Redpaper

Xinghong He
Anthony Ercole

Windows Compute Cluster Server IMB Benchmarks

In 2006, Microsoft® released its first high-performance computing (HPC) software offering, the Windows® Compute Cluster Server 2003 (WCCS 2003). Two years later, a new improved version was released and it was renamed as Windows High Performance Computing Server 2008 (WHPCS 2008) to increase its visibility in the HPC marketplace, which had been dominated by Linux® and UNIX® servers. Windows HPC is gradually gaining attention in the HPC community with a series of events. These include the November 2008 Top 500 list that contains five Windows systems. One of them is in the tenth position in the list.

In this IBM® Redpapers publication we present some Intel® MPI Benchmarks (IMB) on an InfiniBand® 4x SDR cluster of IBM System x® 3550 systems running WCCS 2003. For comparison purposes, we have also performed Linux runs on a similar system: A System x3650 with InfiniBand 4x SDR interconnects. The System x3650 has the same processor and memory configuration as the System x3550.

Benchmarking systems

Two systems were used for this work:

- Windows cluster running WCCS 2003 (hereafter referred to as WCCS system or WCCS)
- Linux cluster running Red Hat Enterprise Linux 4.4 (Linux system or Linux)

WCCS system

The WCCS system consists of the following components (Figure 1):

- One head node: This node is an x3650 system with two dual-core Intel Xeon® 5160 processors running at 3.0 GHz. It has 32 GB memory of PC2-5300 DDR AMF ECC DRAM and 1333 MHz FSB. This node is on three networks: one public network for user access and two private networks for computing (InfiniBand SDR) and cluster management (Gigabit Ethernet).
- 64 x3550 compute nodes: Each has two dual-core Intel Xeon 5160 processors running at 3.0 GHz, 8 GB PC2-5300 ECC Chipkill DDR2 FBDIMM SDRAM, 1333 MHz FSB. The compute nodes are connected by the two private networks: the Gigabit Ethernet for cluster management and the InfiniBand SDR for HPC.
- One Gigabit Ethernet switch and one InfiniBand SDR switch.
- Microsoft Windows Server® 2003 R2 Standard x64 Edition Service Pack 2 on all nodes.
- Intel Cluster Toolkit Compiler Edition (ICTCE) 3.1.1.002 installed on the head node and all compute nodes.
- Windows OpenFabrics 1.0 driver for IB on all compute nodes.

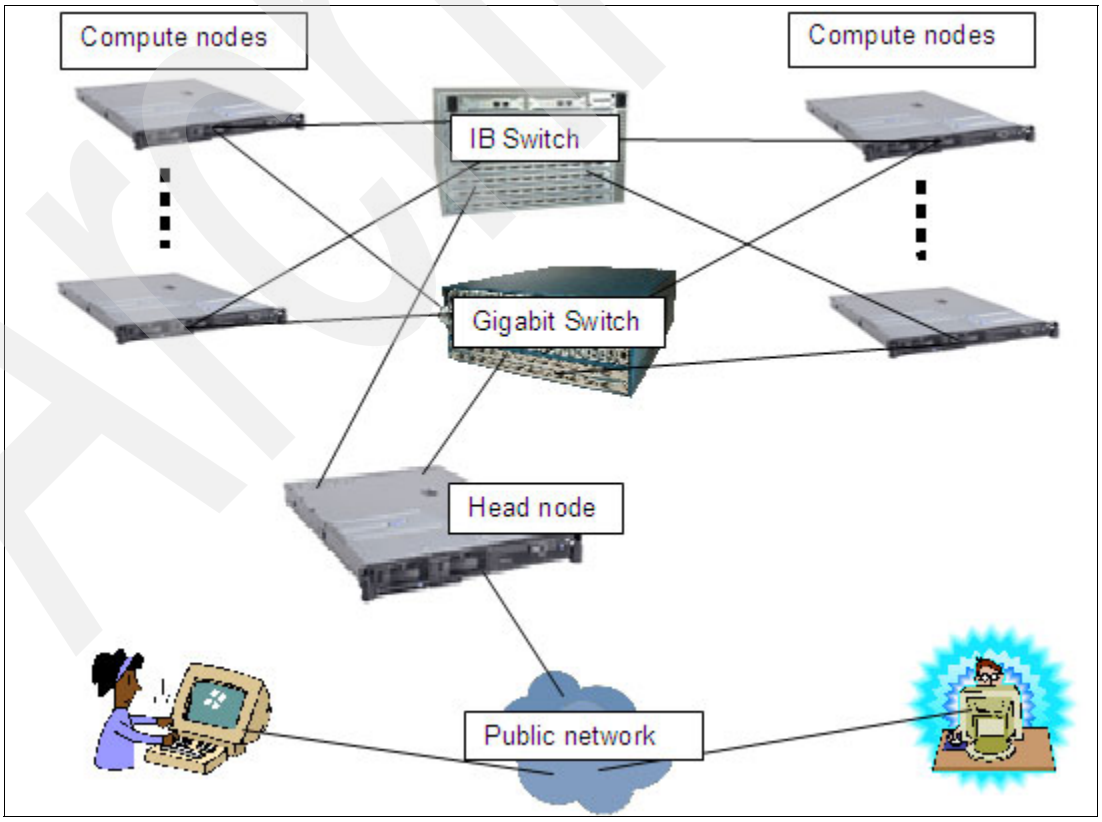

Figure 1 The WCCS system

Linux system

The Linux system has a similar structure as the WCCS system, except that there is no designated head node and every compute node is on both private (IB) and public (GigE) networks. The major components are:

- Four x3650 nodes, each with two dual-core Intel Xeon 5160 processors running at 3.0 GHz. Three of them (lcan71, lcan72, lcan74) have 24 GB of memory and the rest (lcan73) have 32 GB.
- Voltaire SDR switch ISR 9096.
- Red Hat Enterprise Linux AS release 4 (Nahant Update 4), kernel 2.6.9-42.ELsmp.
- Intel compiler 10.1.013.
- Voltaire MPI with IBHOST 3.5.5.

Because of the size (four nodes) of this system, we have limited our WCCS system to four nodes for direct comparison.

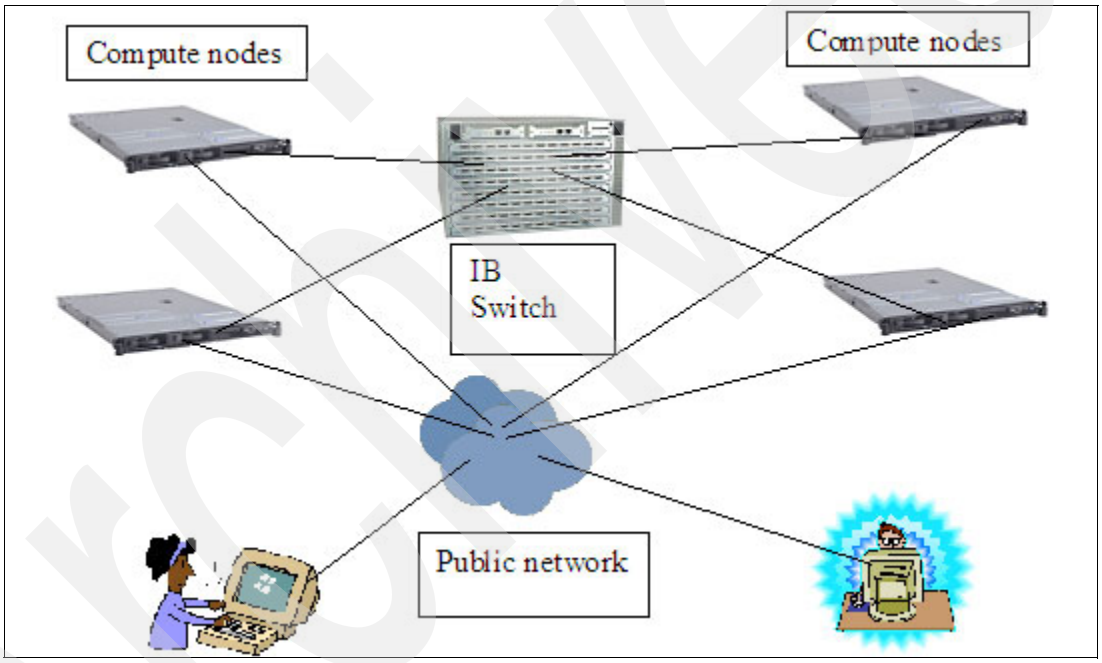

Figure 2 The Linux system

Results

In this section we discuss the Intel MPI Benchmarks 3.1 results for the WCCS and the Linux systems.

IMB PingPong and Multi-PingPong within one node

Table 1 shows four sets of IMB PingPong results on a single node, determined by the underlying OS (WCCS versus Linux) and the processors used (on the same socket versus on different sockets). The four sets are:

- Linux system, processors from the same socket
- Linux system, processors from different sockets
- WCCS system, processors from the same socket
- WCCS system, processors from different sockets

We can see that PingPong performs much better on the same socket than on separate sockets for both WCCS and Linux systems and lower latencies and higher bandwidths. We also notice that WCCS generally performs better than Linux, probably due to differences in MPI implementations (Intel MPI versus Voltaire MPI).

Table 1 shows an IMB 3.1 PingPong on a single Linux node (columns 3–6) and a single WCCS node (columns 7–10). MPI process binding is used. Here p0-p1 means that PingPong communications happen between processors 0 and 1, which are on the same processor socket. p0-p2 means processors 0 and 2 are used in the communication. These two processors are on separate sockets.

Table 1 IMB PingPong

#bytes	#repetitions	Linux p0-p1 t[usec]	MBps	p0-p2 t[usec]	MBps	WCCS p0-p1 t[usec]	MBps	p0-p2 t[usec]	MBps
	1000	0.38		0.84		0.21		0.7	
	1000	0.35	2.76	0.85	1.12	0.28	3.35	0.96	0.99
	1000	0.34	5.6	0.85	2.25	0.27	7.19	0.89	2.15
	1000	0.33	11.63	0.84	4.51	0.27	14.06	0.9	4.25
	1000	0.34	22.64	0.86	8.92	0.26	28.81	0.92	8.32
16	1000	0.36	42.74	0.88	17.26	0.27	56.52	0.86	17.83
32	1000	0.36	85.13	0.89	34.25	0.27	111.51	0.88	34.81
64	1000	0.4	151.45	0.98	62.41	0.3	200.54	1.01	60.33
128	1000	0.44	275.87	1.09	111.99	0.32	384.54	1.05	115.73
256	1000	0.55	443.89	1.34	182.74	0.36	670.39	1.25	195.61
512	1000	0.58	841.86	1.48	330.48	0.44	1117.18	1.63	299.83
1024	1000	0.78	1259.27	1.87	522.65	0.53	1834.68	2.39	407.76
2048	1000	1.11	1761.95	2.72	718.46	0.79	2470.77	4.01	486.96
4096	1000	1.69	2317.56	4.47	874.66	1.31	2971.24	7.25	538.8
8192	1000	3.03	2581.36	8.01	974.92	2.41	3240.57	13.49	579.19
16384	1000	5.44	2874.62	15.39	1015.27	5.49	2848.52	15.41	1014.25
32768	1000	8.96	3487.72	30.75	1016.41	7.52	4154.38	25.87	1208.07

65536	640	16.62	3760.64	61.24	1020.53	11.57	5402.96	46.34	1348.67
131072	320	35.29	3542.49	121.52	1028.65	20.92	5975.28	58.56	2134.42
262144	160	85.87	2911.53	241.83	1033.79	41.88	5969.47	113.63	2200.06
524288	80	315.64	1584.06	495.53	1009.03	85.38	5856.22	294.55	1697.52
1048576	40	736.05	1358.6	724.73	1379.83	353.95	2825.29	578.76	1727.83
2097152	20	1468.5	1361.98	1445.15	1383.94	847.28	2360.49	1395.72	1432.95
4194304	10	2973.1	1345.4	2933.8	1363.42	2151.97	1858.77	2876.78	1390.45

To picture the performance comparisons, Figure 3 shows the bandwidths of PingPong (using data from Table 1 on page 4) in a logarithmic scale.

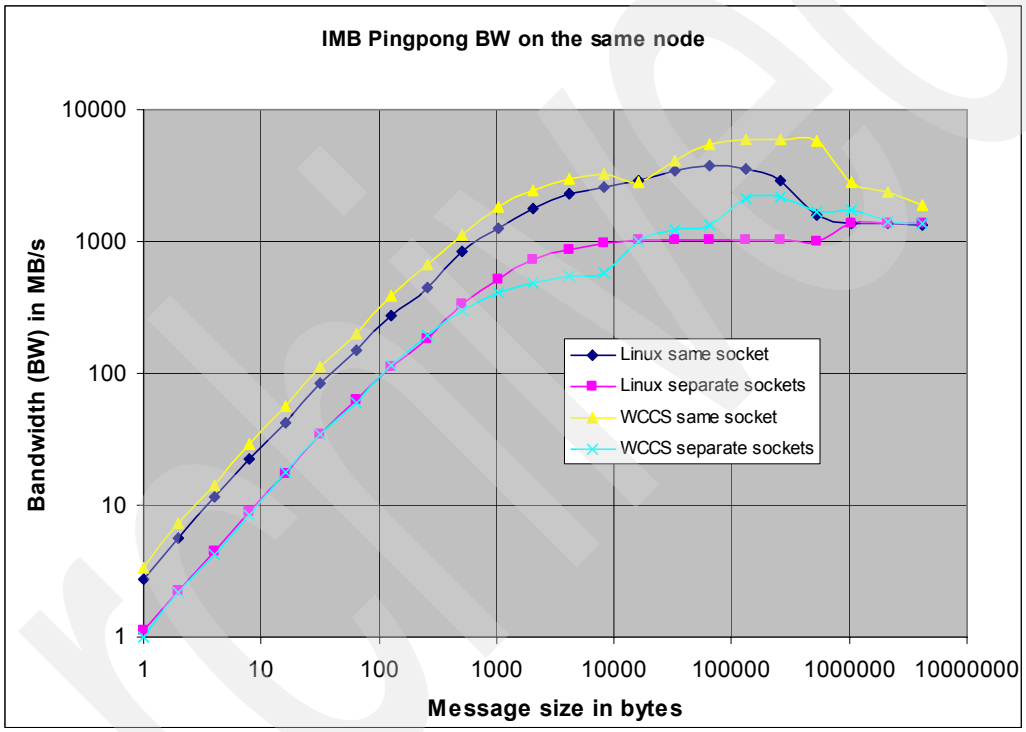

Figure 3 IMB PingPong bandwidths on a single node

The process binding is done through the **taskset** command on the Linux system. On the WCCS system, this is done by the setting of Intel MPI run-time environment variable.

Figure 4 compares Multi-PingPongs on a single node. In this test, there are two communication pairs, each doing MPI PingPong communication. Again, we do process binding to control how the processors are paired up. In Figure 4, the *same socket* means that communication happens within each of the two sockets—one communication pair per socket. *Across sockets* means that processors from separate sockets are paired up for PingPong communications.

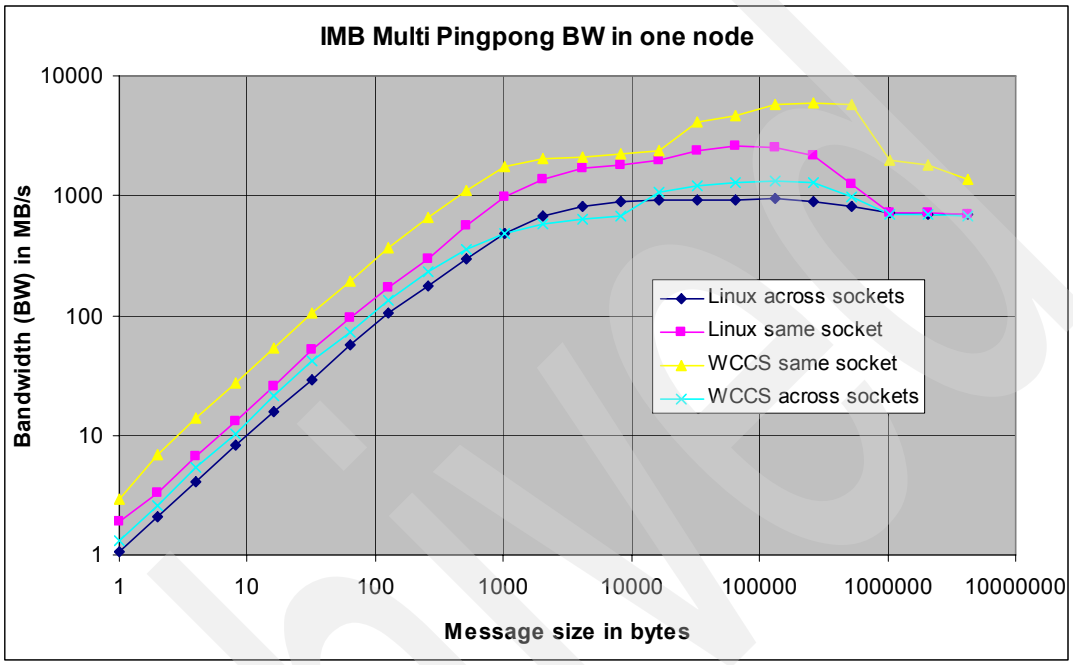

Figure 4 IMB Multi-PingPong bandwidths on the same node

In our Linux case, processors 0 and 1 are on one socket and processors 2 and 3 are on the other socket. Therefore, the *same socket* case corresponds to communication pairs of processors (0,1) and (2,3), while *across sockets* corresponds to (0,2) and (1,3).

IMB PingPong and Multi-PingPong across two nodes

To see how WCCS does MPI communication across nodes, we present some WCCS and Linux communication comparisons in Figure 4 through Figure 8 on page 9.

Figure 5 is a comparison of WCCS and Linux PingPong communications on two different nodes. In this case, there is one communication pair between two nodes. We can see that Linux performs slightly better than WCCS.

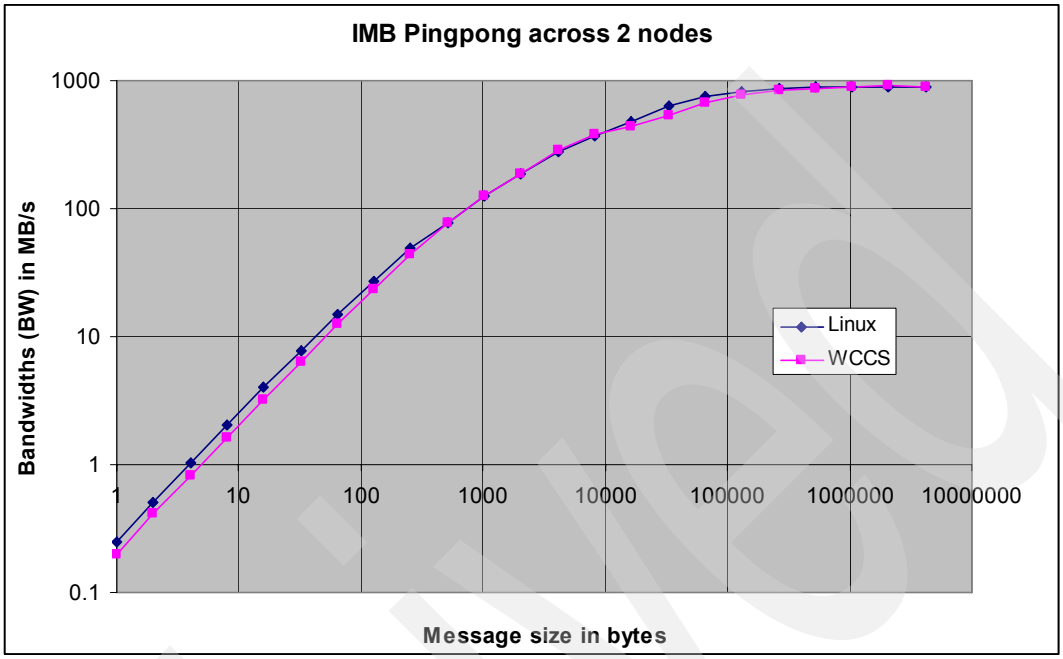

Figure 5 WCCS and Linux comparison of IMB PingPong across two nodes

Figure 6 on page 8 and Figure 7 on page 8 show two-pair and four-pair Multi-PingPong MPI communications between two nodes. In these two cases, there is certain level of communication stress across the IB network and memory stress within each of the two nodes. For small messages less than 1 KB, Linux is slightly better. WCCS is generally better than Linux for larger messages.

In Figure 6 the total number of MPI tasks is four. There are two communication pairs between two nodes.

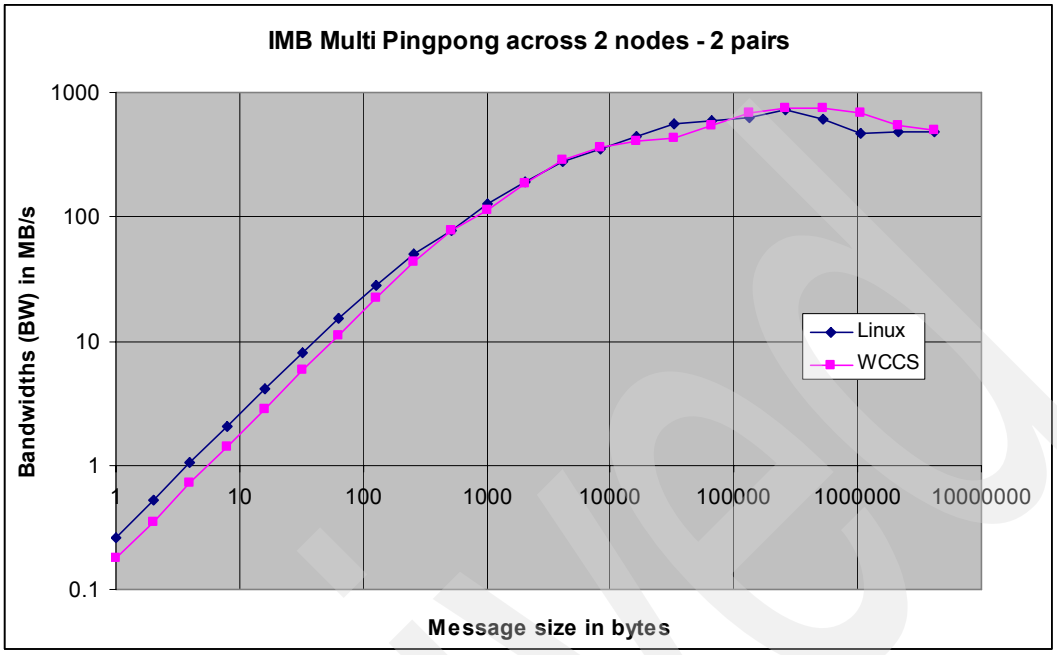

Figure 6 WCCS and Linux comparison of IMB Multi-PingPong across two nodes

In Figure 7 the total number of MPI tasks is eight. There are four communication pairs between two nodes.

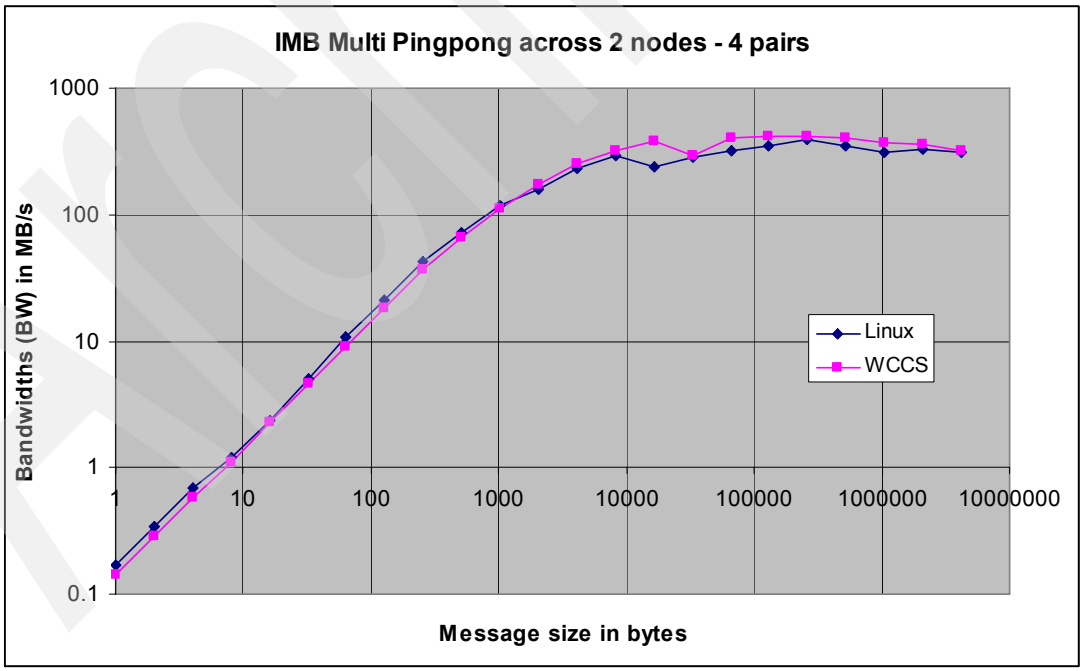

Figure 7 WCCS and Linux comparison of IMB Multi-PingPong across two nodes

In order to see the difference more clearly in the large message range, Figure 8 shows the communication bandwidths of all three cases:

- PingPong
- Two-pair Multi-PingPong
- Four-pair Multi-PingPong

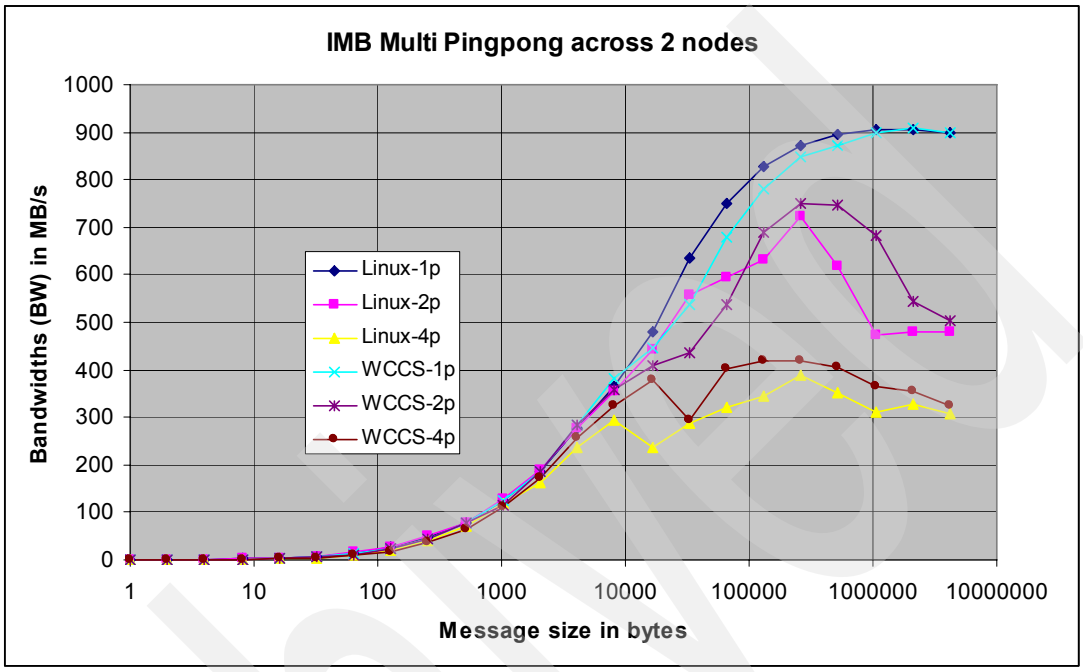

Figure 8 PingPong and Multi-PingPong across two nodes

In Figure 8 we can see that Linux performance is better for PingPong communication across two nodes. However, when there is stress in memory and network stress such as the two-pair and four-pair Multi-PingPong communications, WCCS scales better. As we discussed before, the performance difference could be from the MPI implementation, rather than the underlying operating systems.

Conclusion

We performed IMB measurements on a WCCS cluster and a Linux cluster with similar hardware (x3550 versus x3650; both are Intel Xeon x5160 processor based systems). WCCS with Intel MPI implementation performs better than Linux with Voltaire MPI for communications within the same compute node (the shared memory communication performance). For communications across compute nodes that are inter-connected by InfiniBand SDR switch, the Linux with Voltaire MPI performs slightly better than WCCS with Intel MPI.

The authors of this Redpaper publication

Xinghong He and Anthony Ercole are with IBM STG Sales Support and Education in Poughkeepsie, New York.

Xinghong He is an IBM Certified Consulting IT Specialist at IBM WW HPC Benchmark Center. He has over 15 years of experience in parallel and high-performance computing. He holds a Ph.D. in theoretical and computational physics from Queens University of Belfast. His areas of expertise include applications porting and performance tuning for System p®, System x, and Bluegene.

Anthony Ercole is a Systems Management Specialist at the STG WW Power Systems® (p) Benchmark Center in Poughkeepsie, NY. His areas of expertise include System p, System x, AIX®, Windows, Linux, and Provisioning.

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 illustrate 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.

This document REDP-4494-00 was created or updated on January 23, 2009.

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

Trademarks