Redbooks Paper

Ruzhu Chen
Lerone Latouche

Network Performance Evaluation on an IBM e325 Opteron Cluster

Network performance evaluation is an important measurement that is used to adjust and tune up cluster systems for high-performance computing. The IBM® e325, the first AMD Opteron processor-based server from IBM, is an outstanding computing power designed for HPC applications. This report summarizes the overall performance of all the communication networks (100 Mb/s Ethernet, Gigabit Ethernet (GigE), and Myrinet) that are available on a 16-node Linux cluster. The results indicate that changing the adapter's default parameters and environment variables could significantly influence network performance. The Myrinet performance is, by far, the best in terms of high bandwidth and low latency, while the tunable GigE is better than untunable GigE. Network communications using MPI suffered certain performance loss compared to raw TCP rates.

Introduction

Network communication performance is a critical parameter to evaluate when considering the overall performance of a newly built, high-performance computing cluster. Performance measurement and tuning of the network applications is thus necessary to ensure the optimal performance for the cluster. The tuning of network applications such as different MPI layers, system parameters such as MTU (maximum transmission unit) and TCP/IP buffer sizes, and other environment variables can help the network realize its highest potential by minimizing performance loss due to the parameter settings. The measurement of different network interfaces helps to determine the better price/performance network adaptors. NetPIPE (Network Protocol Independent Performance Evaluator) is one of the tools that provides comprehensive evaluation of communication performance between processors or nodes [Note 1 on page 7]. NetPIPE measures the point-to-point communication at both MPI and TCP layers. NetPIPE is now used for comparing different network interfaces, comparing different message-passing libraries, determining the performance loss, and optimizing the memory copy routine [Notes 2, 3, and 5 on page 7]. In preparing this report, we used NetPIPE to evaluate our IBM e325 cluster built with Myrinet, Gigabit Ethernet, and 100 Mb/s Ethernet.

The IBM e325, the first AMD Opteron processor-based server from IBM, is an outstanding computing power designed for high-performance computing. The e325 is a 1U, 2-way rack-optimized server that is ideal for customers who need leading 32-bit performance with an easy migration path to 64-bit computing at an affordable price. The e325 server features Opteron processors that incorporate an integrated memory controller and Hyper-Transport technology to help eliminate memory access and I/O bandwidth bottlenecks [Note 4 on page 7]. The system on which the tests were done is a 16-node IBM e325 Linux cluster with each node having dual 2.0 GHz AMD Opteron processors, 2 GB memory, Myrinet D card, Gigabit Ethernet, and 100 Mb/s Ethernet. In addition, the software stack consists of SuSE SLES 8.0 for x86-64, Portland Group Compiler (PGI) 5.0-2, and the GNU GCC-3.3 compilers. The MPI compiler MPICH-1.2.5 was built with either PGI compiler linking with GM-2.0.8 library for Myrinet (GM-MPICH) or only PGI compiler for Gigabit Ethernet (public MPICH).

Measurement

Three network interface cards (NICs)—Myrinet, Gigabit Ethernet, and Ethernet cards—were utilized to study the performance. Myrinet and GigE cards were associated with GM-2.08 and Broadcom BCM5700-6.2.17 Linux drivers, respectively. Two types of configurations of GigE, tunable and untunable, were tested. A detailed explanation of the parameters is attached at the end of this report (see “Appendix” on page 8). The MPI-based NetPIPE was compiled by MPICH compiler mpicc with default compiler flags in makefile, and the TCP-based was compiled by gcc with default compiler options. All of the compilations generated 64-bit executables. Example 1 shows the scripts that were used to run the benchmark.

Example 1 The scripts used to run the benchmark

```
#Running MPI version on GigE and Ethernet
export P4_SOCKBUFSIZE=??? # Set the variable to 256000 to 512000
mpirun -p4pg hf -np 2 NPmpi
# Running MPI version on Myrinet
np 2 Npmpi
#Running TCP version on all the interfaces
#Start server on different networks
–b [buffersize]
#Running client on different networks
–b [buffersize]
```

Each NetPIPE benchmark test was repeated multiple times to obtain comparable results. All latencies were obtained by measuring the round-trip time divided by 2 for messages smaller than 64 bytes [Notes 1 and 3 on page 7]. All benchmark tests reflect performance results for the Linux kernel 2.4.19 (NUMA) with SuSE SLES 8.0 Service Pack 2.

Cross-nodes performance

Here we show cross-nodes performance, followed by single node performance.

Performance across two nodes

Figure 1 on page 3 shows the performance throughputs versus transferred message size (bits) across two network interface cards. Of all of the results, the Myrinet card achieved best performance of nearly 1900 Mbps throughput. The NetPIPE MPI module showed stable results on all NICs. The TCP module displayed irregular curves over Myrinet and GigE with

untunable configurations, and smooth curves were shown over Ethernet and tunable GigE cards.

To determine the effective (peak) performance, NetPIPE TCP tested with different socket buffer sizes ranging from 16 K to 128 K (-b flag). Performance loss was observed on the Myrinet card using lower socket buffer size but reached optimal rate at a 64 K buffer size, even though a performance dip occurred. The performance instability on untunable GigE indicated that some system parameters were not set properly.

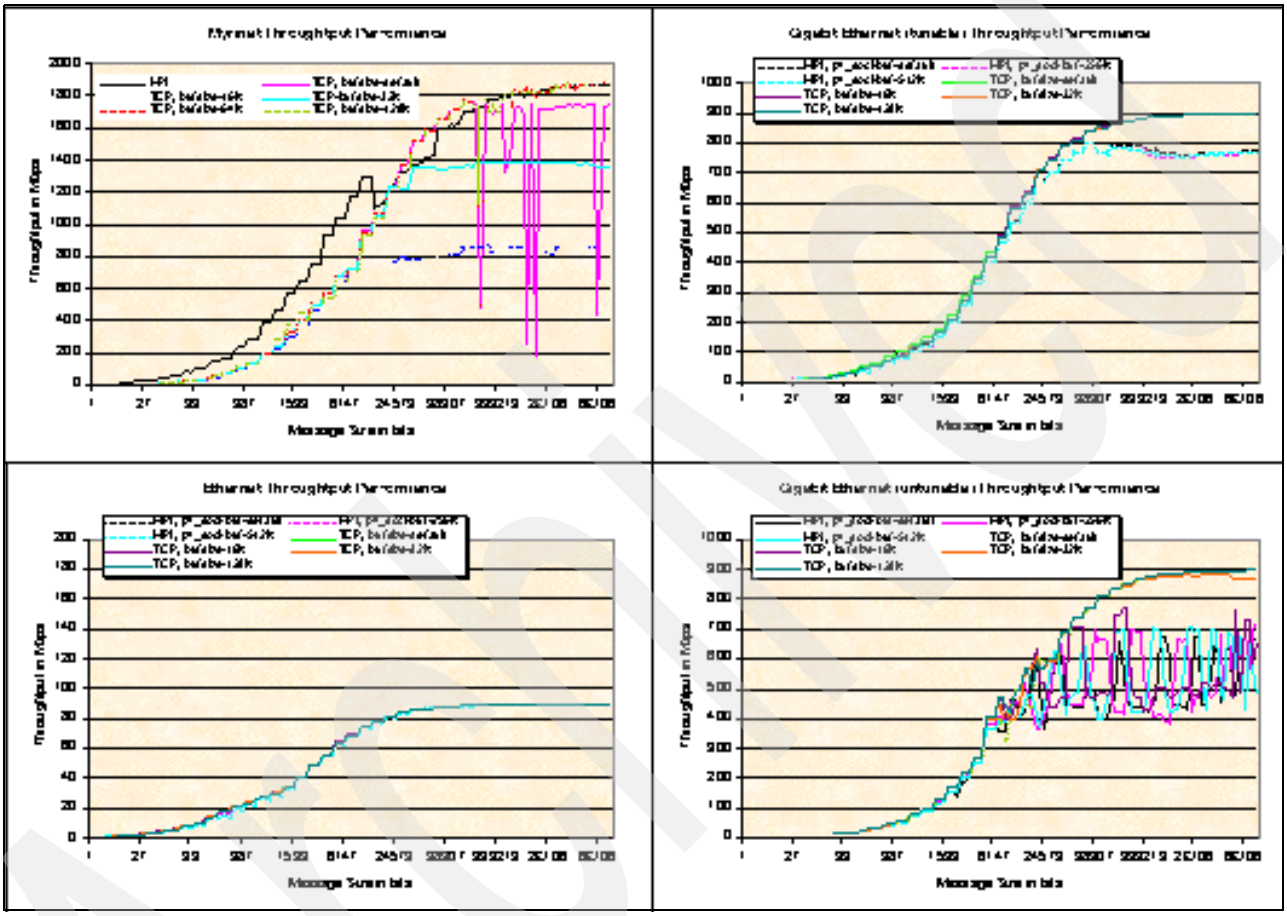

Figure 1 Network performance throughputs across network interface cards

MPI versus TCP performance

Network message transmission over MPI, which was implemented with TCP, often suffers performance loss compared to the raw TCP results [Note 3 on page 7]. Figure 2 on page 4 shows the performance overhead between MPI and TCP layers. In the MPI tests over Ethernet and Gigabit Ethernet, the environment variable P4_SOCKBUFSIZE was optimized. The results indicate no overhead on the Myrinet and 100 Mb/s Ethernet, but there is about 10-15% loss on Gigabit Ethernet. Compared to previous results that were obtained on 1.8 GHz Pentium® 4 systems connecting with Gigabit Ethernet [Note 3 on page 7], the AMD cluster suffers less overhead when using MPICH libraries.

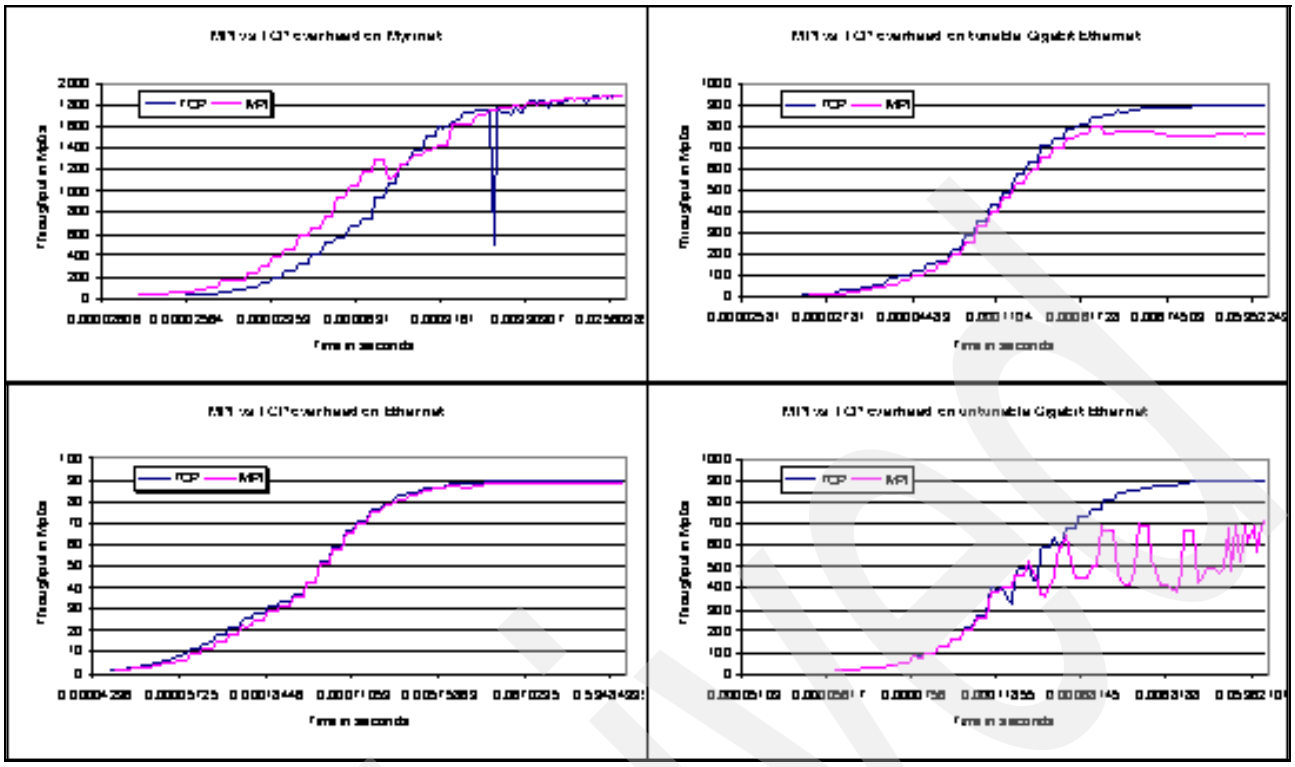

Figure 2 MPI versus TCP performance overhead on all NICs across nodes

Single node performance

Even though the network performance of the cluster is determined mainly by the cross-node performance, it would be interesting to see the NetPIPE benchmark test results in the same node over different NICs.

Network performance in the same node

Figure 3 on page 5 displays the test results on all of the NICs that are being tested between two processors in the same node. Throughputs of raw TCP tests were nearly the same on all of the network cards. Both MPI and TCP displayed relatively stable performance and no performance dip was observed.

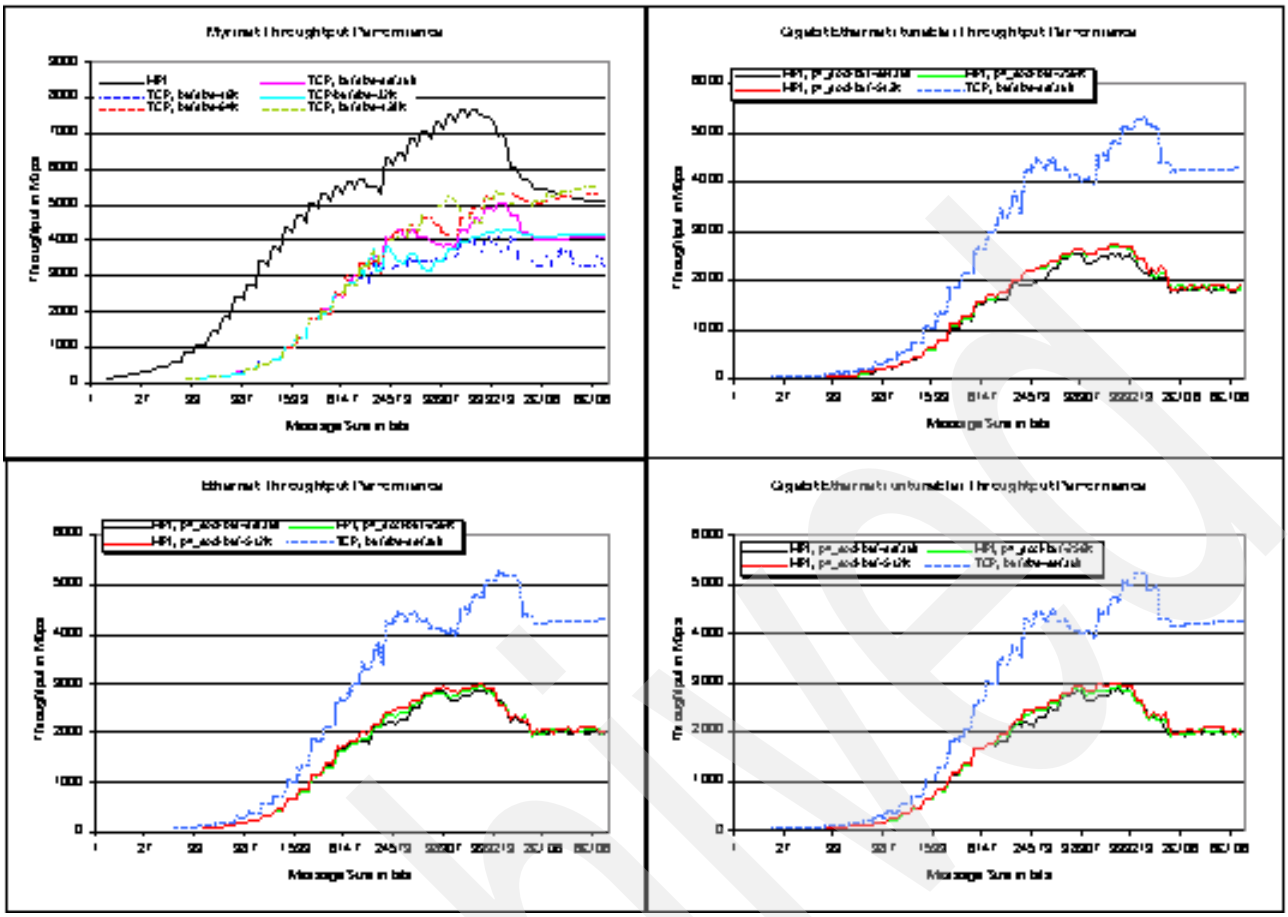

Figure 3 Network throughputs between processors in the same node

The MPI performed much better than TCP on the Myrinet card, but suffered about 50% loss on all of the Ethernet cards; this is shown in Figure 4 on page 6.

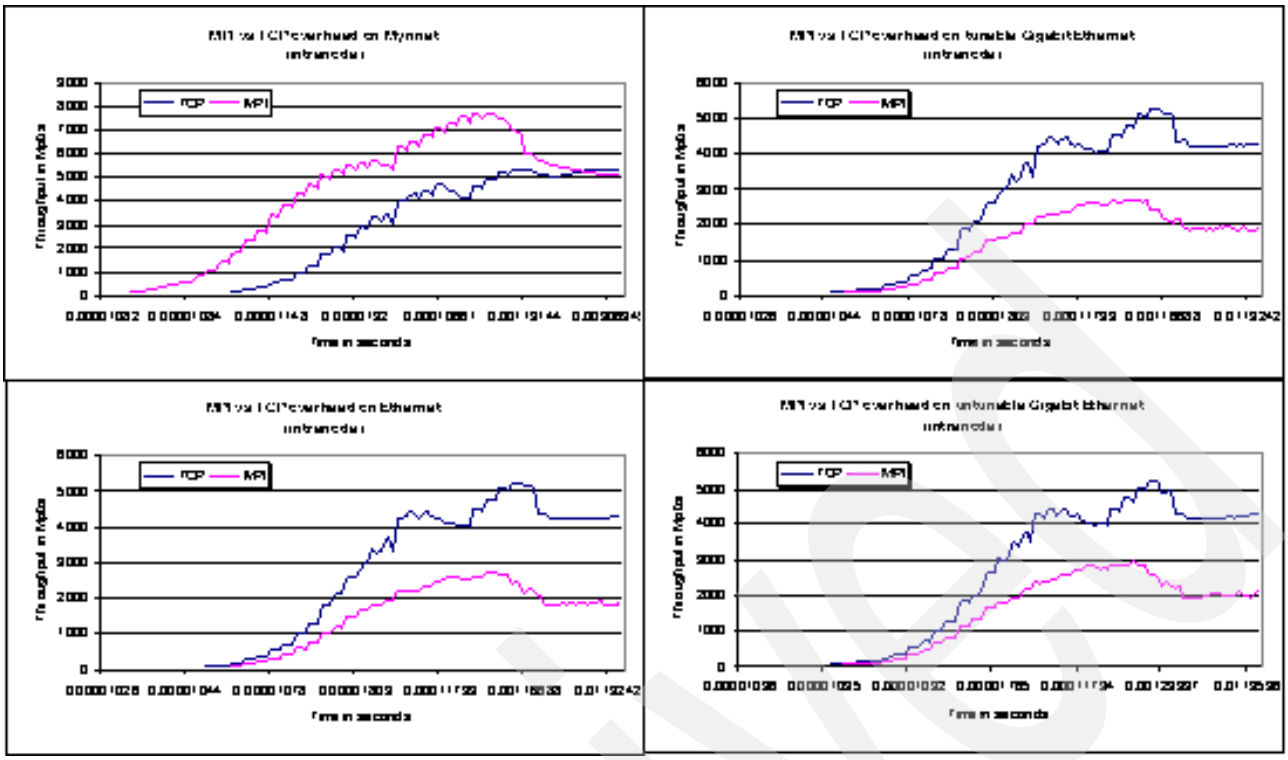

Figure 4 MPI versus TCP performance overhead on all of the NICs in the same node

Overall performance

The performance of the Myrinet, Gigabit Ethernet, and 100 Mb/s Ethernet are summarized in Table 1. Communication over MPI on the Myrinet card has the lowest latency for inter-processor communication, and little overhead was observed when compared to raw TCP tests. Our results also showed that network communication using Myrinet and the current version of GM driver gave higher bandwidth and latency at both MPI and TCP layers on our AMD e325 cluster.

For cross-node communication, the effective bandwidth and latency over MPI suffer 10% to 15% performance loss on a Gigabit Ethernet card with tunable configuration and more than 20% on the untunable configuration. For intranode communication, about 50% performance loss was experienced using MPI on either Ethernet card. MPICH uses a p4-library to interface with TCP on Gigabit Ethernet and p4 receives all messages to the buffer rather than directly to the memory. Thus, MPICH has to use **`memcpy`** to move messages out of the buffer, causing the performance loss for large messages [Note 3 on page 7]. With proper configurations, Gigabit Ethernet was able to reach half of Myrinet's bandwidth, but with much higher latency. Our results indicate that the performance of all NICs still has room for improvement via additional tuning, even though the results have reached 80% to 90% of the theoretical peak values.

Table 1 Summary of performance of all the networks on the IBM e325 cluster

Internode	Network type		Max. bandwidth (Mbps)	Message size (bits)	Latency (µsec)
	Myrinet	MPI	1876	8388608	3.98
		TCP	1880	8388605	14.06
	GigE (tunable)	MPI	788	262144	17.53
		TCP	897	8388611	13.91
	GigE (untunable)	MPI	700	524291	30.94
		TCP	896	8388608	26.18
	Ethernet	MPI	89	8388605	50.62
		TCP	90	8388611	38.00
Intranode	Myrinet	MPI	7688	262141	0.42
		TCP	5328	524291	5.44
	GigE (tunable)	MPI	2554	393216	8.66
		TCP	5233	524291	5.18
	GigE (untunable)	MPI	2939	393219	7.74
		TCP	5208	524285	5.28
	Ethernet	MPI	2863	262147	7.75
		TCP	5250	524285	5.22

Reference notes

1. *NetPIPE: A Network Protocol Independent Performance Evaluator* by Q.O. Snell, A.R. Mikler, and J.L. Gustafson

   http://www.scl.ameslab.gov/netpipe/paper/full.html

2. *Performance Evaluation of Copper-based Gigabit Ethernet Interfaces* by P. Gray and A. Betz

   http://rtlin1.dl.ac.uk/gridmon/docs/Performance%20Evaluation%20of%20Copper-based%20Gigabit%20Ethernet%20Interfaces.pdf

3. *Protocol-Dependent Message-Passing Performance on Linux Clusters* by D. Gray and X. Chen

   http://www.scl.ameslab.gov/netpipe/np_cluster2002.pdf

4. *IBM eServer Linux Clusters White Paper*

   http://www-1.ibm.com/servers/eserver/clusters/whitepapers/linux_wp.pdf

5. *Integrating New Capabilities into NetPIPE* by D. Turner. A. Oline, X. Chen, and T. Benjegerdes

   http://www.scl.ameslab.gov/netpipe/np_euro.pdf

Appendix

Gigabit Ethernet Tunable Parameters that are recommended by Broadcom for minimal latency are:

```
options bcm5700 adaptive_coalesce=0,0 rx_coalesce_ticks=1,1 rx_max_coalesce_frames=1,1
tx_coalesce_ticks=1,1 tx_max_coalesce_frames=1,1
```

adaptive_coalesce

Enables or disables adaptive adjustments to the various interrupt coalescing parameters. Enabling it allows the driver to dynamically adjust the interrupt coalescing parameters to achieve high throughput during heavy traffic and low latency during light traffic. rx_std_desc_cnt (and rx_jumbo_desc_desc_cnt if using jumbo frames) should not be set less than 50, and tx_pkt_desc_cnt should not be set less than 80 when this parameter is enabled. The valid values are: 0 – disabled, 1 – enabled (default).

rx_coalesce_ticks

Configures the number of 1 usec ticks before the NIC generates receive interrupt after receiving a frame. This parameter works in conjunction with the rx_max_coalesce_frames parameter. Interrupt will be generated when either of these thresholds is exceeded. 0 means this parameter is ignored and interrupt will be generated when the rx_max_coalesce_frames threshold is reached. The valid range is from 0 to 500, and the default is 80. This parameter is not used and will be adjusted automatically if adaptive_coalesce is set to 1.

rx_max_coalesce_frames

Configures the number of received frames before the NIC generates receive interrupt. The valid range is from 0 to 100; the default is 15. This parameter and rx_coalesce_ticks both cannot be 0, otherwise no receive interrupts will be generated. It should also be set significantly lower than rx_std_desc_cnt (and rx_jumbo_desc_cnt if using jumbo frames). This parameter is not used and will be adjusted automatically if adaptive_coalesce is set to 1.

tx_coalesce_ticks

Configures the number of 1 usec ticks before the NIC generates transmit interrupt after transmitting a frame. This parameter works in conjunction with the tx_max_coalesce_frames parameter. Interrupt will be generated when either of these thresholds is exceeded. 0 means that this parameter is ignored and interrupt will be generated when the tx_max_coalesce_frames threshold is reached. The valid range is from 0 to 500, and default is 200. This parameter is not used and will be adjusted automatically if adaptive_coalesce is set to 1.

tx_max_coalesce_frames

Configures the number of transmitted frames before the NIC generates transmit interrupt. The valid range is from 0 to 100, and the default is 35. This parameter and tx_coalesce_ticks both cannot be 0, otherwise no transmit completion interrupt will be generated. This parameter should always be set lower than tx_pkt_desc_cnt. This parameter is not used and will be adjusted automatically if adaptive_coalesce is set to 1.

About the authors

Ruzhu Chen (mailto:ruzhuchen@us.ibm.com) is an Advisory Engineer/Scientist at IBM pSeries® and HPC Benchmark Center. He joined the IBM HPC benchmarking team in 2001 and worked on benchmarking and performance analysis of scientific and technical applications on IBM pSeries and xSeries® servers. He holds a Ph.D. in life sciences and a

Master of Computer Science degree from the University of Oklahoma, where he also performed two years of postdoctoral research on gene design and cloning before joining IBM.

Lerone Latouche (mailto:latouche@us.ibm.com) is an Advisory Software Engineer at the IBM xSeries Benchmark Center and On-Demand Infrastructure. Prior to joining IBM, he served eight years in the United States Army as an electrician. He is a Certified Specialist in IBM pSeries AIX® Systems and a Red Hat Certified Engineer (RHCE). His areas of technical expertise include building clusters for high-performance computing and technical applications on IBM pSeries and xSeries servers. He holds a Bachelor of Science degree in Computer Science from St. John's University in Jamaica, New York, and a Master of Science degree in Computer Science from Marist College in Poughkeepsie, NY.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. 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.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

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.

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

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

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

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 and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

This document created or updated on July 28, 2004.

Send us your comments in one of the following ways:

- Use the online **Contact us** review redbook form found at:
  ibm.com/redbooks
- Send your comments in an e-mail to:
  redbook@us.ibm.com
- Mail your comments to:
  IBM Corporation, International Technical Support Organization
  Dept. HYJ Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400 U.S.A.

Trademarks

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

AIX®	pSeries®	xSeries®
IBM®	Redbooks (logo) ™	

The following terms are trademarks of other companies:

Pentium is a trademark of Intel Corporation in the United States, other countries, or both.

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