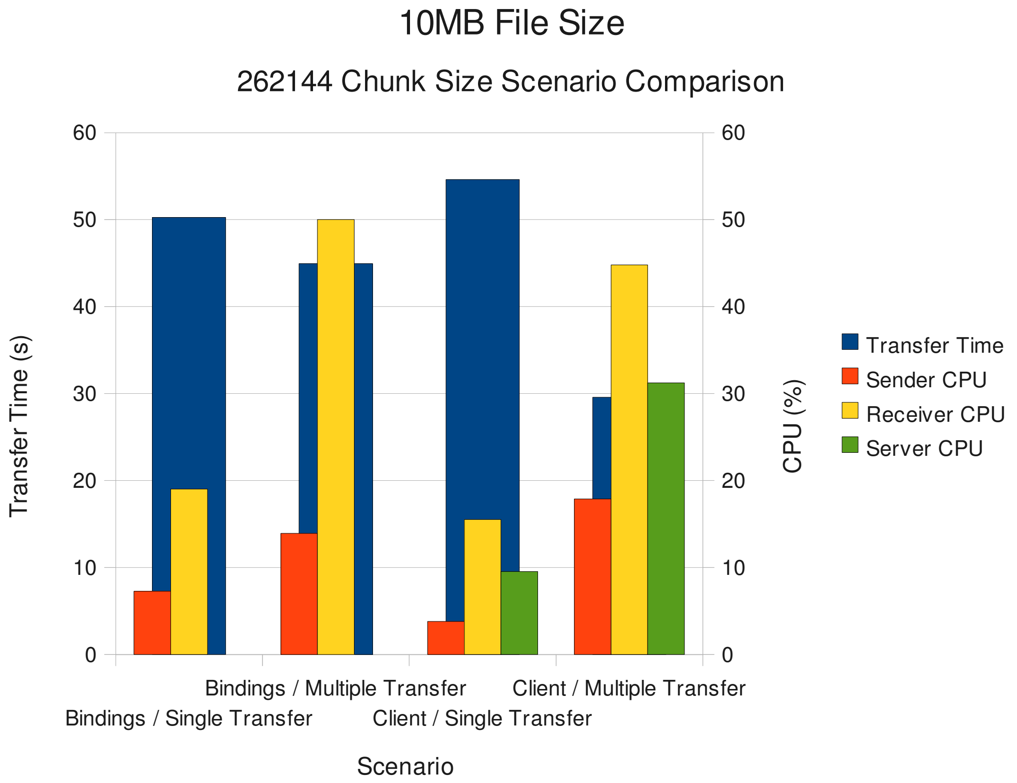


Figure 19

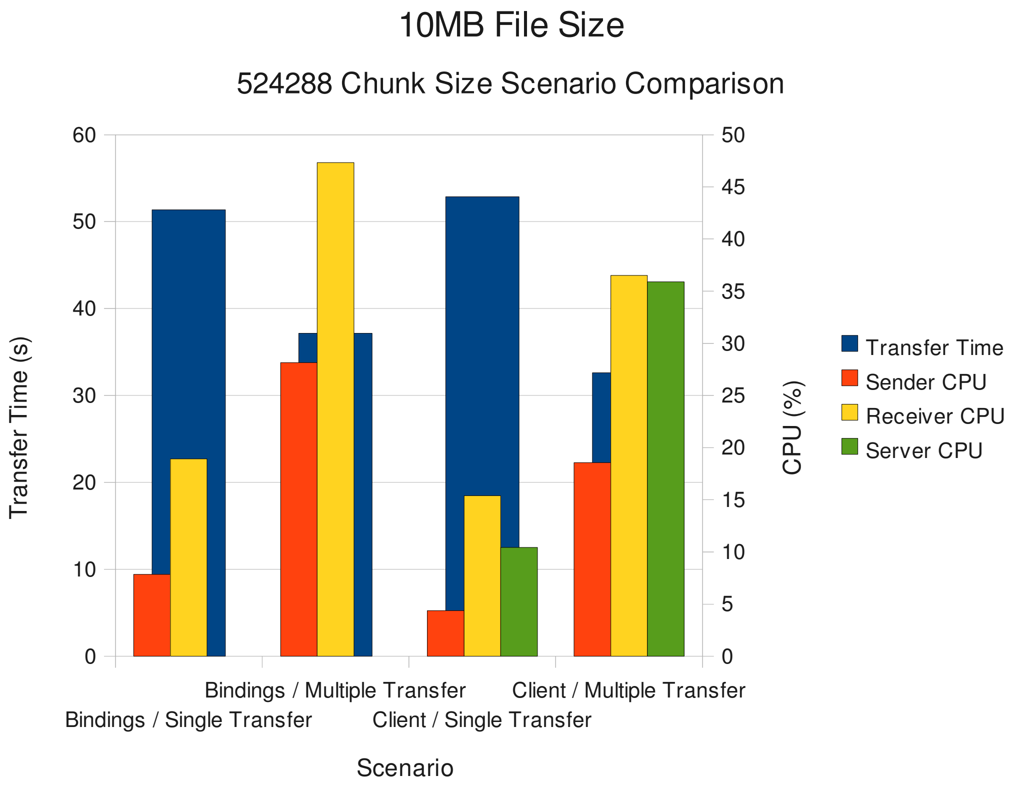


Figure 20

These results continue the trend shown in the 1MB file size scenarios. Multiple client transfers are the quickest to complete. For the Single transfer scenarios there is little difference between the transfer times.

2.4 - 100MB Scenario Comparison

The following tables and figures show the difference in transfer time and CPU utilisation by scenario when using the 100MB file size.

Bindings/Client	Single/Multiple	ChunkSize	Scenario	Transfer Time	Sender CPU	Receiver CPU	Server CPU
Bindings	Single	65536	Bindings / Single Transfer	55.99	6.53	18.42	N/A
Bindings	Multiple	65536	Bindings / Multiple Transfer	45.07	9.28	38.97	N/A
Client	Single	65536	Client / Single Transfer	56.86	3.93	15.76	10.89
Client	Multiple	65536	Client / Multiple Transfer	30.11	9.64	43.2	27.51
Bindings/Client	Single/Multiple	ChunkSize	Scenario	Transfer Time	Sender CPU	Receiver CPU	Server CPU
Bindings	Single	131072	Bindings / Single Transfer	49.73	5.9	18.68	N/A
Bindings	Multiple	131072	Bindings / Multiple Transfer	42.24	10.88	33.62	N/A
Client	Single	131072	Client / Single Transfer	50.47	4.09	15.86	10.39
Client	Multiple	131072	Client / Multiple Transfer	29.09	12.92	47.16	28.17
Bindings/Client	Single/Multiple	ChunkSize	Scenario	Transfer Time	Sender CPU	Receiver CPU	Server CPU
Bindings	Single	262144	Bindings / Single Transfer	46.69	7.67	18.6	N/A
Bindings	Multiple	262144	Bindings / Multiple Transfer	39.43	12.62	51.76	N/A
Client	Single	262144	Client / Single Transfer	48.85	4.18	15.84	10.31
Client	Multiple	262144	Client / Multiple Transfer	29.37	16.2	38.04	32
Bindings/Client	Single/Multiple	ChunkSize	Scenario	Transfer Time	Sender CPU	Receiver CPU	Server CPU
Bindings	Single	524288	Bindings / Single Transfer	47.44	7.68	19.02	N/A
Bindings	Multiple	524288	Bindings / Multiple Transfer	31.11	28.07	51.38	N/A
Client	Single	524288	Client / Single Transfer	48.12	5.12	15.49	12.78
Client	Multiple	524288	Client / Multiple Transfer	31.49	25.73	35.17	36.39

Table 6

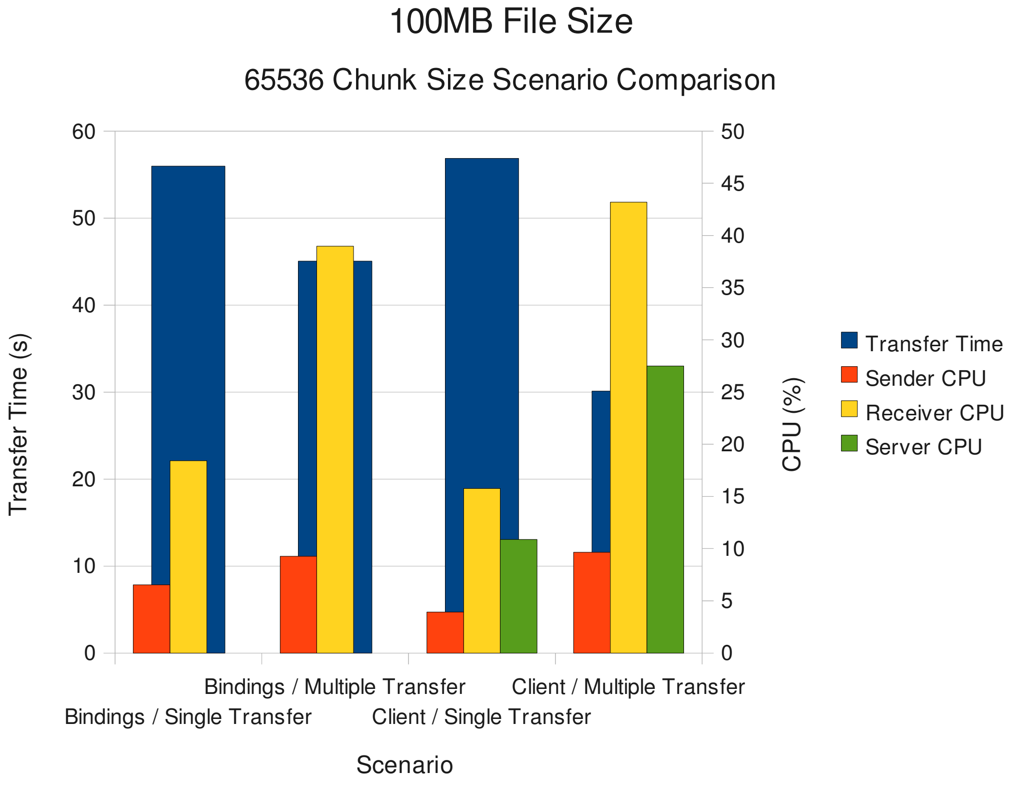


Figure 21

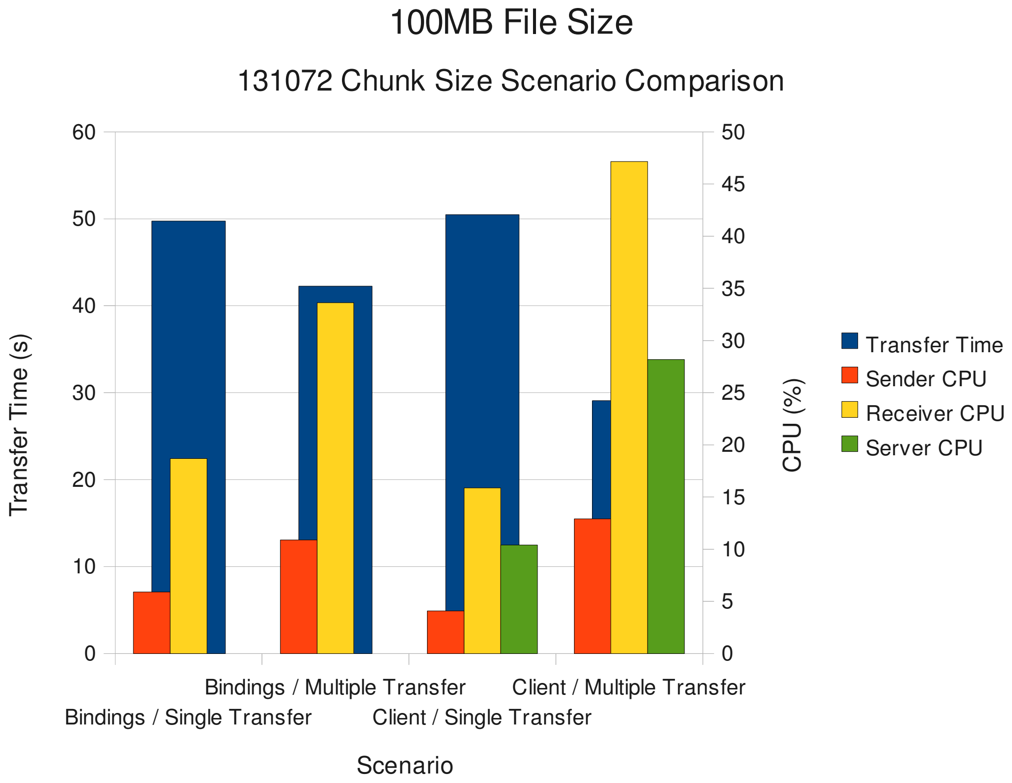


Figure 22

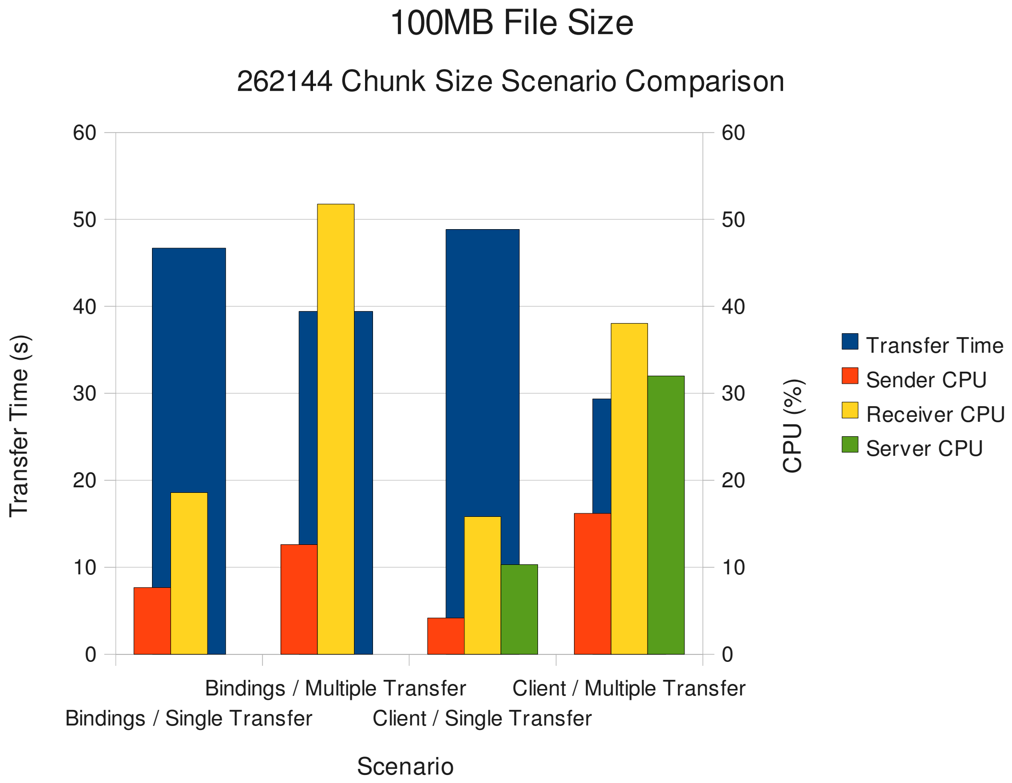


Figure 23

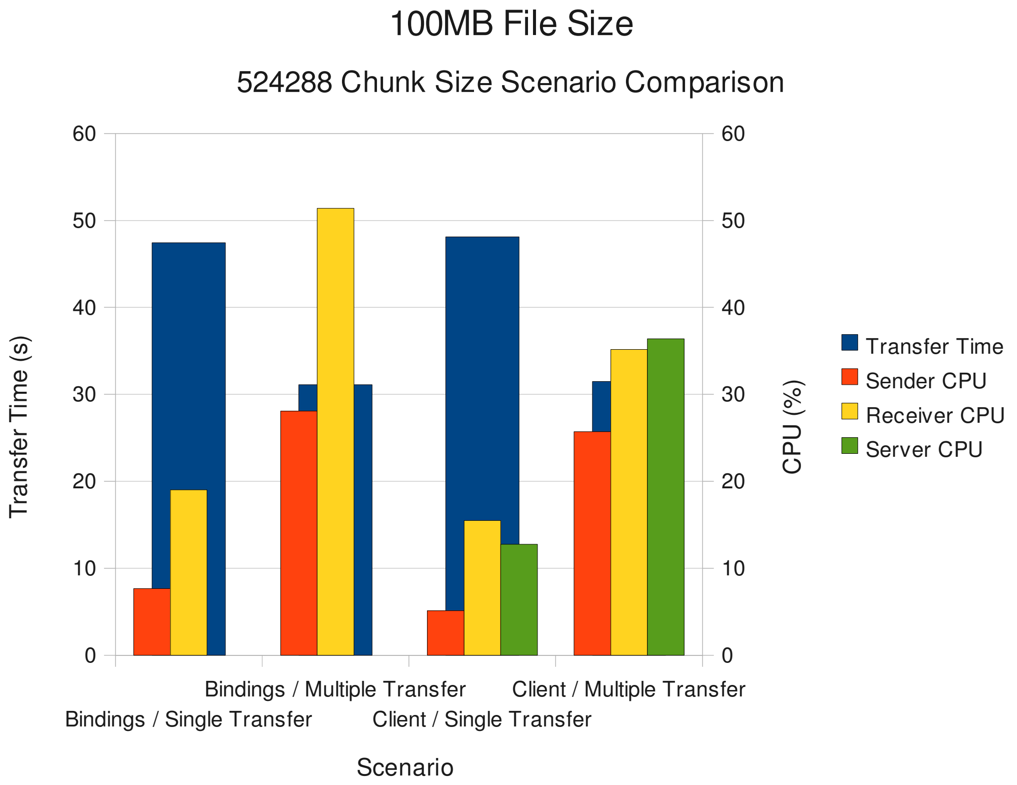


Figure 24

With an agent chunk size value of 524288, we can see that multiple bindings and multiple client scenarios are almost identical in transfer time. The addition of the off board MQ queue manager instance altered the CPU utilisation as expected, but did not give a decrease in transfer time. This differs from the results of the smaller (1MB, 10MB) files. At all other chunk sizes, we see the trend continuing of multiple client transfers being the best performing scenario.

3 Tuning Recommendations

3.1 - WebSphere MQ Setup

Readers of this performance guide should make themselves familiar with the WebSphere MQ Performance Supportpacs that are continually released. They can be found here: http://www-01.ibm.com/support/docview.wss?rs=171&uid=swg27007197#1. Of particular interest for Linux are Supportpacs MPL3 for MQ Version 7 and MPL6 for MQ Version 6.

For this performance report, advice was taken from the aforementioned (MPL3) and applied to the queue managers created accordingly. Queue managers were created using the following *crtmqm* command:

```
crtmqm -q -u SYSTEM.DEAD.LETTER.QUEUE -lp 16 -lf 16384 <QueueManagerName>
```

Once the queue manager was created, tuning parameters were added to the queue managers' qm.ini as follows:

```
Channels:
  MQIBindType=FASTPATH

TuningParameters:
  DefaultPQBufferSize=1045876
  DefaultQBufferSize=1048576
```

Note that the qm.ini was updated before the queue manager was started (and therefore before the WebSphere MQ Managed File Transfer objects were created).

By increasing the amount of memory available to queues for persistent and non-persistent messages, you can help to avoid writing messages out to disk unnecessarily. Turning on FASTPATH for channels removes the channel process, and enables the channel to run within the main queue manager process. Please consult your documentation to understand what this means for your WebSphere MQ installation.

For more information on tuning a WebSphere MQ queue manager, please refer to the Supportpacs mentioned above.

The use of high performance disks (SAN for example) is recommended for a WebSphere MQ installation. Separating out your /var/mqm/log directory structure from the regular file system is a well documented best practice that helps to create a queue manager that responds well to high throughput scenarios.

3.2 - WebSphere MQ File Transfer Edition Setup

When running agents for this performance report, the following environment property was used:

> *export FTE_JVM_PROPERTIES=”-Xmx2048M -Xms2048M”*

This property was set before starting an agent and sets the starting and maximum JVM heapsize to be 2GB. These values were used to ensure that the agent had sufficient memory to allocate when running the multiple transfer scenarios.

As demonstrated in the results, altering the *agentChunkSize* can have a significant impact on both CPU utilisation and transfer time. There is another property *agentWindowSize* that can be used to control the amount of syncpoints commited, and the number of acknowledgements sent between two agents when transferring files. This property has a default value of 10. This means that for every 10 chunks of data sent over WebSphere MQ, the sending agent will take an internal checkpoint, and wait to receive an acknowledgement from the receiving agent before sending more data. The property's default value was determined after extensive performance work during the development of version 7.0.1. Increasing this property increases the amount of data that could potentially need to be re-transmitted if a recovery is required, and is not recommended for unreliable networks.

3.3 – WebSphere MQ File Transfer Edition Transfer recommendations

The following are a list of bullet pointed recommendations when planning your WebSphere MQ File Transfer Edition network.

- Send large numbers of files over multiple transfers, rather then a single large transfer. This will increase the efficiency of the I/O involved in transferring the files, which will ultimately decrease the transfer time.

- Test your typical transfers using a range of agentChunkSize parameters. Depending on the underlying hardware, you may find an optimum value for your setup.

- Multiple smaller files place the agent under strain due to the operating system open/close costs associated with more files. Where possible configure your file creation processes to generate archives of smaller files, enabling FTE to use less open/close calls.

- Reading and writing to physical disk is often going to be the performance bottleneck. For agents that will see a large number of incoming, and outgoing transfers it would be best if high performance disks were used to read data from and write data to. This is demonstrated by the multiple/client and multiple/bindings scenarios. Due to the use of internal disks for WebSphere MQ, Client connectivity actually outperformed Bindings. This behaviour can be explained because the Agents reading/writing to the physical disks at the same time as a local MQ instance, causing an I/O bottleneck.

- When configuring your MQ network, use the appropriate WebSphere MQ Performance Report to apply optimal settings for your platform.

- Ensure that you have sufficient RAM for your Agent. The performance tests used 2GB of RAM, it is recommended that you read your Operating System guide on memory usage and plan accordingly.

4 - Measurement Environment

4.1 - Agents

- WebSphere MQ File Transfer Edition Version 7.0.1 was used for this report.
- Agents connected to WebSphere MQ Version 6.0.2.7 queue managers.
- Default properties were used for agents, except for *agentChunkSize*
- Agents were reading/writing files to the local file system, not the SAN.

4.2 - WebSphere MQ

- WebSphere MQ Version 7.0.0.1 was used for the coordination queue manager
- WebSphere MQ Version 6.0.2.7 was used for agent queue managers
- Queue managers created in accordance with Performance report
- /var/mqm and /var/mqm/log were mounted on SAN disks

4.3 - Operating System

- Red Hat Enterprise Linux 5.3 (Tikanga) 32bit

4.4 - Hardware

eServer x366:	mqperfx3
Processor:	Intel(R) XEON(TM) MP CPU 2.00GHz
Architecture:	4 CPU
Memory (RAM):	8Gb
Disk:	Internal disks for measurements
Network:	1Gbit Ethernet Adapter (onboard)

eServer x366:	mqperfxa, mqperfxb
Processor:	Intel(R) Xeon(TM) MP CPU 3.66GHz
Architecture	4 CPU
Memory (RAM):	4Gb
Disk:	Internal disks for measurements
Network:	1Gbit Ethernet Adapter (onboard)