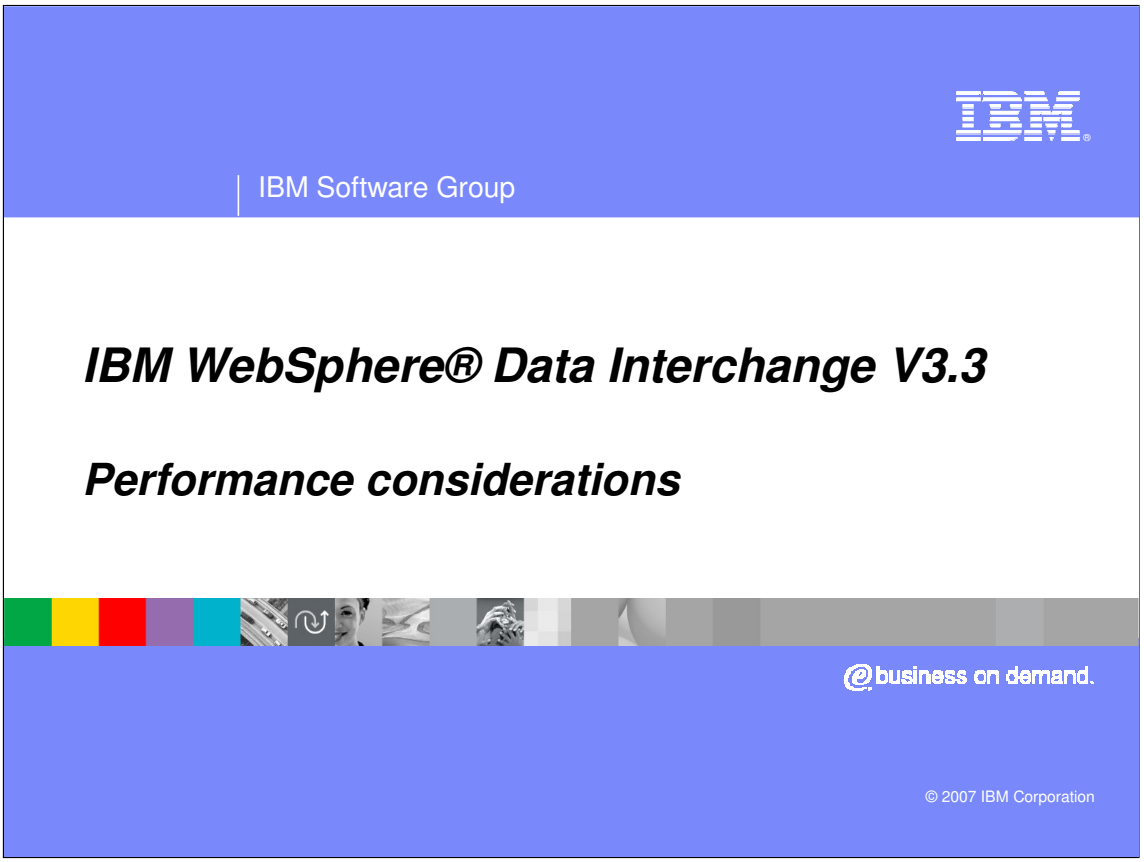


This presentation will review performance considerations with the WebSphere Data Interchange product.

Discussion Objectives

- Variables affecting performance
- Measures of performance
- Techniques for each measure

Some of the topics that will be discussed are:

1) Variables affecting performance – that is, what kind of things can make the translation time vary,

2) Measures of performance – again, how is performance measured and what are the “units” of measurement; elapsed time is an example of one measure, and ,

3) What kind of techniques have been implemented in WDI to reduce the use of resources of measurement.

Variables affecting Performance

- Server Platform
- Number of concurrent executions
- Number of independent MQ messages
- Number of documents in an Interchange
- Number of Interchanges in a MQ Message
- Number of Documents in a file
- Type of map used - Send/Receive vs Data Transform
- Map complexity - Source / Target, number of mappings
- Data size
- Use of Auditing features - MR, OPTRECs, TS
- Persistent MQ messaging
- DB2 binding - Syncpoint TWOPHASE
- Functional Acknowledgments generated

One of the major reasons why it is so difficult to talk about translation performance is that there are a large number of variables affecting the outcome. We'll identify a number of these and offer some discussion:

1) The Server Platform – is this a mainframe or another platform? How fast is the server and how much memory or main storage does it have available.
2) The number of concurrent executions that WDI is handling at once – is it a single thread or many threads that are being handled? Is WDI contending with itself?
3) The number of independent WMQ messages – how many times did WDI have to go to WMQ to get a message to process? As the number of messages increases the length of time and resources used would be expected to increase
4) The number of documents in an Interchange – with EDI, a single interchange can contain a number of transactions; since WDI processes a transaction at a time, an interchange with multiple transactions would be expected to take longer to translate than an interchange with one transaction – if all transactions were the same kind
5) The number of Interchanges in a WMQ message – WDI processes all interchanges in a WMQ message before continuing; if WMQ message time is of concern, the messages with multiple EDI interchanges would have a different effect that WMQ messages with just one Interchange.
6) The number of documents in a file – when file times are compared, the type of document and number of documents in the file are a factor in the relative times.
7) The Type of map used – WDI uses Send / Receive maps and Data Transform maps; the underlying translation algorithms are different and hence the instruction path to process a translation differs
8) The complexity of the maps – Is this a DT Source map or DT Target based map? Are there many mappings or just a few? Both variables affect performance.
9) The size of the data – large transactions take longer than small transactions, but it may not be proportional. The meaning of "Large" may also be different depending upon the type of data – EDI, XML, flat file.
10) The use of auditing features – WDI translates but also allows for reporting and replay depending on installation needs; the use of these features has an effect on translation, but is that effect greater than the need for the function or for doing the function outside of WDI?
11) Persistent MQ messaging – features on related products, like WMQ, can also cause different effects on WDI translation comparisons. When comparing two identical messages, it was determined that defining the WMQ Queue as persistent had a dramatic effect of the processing time. The function is highly desirable in WMQ, but does have an effect.
12) DB2 binding – TWOPHASE commits – like WMQ, DB2 also can have a role with throughput times. The two phase commit option is required if WDI user exits are used and user data tables are updated in that processing.
13) The generation of Functional Acknowledgments – if an EDI FA is generated – an option at the user's discretion, then it takes longer for a transaction to finish processing. The FA requires a translation of its own and the increase should be expected.

There are a number of major considerations when trying to evaluate translation performance. In valid experiments, the experimenter would attempt to hold as many variables constant as possible so that the effect of the remaining variables can be determined. For example, if one were trying to determine how many concurrent threads should be run on a given machine, then a test might be devised with a number of sets of data of the same size, but with different Trading Partner identifiers for each thread. When the machine is exhausted of cycles, the thread data could not be processed.

Measures of Performance

- CPU Time
- Elapsed time
- Thread locking - stretch time
- File Activity
- Database activity - updates, reads, locks
- Number of transactions processed - Throughput

Performance is measured in may ways. Some ways are obvious and address physical resources used. Examples of these are CPU time, or Input / Output requests (I/Os) to files. Others are measures of business requirements, such as throughput – e.g. How many can I get done in 5 minutes? Measures are usually selected based on the measurer or the need.

CPU time is the raw processor time used to translate or complete the translation task. If a server is loaded then CPU time is critical to doing more. Storage and CPU are usually directly related – the more I have in main storage then the more CPU time needed to process the data.

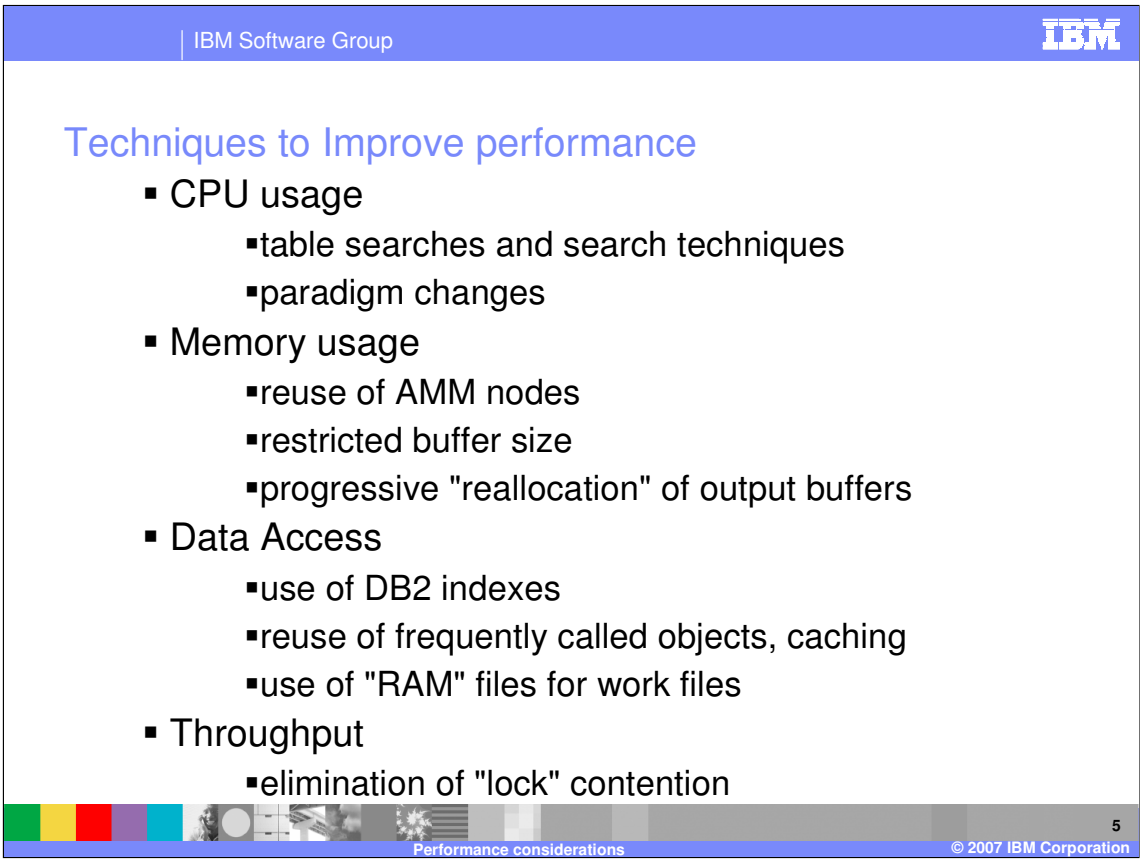
Elapsed time is clock time – e.g. When did the task start and when did it end? That's the time people or dependent tasks have to wait.

Thread locking or "stretch time" – thread locking is something that keeps a task from completing; it makes a task wait – usually because a needed resource is not available. Stretch time is the amount of time the task waited on something, maybe another task, or a higher priority request.

File activity is measured in I/O requests and the time taken to satisfy the requests. When data is not in main storage it must be retrieved from another device – this is an I/O request, commonly called "reads" and writes". The CPU usually waits for a device to process the request, so the processing of an I/O request can be included in the system CPU time but much is on the device itself and is attributed as "Wait time" to the task.

Database activity is file activity to a Database management system, like DB2. It has requests in the form of reads, writes, updates, browses, and has wait time. Because most DBM Systems execute in their own regions a picture of translation activity for the DBMS must be acquired from an outside tool or monitor. The way DB indexes are created and populated can affect the time to retrieve data and thus the elapsed time to process.

Throughput is the "name of the game" most times. It is the desired effect, but is really a composite of the above measures. When trying to improve throughput, one usually has to observe how locks and wait time and stretch time and load interact. Sometimes a faster CPU "frees up" tasks sooner and eliminates or reduces elapsed time significantly and hence improves throughput. Sometimes increasing the number of threads improves throughput – but also reducing the number of threads (i.e. contention) improves throughput.



Techniques that can be used to improve performance include the following:

1) For CPU time
   A) search algorithms for table lookups – are they always sequential, would a binary search technique, or weighted entry technique apply?
   B) Is the processing paradigm the best? Are there extra or unnecessary steps that could be eliminated
2) For main storage or memory usage
   A) Are repetitively acquired storage areas a candidate for a “storage pool technique”, like the Abstract Message Model nodes of WDI? Reusing nodes from a pool saves acquisition or initialization cycles.
   B) How are memory buffers handled? Are there upper limits set, what is the technique for increasing buffers to accommodate data variations?
   C) If a progressive reallocation of buffers is used, are they released so that the maximum size buffer is not in used at all times?
3) For Data Access
   A) Are DB2 indexes used appropriately and are they reorganized regularly?
   B) Is there any main storage caching of frequently used objects, e.g. Trading Partners, Map Control Strings
   C) How are work files defined, and are they appropriate for the platform, i.e. are RAM work files used on AIX?
4) Throughput
   A) Are DB2 Table locks used appropriately to single thread where required and released in a timely way? Are locks or ENQ/DEQ sued at all?

Conclusions

- Performance tuning is not straightforward or obvious, the car with the biggest engine doesn't always win the race
- Tools that provide information are needed
- WDI attempts to use a number of performance techniques to make it perform optimally
  - Some of these are "built-in"
  - Others are controllable by users as keyword options
  - Using or over-using some options may degrade performance
- Form a theory and test it

In conclusion, there are several considerations when determining performance. Tuning is not always as obvious as one might expect. Make sure you understand the variables that control the performance measures you are trying to improve. Use appropriate tools to gather the data – machine or system monitors for CPU time , main storage usage, device wait time, or DBMS monitors for DB statistics.

WDI uses many of the techniques associated with good performance techniques. Most of these are not under user control, but some are. Utilize the ones that you can for DB indexes or work files and make sure you provide sufficient main storage to WDI regions. WDI will use as much CPU as available to translate if sufficient memory is available to house input and output data and work objects.

Certain options allow WDI to process data to improve performance by processing data more efficiently, particularly for large messages. For example, pageable AMM and other options allow WDI to keep parts of large messages on disk instead of in memory while processing the data. However, these options should be used with care, since for smaller messages they may actually adversely effect performance or may impose restrictions on the input data.

Tuning takes insight. The best results are obtained when performance tuning objectives are established, a theory is conceived, and then results are captured as variables are manipulated. Remember, there are no failures in testing a theory – even if the theory is proven to be "not true" it still increases understanding.

Trademarks, copyrights, and disclaimers

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

IBM	CICS	IMS	WMQ	Tivoli
IBM(logo)	Cloudscape	Informix	OS/390	WebSphere
e(logo)business	DB2	iSeries	OS/400	xSeries
AIX	DB2 Universal Database	Lotus	pSeries	zSeries

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds.

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

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or program(s) described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

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 and IBM Corp.