

# Evolution of the IBM System Blue Gene Solution

A new generation  
of hardware

Additional software  
functionality

Enhanced control  
system software



Gary Lakner  
Carlos P. Sosa





International Technical Support Organization

**Evolution of the IBM System Blue Gene Solution**

February 2008

**Note:** Before using this information and the product it supports, read the information in “Notices” on page iii.

**First Edition (February 2008)**

This edition applies to Version 1, Release 3, Modification 3 of the IBM System Blue Gene/L Solution and Version 1, Release 1, Modification 2 of the IBM System Blue Gene/P Solution.

This document was created or updated on February 14, 2008.

© Copyright International Business Machines Corporation 2008. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

# 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.

## Trademarks

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

Blue Gene/L™  
Blue Gene/P™  
Blue Gene®  
DB2®

eServer™  
IBM®  
iSeries®  
LoadLeveler®

PowerPC®  
POWER™  
Redbooks®  
Redbooks (logo) ®

The following terms are trademarks of other companies:

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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

# Contents

<b>Notices</b> .....	iii
Trademarks .....	iv
<b>Preface</b> .....	vii
The team that wrote this paper .....	vii
Become a published author .....	viii
Comments welcome .....	viii
<b>Chapter 1. Hardware changes</b> .....	1
1.1 View from the outside .....	2
1.1.1 Packaging .....	2
1.1.2 Power requirements .....	4
1.1.3 Cooling requirements .....	4
1.2 Compute and I/O nodes .....	4
1.3 Networking updates .....	5
<b>Chapter 2. Software improvements</b> .....	7
2.1 Location identification .....	8
2.2 Database .....	9
2.3 Reliability, availability, and serviceability infrastructure .....	10
2.4 Blocks .....	10
2.5 Threading .....	11
2.6 Job modes .....	11
2.7 Control system .....	13
2.7.1 Proxy replaced .....	13
2.7.2 CIOD .....	13
2.7.3 Midplane Management Control System .....	14
2.8 Hardware monitor .....	15
2.9 Client/server mpirun .....	16
2.10 Bridge APIs .....	16
2.11 Navigator updates .....	17
2.12 Parts replacement .....	18
2.13 Diagnostics .....	18
<b>Appendix A. Statement of completion</b> .....	19
<b>Related publications</b> .....	21
IBM Redbooks .....	21
How to get IBM Redbooks .....	21
Help from IBM .....	21





# Preface

In this IBM® Redpaper publication, we discuss the evolution of the IBM System Blue Gene® Solution to the new generation IBM System Blue Gene/P™ Solution. This paper is intended for those who are familiar with IBM System Blue Gene/L™ Solution and are interested in the improvements made to the Blue Gene/P Solution.

This paper outlines many of the features that are useful to users, administrators, and facility planners. We present an overview of hardware changes and software improvements at both the user and administrative levels. On the hardware level, we discuss such topics as increased size, power consumption, and airflow. On the software level, we discuss the improvements to the control system and job submission process.

## The team that wrote this paper

This paper was produced by a team of specialists from around the world working at the International Technical Support Organization (ITSO), Rochester Center.

**Gary Lakner** is Staff Software Engineer for IBM Rochester on assignment in the ITSO Rochester Center. He is a member of the IBM System Blue Gene/L Support Team in the IBM Rochester Support Center, where he specializes in both Blue Gene hardware and software, as well as performs customer installations. Prior to joining the Blue Gene team, Gary supported TCP/IP communications on the IBM eServer™ iSeries® server. Gary has been with IBM since 1998.

**Carlos P. Sosa** is a Senior Technical Staff Member in the Blue Gene Development Group of IBM, where he has been the team lead of the Chemistry and Life Sciences high-performance effort since 2006. For the past 18 years, he has focused on scientific applications with emphasis in Life Sciences, parallel programming, benchmarking, and performance tuning. He received a Ph.D. degree in Physical Chemistry from Wayne State University and completed his post-doctoral work at the Pacific Northwest National Laboratory. His areas of interest are future IBM POWER™ architectures, Blue Gene, Cell Broadband, and cellular molecular biology.

Thanks to the following people for their contributions to this project:

Todd Kelsey  
Jenifer Servais  
ITSO, Rochester Center

Linda Robinson  
ITSO, Raleigh Center

Lynn Boger  
Tom Budnik  
Tom Gooding  
Todd Inglett  
Brant L. Knudson  
Cory Lappi  
Tom Liebsch  
Benjamin Mayer  
Mark Megerian

Mike Mundy  
Mike Neslon  
John Orbeck  
Ruth Poole  
Joan Rabe  
Don Reed  
Matt Scheckel  
Karl Solie  
Will Stockdell  
Todd Takken  
Micheal Woiwood  
IBM Rochester

Dino Quintero  
IBM Poughkeepsie

## Become a published author

Join us for a two- to six-week residency program! Help write a book dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You will have the opportunity to team with IBM technical professionals, Business Partners, and Clients.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you will develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

[ibm.com/redbooks/residencies.html](http://ibm.com/redbooks/residencies.html)

## Comments welcome

Your comments are important to us!

We want our papers to be as helpful as possible. Send us your comments about this paper or other IBM Redbooks® in one of the following ways:

- ▶ Use the online **Contact us** review Redbooks form found at:

[ibm.com/redbooks](http://ibm.com/redbooks)

- ▶ Send your comments in an e-mail to:

[redbooks@us.ibm.com](mailto: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



# Hardware changes

The IBM System Blue Gene Solution is a revolutionary and important milestone for IBM in the high-performance computing arena. The Blue Gene/L Solution has maintained its status as the fastest supercomputer for the last few years. Now IBM has introduced the Blue Gene/P Solution as the next-generation of massively-parallel supercomputers, based on the same successful architecture as the Blue Gene/L Solution.

In this chapter, we begin with a brief overview of the Blue Gene hardware and highlight a few of the significant changes that were made to the Blue Gene/L hardware, resulting in the evolution of the Blue Gene/P system.

## 1.1 View from the outside

The Blue Gene/P system has the familiar, slanted profile that was introduced with the Blue Gene/L system. However the increased compute power requires an increase in airflow, resulting in a larger footprint. Each of the air plenums on the Blue Gene/P system are just over ten inches wider than the plenums of the previous model. Additionally, each Blue Gene/P rack is approximately four inches wider. There are two additional Bulk Power Modules mounted in the Bulk Power enclosure on the top of the rack. Rather than a circuit breaker style switch, there is an on/off toggle switch to power on the machine.

### 1.1.1 Packaging

Figure 1-1 illustrates the packaging of the Blue Gene/L system.

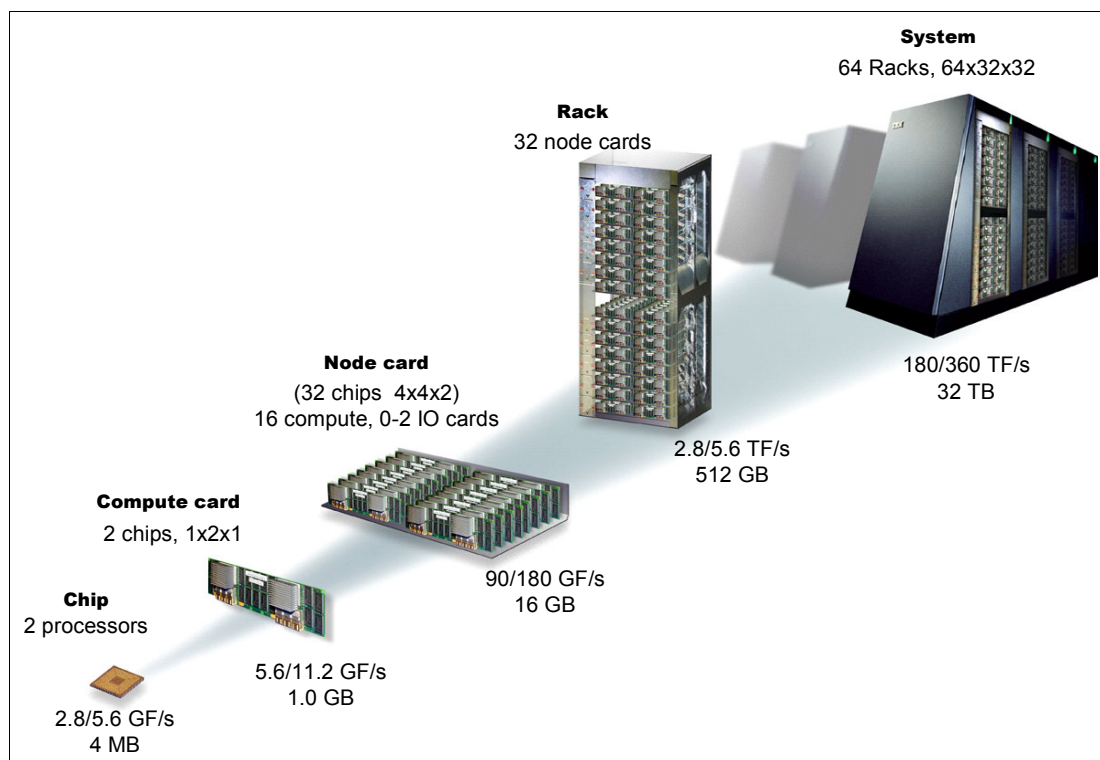


Figure 1-1 Blue Gene/L packaging

Figure 1-2 on page 3 shows how the Blue Gene/P system is packaged. The changes start at the lowest point of the chain. Each chip is made up of four processors rather than just two processors like the Blue Gene/L system supports.

At the next level, only one chip is on each of the compute (processor) cards. This design is easier to maintain with less waste. On the Blue Gene/L system, the replacement of a compute node because of a single failed processor requires the discard of one usable chip because the chips are packaged with two per card. The design of the Blue Gene/P system has only one chip per processor card, eliminating the disposal of a good chip when a compute card is replaced.

Each node card still has 32 chips, but now the maximum number of I/O nodes per node card is two, so that only two Ethernet ports are on the front of each node card. Like the Blue Gene/L system, there are two midplanes per rack. The lower midplane is considered to be the

master. Each midplane drives one service card, four link cards, and sixteen node cards. In addition, twenty fan modules pull cooled air across the cards that plug into the rack's midplanes.

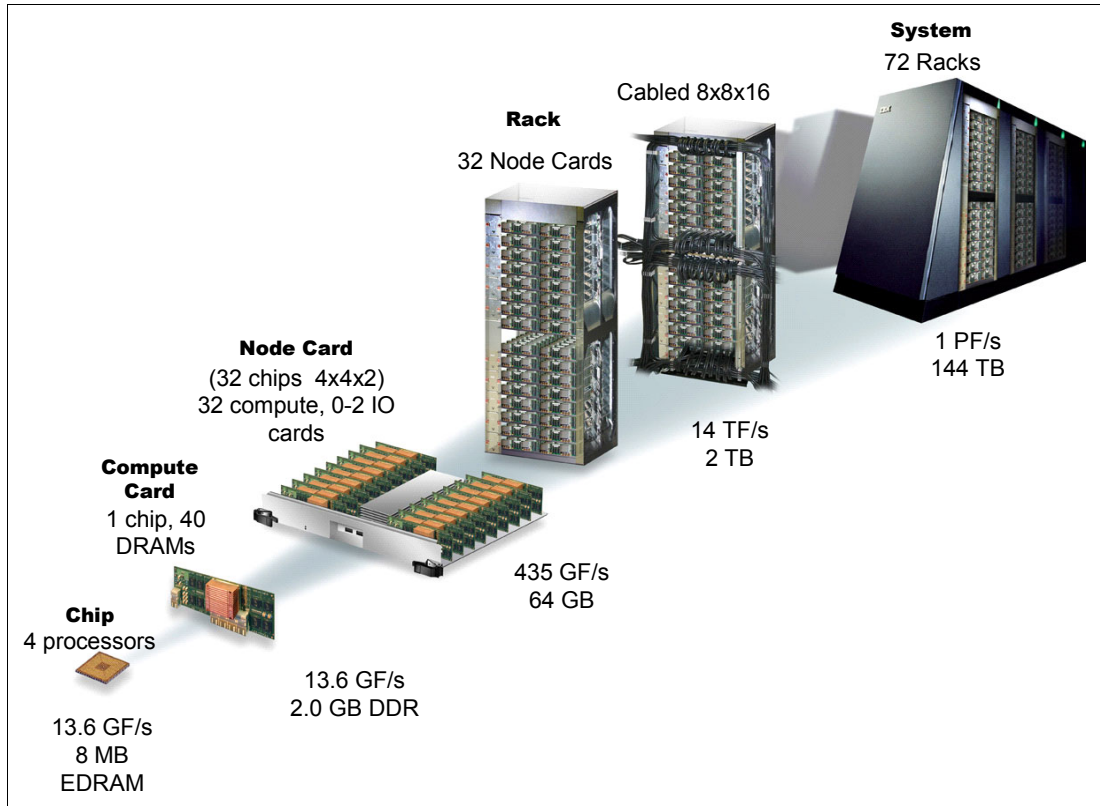


Figure 1-2 Blue Gene/P packaging

Table 1-1 summarizes the packaging between the Blue Gene/L and Blue Gene/P systems.

Table 1-1 Comparison of Blue Gene/L and Blue Gene/P packaging

	Blue Gene/L		Blue Gene/P	
	Quantity per component	To obtain processing speed <sup>a</sup>	Quantity per component	To obtain processing speed <sup>b</sup>
Chip	2 processors	2.8 GF/s 5.6 GF/s	4 processors	13.6 GF/s
Compute card	2 chips	5.6 GF/s 11.2 GF/s	1 chip	13.6 GF/s
Node card	32 chips; 16 per midplane	90 GF/s 180 GF/s	32 chips; 16 per midplane	435 GF/s
Rack	32 node cards	2.8 TF/s 5.6 TF/s	32 node cards	14 TF/s
System	64 racks	180 TF/s 360 TF/s	72 racks	1 PF/s

a. Blue Gene/L runs in coprocessor mode and virtual node mode. The top number reflects the speed in coprocessor mode, and the bottom number reflects the speed in virtual node mode.

b. Blue Gene/P Linpack runs only in virtual node mode.

## 1.1.2 Power requirements

Each Blue Gene/P rack has its own power supply. The power supply can be wired for 480 V 3 phase input and have a 100A line cord with a plug that is connected to a dedicated circuit breaker. Alternatively, the power supply can be wired for 200 V/400 V operation with a 175A line card that is hard wired to an appropriately sized circuit breaker.

On the rack, there are no circuit breakers, but there is an on/off switch at the top of the exhaust plenum. A power supply of 48 V dc is supplied to the rack by using nine 5 kW wide-ranging power supplies that cover both the U.S. 208 V and 480 V 3 phase ac power requirements and the 200 V power requirements used in other parts of the world. The ninth power supply is redundant. Eight power supplies are enough to run the rack. From that point, we use local, point-of-load dc-dc power supplies whenever the local power consumption is of 50 W or more.

## 1.1.3 Cooling requirements

Each Blue Gene/P rack is cooled by using 60 (20 sets of 3) 120 mm fans. These fans operate at a maximum speed of 6800 rpm, pulling air across the midplane. The Blue Gene/L system has the same configuration of fans, but the peak speed is 6000 rpm.

As with later versions of the Blue Gene/L system, the fans can be fine tuned to adjust the amount of airflow within the various areas inside the rack. Also, as we mentioned earlier, the racks are wider to provide better airflow across the cards in the rack.

## 1.2 Compute and I/O nodes

The compute nodes and I/O nodes in the Blue Gene/L system are two unique pieces of hardware. In the Blue Gene/P system, the two parts, although performing different functions, are interchangeable. Table 1-2 shows a comparison of the compute nodes.

Table 1-2 Comparison of the Blue Gene/L and Blue Gene/P nodes

Feature	Blue Gene/L	Blue Gene/P
Cores per node	2	4
Core clock speed	700 MHz	850 MHz
Cache coherency	Software managed	SMP
Private L1 cache	32 KB per core	32 KB per core
Private L2 cache	14 stream prefetching	14 stream prefetching
Shared L3 cache	4 MB	8 MB
Physical memory per node	512 MB - 1 GB	2 GB
Main memory bandwidth	5.6 GB/s	13.6 GB/s
Peak performance	5.6 GFlop/s per node	13.6 GFlop/s per node

Feature	Blue Gene/L	Blue Gene/P
<b>Network topologies</b>		
<b>Torus network</b>		
Bandwidth	2.1 GB/s	5.1 GB/s
Hardware latency (nearest neighbor)	200 ns (32B packet) and 1.6 $\mu$ s (256B packet)	100 ns (32B packet) and 800 ns (256B packet)
<b>Global collective network</b>		
Bandwidth	700 MB/s	1.7 GB/s
Hardware Latency (round trip worst case)	5.0 $\mu$ s	3.0 $\mu$ s
<b>Full system (72 rack comparison)</b>		
Peak performance	410 TFlop/s	~1 PFlop/s
Power	1.7 MW	~2.3 MW

The compute nodes contain four IBM PowerPC® 450 processors with 2 GB of RAM and run a lightweight kernel to execute user-mode applications only. Typically all four cores are used for computation either in dual node mode, virtual node mode, or symmetric multiprocessor mode. Data is moved between the compute and I/O nodes over the global collective network.

The I/O nodes run an embedded Linux® kernel with minimal packages and are required to support a Network File System (NFS) client and Ethernet network connections. The nodes are also required to act as a gateway for the compute nodes in their respective rack to the external world. The I/O nodes present a subset of standard Linux operating interfaces to the user. The 10 Gigabit Ethernet interface of the I/O Nodes is connected to the core Ethernet switch.

### 1.3 Networking updates

The Blue Gene/P system operates with the same five basic networks that are available on the Blue Gene/L system. The three-dimensional Torus, collective, and global barrier networks provide the high performance intra-application communications. The functional network enables the I/O nodes to send data to, and receive data from, the outside world, while the service (or control) network provides a link from the service node directly to the hardware.

To offload message handling overhead from the cores, injection and reception Direct Memory Access (DMA) has been added to the Blue Gene/P system. In addition to reducing the load on the core, network blockages are expected to be prevented if the memory queues are not drained fast enough.

The Blue Gene/L system provides a 1 Gb Functional Ethernet, while the Blue Gene/P system has been upgraded to a 10 Gb Ethernet. The new functional network uses full duplexing and implements IPv4 checksums for both transmit and receive paths. On the Blue Gene/P system, the functional network can consist of up to 32 links per midplane and can transmit up to 300 meters over a 50 mm multimode fiber cable.







## Software improvements

From a software perspective, the overall design of the Blue Gene/P system is relatively unchanged from the Blue Gene/L system. Many of the changes that have been implemented were made to accommodate the new hardware and enhance the existing features.

## 2.1 Location identification

Location strings are used to identify hardware in the Blue Gene rack. The Blue Gene/L system uses multiple methods to locate hardware depending on whether it is referred to by the baremetal code or the software code. For example, in the Blue Gene/L system, you see either R001 or R00-M1, both referring to the same midplane. With the Blue Gene/P system, the references to the various pieces of hardware are all consistent.

Figure 2-1 shows the convention that is used for the Blue Gene/P system. These locations may also be used as regular expressions in MMCS commands such as `write_con`.

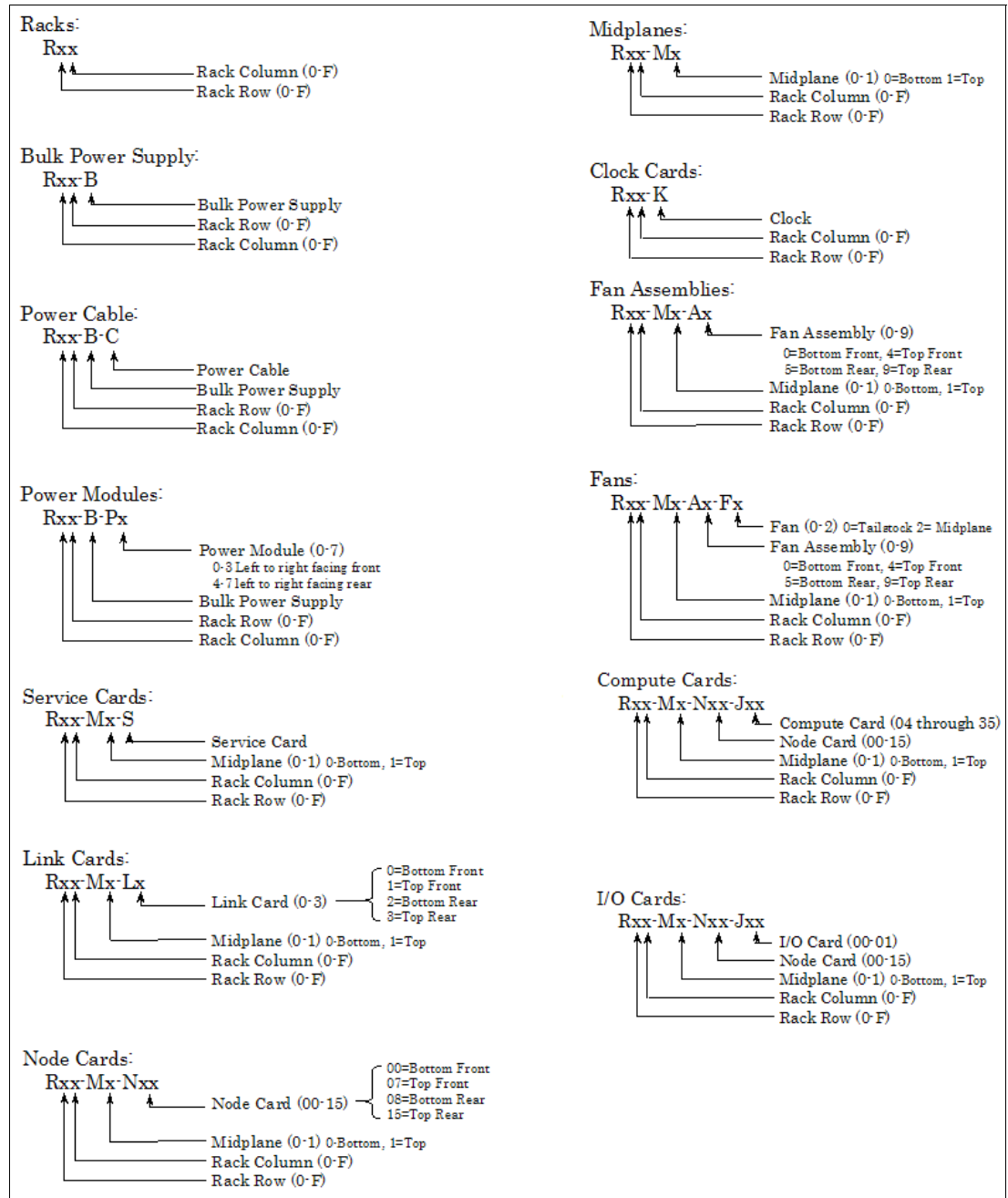


Figure 2-1 Blue Gene/P hardware naming convention

## 2.2 Database

The database has remained largely unchanged in the Blue Gene/P system. Figure 2-2 illustrates the structure of the database on the Blue Gene/P system.

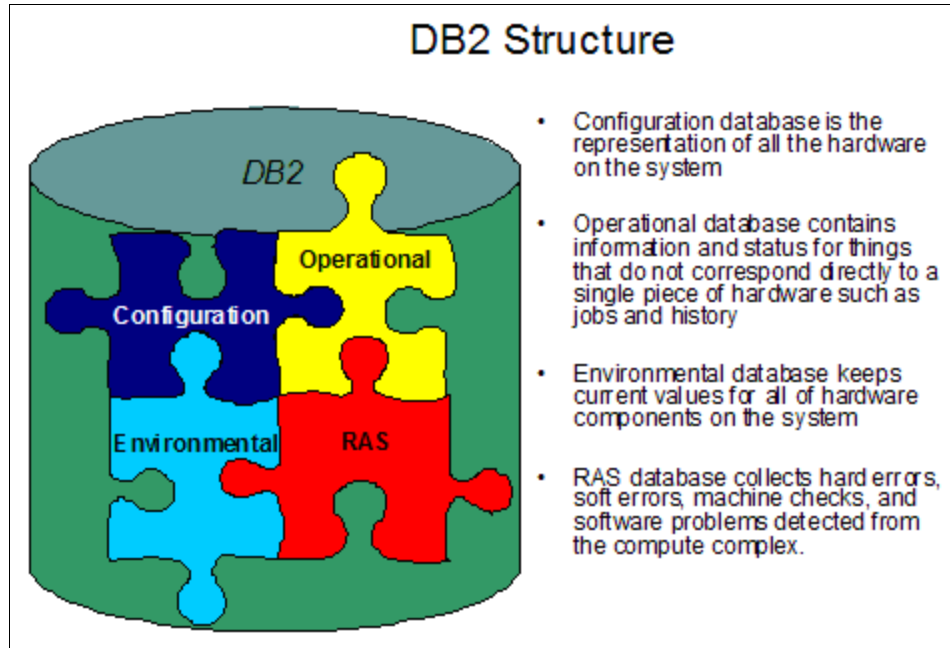


Figure 2-2 Blue Gene/P database structure

One of the more notable improvements in the database schema is the change to the primary key in all of the hardware tables. The primary key that is used in the Blue Gene/L system is the serial number, which is somewhat cumbersome. Now with the Blue Gene/P system, the location string is the primary key in the tables.

With the Blue Gene/L system, it is possible to show more than one piece of hardware for any given location on the system. For example, if a link card is replaced, two link cards are associated with that location, although only one can be active at a time. However, if you perform some action that requires the hardware to be marked as missing in the database, you cannot distinguish which entry in the database is for hardware that is currently installed in the system.

In the Blue Gene/P system, this is no longer a problem. Parts that have been replaced are still maintained in the database, but now they are stored in a history table (BGPREPLACEMENT\_HISTORY table). From our example, the data for the currently installed link card is stored in the BPGLINKCARD table. The data for all of the link cards that have been removed from the system now reside in the BGPLINKCARD\_HISTORY table.

Some changes that were made to the database stemmed from input by customers. One example is the BGPBLOCK table. The notion of processor sets (psets) has been confusing to many. Now, rather than using the number of psets in a block, we report the ratio of I/O nodes to compute nodes that are contained in a given block.

## Database populate

After a Blue Gene/L system hardware installation is completed, a discovery script is run that polls the hardware and writes information about that hardware back to the database.

When a Blue Gene/P system is installed, after the schema is set up, a script called *dbPopulate.pl* is run. The *dbpopulate* script requires you to provide the number of racks in the system and specify how the system is configured. For example, if you have an eight-rack system, you must specify whether it is configured as a one by eight (1 x 8), two by four (2 x 4), or a four by two (4 x 2) configuration. The script then fills in the database with all the possible hardware that can be on the system.

## InstallServiceAction

After the *dbpopulate* script runs, then the *InstallServiceAction* program is started. In short, the *InstallServiceAction* program updates the database with information about the current state of the hardware (active, missing, or in error). Service actions might be necessary to replace hardware in error or that is missing.

## VerifyCables

The Cable Discovery scripts on the Blue Gene/L system go to the system and find each of the cables prior to updating the database. The *dbpopulate* scripts of the Blue Gene/P system adds the cables that should exist on the system to the database. The function of the *VerifyCables* script is similar to the *InstallServiceAction* script. The *VerifyCables* script updates the database with information about the current state of the cables.

## 2.3 Reliability, availability, and serviceability infrastructure

Reliability, availability, and serviceability (RAS) has taken on a completely new format in the Blue Gene/P system. On the Blue Gene/L system, RAS is text based with no predefined format for RAS events. On the Blue Gene/P system, the RAS messages are structured.

A finite list of RAS messages is used on the Blue Gene/P system. All messages have an identification (ID) number and a detailed message. The list of RAS events that have occurred, or may occur, on your system are available for viewing through the Navigator. Additionally, forty user-defined codes are available that can be used to generate RAS messages that will be meaningful in the customer's environment.

## 2.4 Blocks

The Blue Gene/P system offers much more in the way of blocks, or *partitions*, as they are also known. On the Blue Gene/L system, only small blocks with 32 or 128 nodes are supported due to the global interrupt wiring. With the Blue Gene/P system, you can configure any size of small block (with a minimum of 16 compute nodes), as long as one I/O node is contained within the block.

Small block allocation is optimized with the *dynamic allocator*. With the new version of the dynamic allocator, the code looks at a base partition. If a node card is missing, the base partition is marked as unavailable for use with a large partition, but it is made available for small block allocation. If an allocation attempt is made that requires resources equal to, or less than, the number of node cards that are available, the midplane is subdivided further. A minimal amount of fragmentation (orphaning) occurs because of the optimization.

One of the hardware changes that we previously mentioned is that it is only possible to have a maximum of two I/O nodes per node card. This means that the smallest I/O node to compute node ratio available is one to sixteen (1:16). The Blue Gene/L system offers ratios as low as one to eight (1:8) because it is possible to have as many as four I/O nodes per node card.

## 2.5 Threading

The threading implementation on the Blue Gene/P system supports OpenMP. The XL OpenMP implementation provides a futex-compatible syscall interface so that the native POSIX Thread Library (NPTL) pthreads implementation in glibc runs without modification. These syscalls allow only a total of four threads, limited support for mmap, and testing only with usage behavior of OpenMP. The Compute Node Kernel provides a special thread function for I/O handling in Message Passing Interface (MPI).

## 2.6 Job modes

With the Blue Gene/L system, you have the choice of coprocessor mode or virtual node mode. That is to say your program can run on a single core in coprocessor mode. Alternatively, you can split the memory and run your program on both cores in virtual node mode.

The Blue Gene/P system offers three choices in this area. The first option is *symmetric multi-processing (SMP) mode* in which CPU 0 (MPI rank 0) runs the program's main process (Figure 2-3). The program can spawn up to three additional threads on the remaining processors.

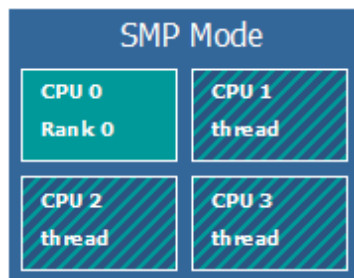


Figure 2-3 SMP mode

The second choice is *dual node mode*. In dual node mode, CPUs 0 and 2 each run a main program process (Figure 2-4). Each of the cores has an MPI rank and can spawn one additional thread. There is a fixed relationship between the cores. That is to say that CPU 0 cannot start a thread on CPUs 1 and 3 and leave only the main process running on CPU 2. CPU 0 is only allowed to send work to CPU 1, and likewise, CPU 2 can only use CPU 3 for additional threads.

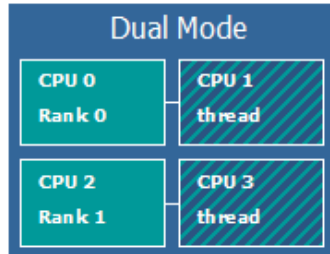


Figure 2-4 Dual node mode

The third choice is *virtual node mode* (Figure 2-5). In this mode, each of the cores in the processor has an MPI rank and runs a program process. There is no additional threading capability in virtual node mode.

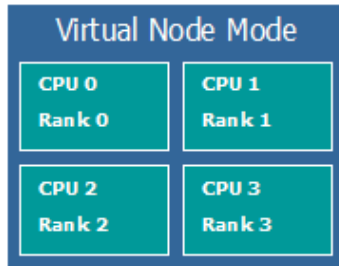


Figure 2-5 Virtual node mode

You can decide which mode your program runs in after the block has been booted. The mode is a *job attribute*, not a block attribute. Dual node mode and virtual node mode require you to split the node's memory evenly between the main processes.

## 2.7 Control system

The Blue Gene/P control system has undergone some streamlining that makes it more efficient. When referring to the control system (illustrated in Figure 2-6), we are generally referring to the following components:

- ▶ Low Level Control System (LLCS)
  - Machine Controller (mc)
  - mcServer
- ▶ High Level Control System, which is also known as the Midplane Management Control System (MMCS)
- ▶ DB2® (RAS interfaces)
- ▶ Scheduler APIs
- ▶ Control and I/O Daemon (CIOD)

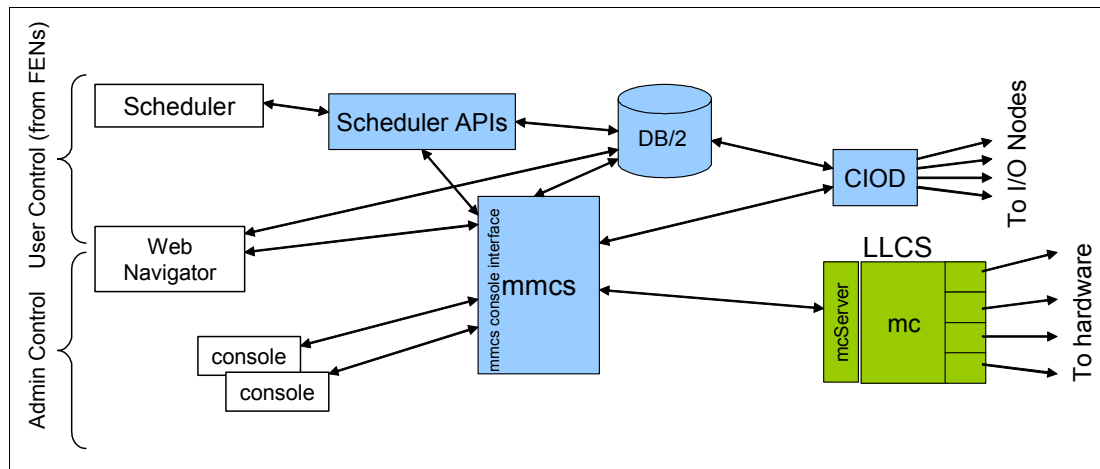


Figure 2-6 Blue Gene/P control system

### 2.7.1 Proxy replaced

mcServer provides the low-level access to the hardware for the Blue Gene/P system. mcServer replaces the IDOproxy functions that are used in the Blue Gene/L system. Much like IDOproxy, mcServer interfaces with the Machine Controller code to limit access to the hardware to one user at a time.

### 2.7.2 CIOD

Job submission requests from the various sources (schedulers, mpirun, and MMCS console) update the job-related database tables. The Blue Gene/L system has a separate process, CIODB, that runs on the service node and handles the job loading and starting. This code polls the job table, starts jobs, monitors jobs, and updates the job table as the job goes through the states of being loaded, started, debugged, terminated, and so on. This code communicates with the I/O node daemon, CIOD, running on each I/O node, to control the job's progress and monitor status.

While this processing is similar on the Blue Gene/P system, there is no longer a separate CIODB process. Since the CIODB code is so closely tied to the MMCS server, the two processes have been merged and run under the MMCS server process.

## 2.7.3 Midplane Management Control System

The Blue Gene/L MMCS binaries are built in 32-bit mode, but later, `mpirun` and the libraries on which it depends were converted to 64-bit mode after address space constraints arose. The Blue Gene/P system has been built with 64-bit mode to avoid the problems that were encountered with the previous version.

On the Blue Gene/L system, you can direct certain commands, such as `write_con`, to a specific node by prefixing the command with the node's index as shown by the `locate` command, such as `{0} write_con ps`. On the Blue Gene/P system, regular expressions might also be used as prefixes. For example, the following command is valid:

```
{R00-M0-N04-J00} write_con ps
```

The `connect` command is new and can be used with a target specification. This command can be used to boot an individual node. Some of the parameters that are on the `allocate_block` command in the Blue Gene/L system have been moved to the `connect` command.

In addition, the new option for MMCS, `--ciod-persist`, allows job control connections to remain open between jobs for the life of the block. It eliminates the need to create a new TCP connection to CIOD every time a job starts.

The `boot_block` command on the Blue Gene/P system has different parameters from the Blue Gene/L system. You can specify the microloader, compute node, and I/O node elf images on '`boot_block`' rather than issuing individual load commands.

Another feature that has been added to the Blue Gene/P system is the ability to perform multiple *connect-boot-disconnect sequences* without reallocating the block. You can disconnect from the block's hardware resources without freeing the block and reconnect without reallocating the block. Therefore, you can boot the block several times without freeing it.

### Reconnecting blocks and jobs

If CIODB ends for some reason while a job is running on a Blue Gene/L system, both the job and the block stay active. However, the ability to monitor and manage the job is lost.

Similarly on the Blue Gene/P system, if the MMCS server stops, the block stays booted and the job keeps running on the nodes. However now with the Blue Gene/P system, you have the option to allow reconnects when the server is restarted. This functionality is enabled by using the `--reconnect` option when starting the MMCS server. After the server is restarted, the job can continue to write to stdout and stderr. It is important to remember that any of the application's output that is sent to stderr/stdout while the server is down is missing.



## 2.8 Hardware monitor

The Blue Gene/L system has a separate process called *Hardware Monitor*. In the Blue Gene/L system, there are concerns about the amount of resources that are required to monitor the hardware. There are also several methods for starting the monitor functions on the Blue Gene/L system. On the Blue Gene/P system, the monitoring functions are integrated with the control system. The Environmental Monitor starts and ends with the MMCS server. The impact on the system as a whole is reduced because there are only three connections to mcServer to collect the data at a time. The monitor comes with the polling interval set to five minutes.

With the original Hardware Monitor, there were two different graphical user interfaces (GUIs) that could be used to view the data collected: Navigator and VNC. The results gathered by the monitor in the Blue Gene/P system can only be accessed by using the Blue Gene Navigator. Figure 2-7 shows the initial page that is displayed when Environmental Queries is selected in the Navigator.

The screenshot shows the Blue Gene Navigator interface. The top bar includes the title 'Blue Gene Navigator', a user greeting 'Welcome, bgpuser', and an 'End session' button. A sidebar on the left lists navigation options under 'Administration' and 'Resources'. The main content area shows system status: 'All Midplanes available', 'All Service cards available', 'All Link cards available', and 'All Node cards available'. A large blue banner reads 'No Attention Required'. Below this, there are tabs for 'Environmental Queries', 'Bulk Power Modules', 'Link Card Power', 'Node Card Power', and 'Service Card Power'. A sub-section for 'Filter Options' includes a table with the following data:

Location	Time	Ref Input Fault	PLL Lock Fault	Frequency
R00-K	6/6/07 11:32:54 AM	False	True	425.000

At the bottom of the filter options, there are navigation buttons: 'First', 'Previous', 'Next', 'Page 1', and a 'Page size' dropdown set to '50' with an 'Apply' button.

Figure 2-7 Environmental Queries

In the Blue Gene/L version of the monitor, there is no limit on the length of time that the collected data will be stored. Monitoring and controlling the amount of data that is kept is entirely up to the administrators of the system. By default, the Blue Gene/P system purges the data every three months. The configuration can be altered to store more or less data as required by the local environment.

## 2.9 Client/server mpirun

`mpirun` on the Blue Gene/P system remains largely unchanged from the Blue Gene/L system. The most notable exception is the removal of the rsh/ssh mechanism for initiating the back-end process. One of the drawbacks to using the rsh or ssh protocols is that they require each user to have a profile on the service node.

On the Blue Gene/P system, this is replaced with a daemon process running on the service node whose purpose is to handle connections from front-end `mpirun` processes and fork back-end (`mpirun_be`) `mpirun` processes. Figure 2-8 illustrates how `mpirun` interacts with the rest of the control system. After `mpirun_be` is forked, the sequence of events for booting partitions, starting jobs, and collecting stdout/stderr is similar to using `mpirun` on the Blue Gene/L system.

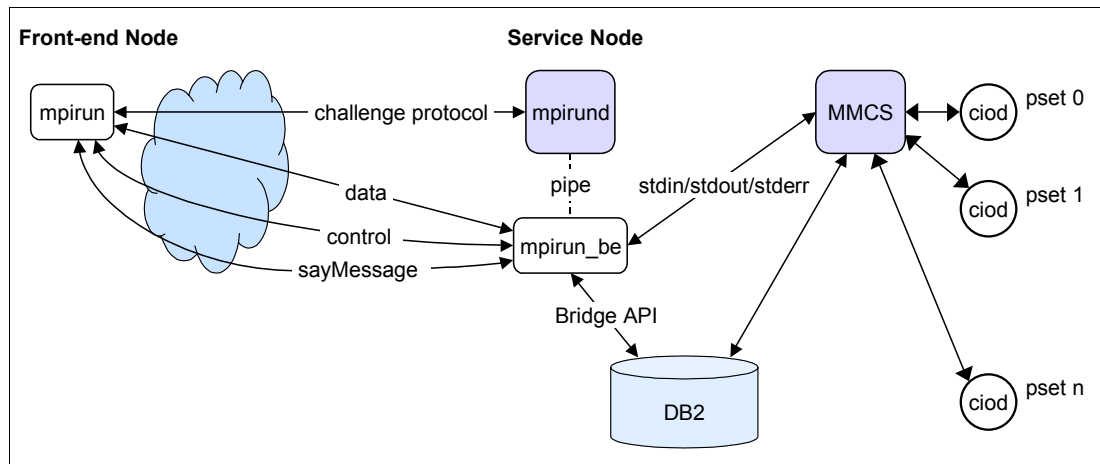


Figure 2-8 `mpirun` flow

Another change in the Blue Gene/P version of `mpirun` is the support for Multiple Program Multiple Data (MPMD) style jobs. With MPMD, a different executable, arguments, environment, and current working directory can be supplied for a single job on a pset basis. For example, a user can run four different executables on a partition that contains four psets. This function is handled by a new tool called `mpiexec`.

**mpiexec:** Be careful not to confuse the `mpiexec` tool with the `mpiexec` style of submitting a Single Program Multiple Data (SPMD) parallel MPI job.

## 2.10 Bridge APIs

In the Blue Gene/L system, the control system stores temporary XML files in the `/tmp` directory, which can occasionally cause problems if the `/tmp` file system become full. The Blue Gene/P system has resolved this problem by no longer writing XML files to `/tmp`. Instead they are passed in memory. This change has improved the overall performance of the control system as well.

Users of the Bridge APIs, for example `mpirun` and LoadLeveler®, should see significant performance improvements because of the implementation of database connection pooling and a reduction in the number of database connections required on some Bridge APIs.

## 2.11 Navigator updates

The original purpose of the Blue Gene Navigator was to provide administrators with a GUI from which they could manage their system. Since that time, the Navigator has evolved into a user tool that shows the health of the system, its utilization, and the availability of system resources. In the following sections, we highlight the additional improvements that were made to the Blue Gene Navigator for the Blue Gene/P system.

### Replacement History

One of the new items that has been added to the Navigator is the Replacement History link. By clicking this link, you can view all of the various hardware types that have been replaced and filter your view based on specific times, location, serial number, or electronic chip ID (ECID).

### Block Builder

The Block Builder feature was introduced in the Blue Gene/L version of the Navigator. By using this feature, users can create blocks that consist of one or more midplanes. In the Blue Gene/L system, the more complicated blocks, such as sub-midplane (small) blocks, must be defined by using XML. The Blue Gene/P system has eliminated the need for users to use any XML. All of the necessary functionality is provided in the Navigator and the bridge APIs, including small blocks and blocks doing passthrough.

### Service Actions

With the Blue Gene/L system, you have the ability to query the Service Actions table to view current or previous actions that were performed on the system. On the Blue Gene/P system, administrators can initiate a Service Action from the Navigator.

### BGPMaster

An interface to manage BGPMaster was added to the Navigator. You have the options to start, restart, or stop mcServer, mmcs\_server, mpirund, navigator\_server, and realtime\_server from the GUI.

Status messages that concern each of the servers is displayed. For example, if a server has been stopped, the message shows the action that is required to recover and how long the server has been down.

### RAS message types

A link was added to the Navigator from which you can search on the types of messages that have occurred on your system. The default query brings up all message types that have occurred in the last day (24 hours), but you can filter on specific message IDs or time periods.

### Plug-ins

The ability to change some of the information displayed by Blue Gene Navigator has been added to the Blue Gene/P system. Now you can write plug-ins to customize your interface. You can add to or replace the existing graphs at the bottom of the page. Additional links can be added to the navigation pane. By default, the Attention area at the top of the page alerts you when certain events occur, such as hardware failures or RAS events. You can customize this area to notify you of events that may have significance in your environment.

## 2.12 Parts replacement

Performing service actions on the Blue Gene/P system does not have the same impact that it does on the Blue Gene/L system. Originally, even changing a compute node required an entire midplane to be taken out of service for a short time.

In the Blue Gene/P system, the service actions are more finite. They affect only the specific hardware that is being serviced. A good example is the same procedure, replacing a compute node. Now this process only affects partitions (and jobs running on them) that include the specific node card that contains that compute node rather than the whole midplane. An even better example is the service action that is performed to replace a link card. On a multiple rack system, this type of service requires that all the racks in the same row, and all the racks in the column be put into service mode. The service actions of the Blue Gene/P system no longer need to power down all of the neighboring link cards. This results in a dramatic difference in the amount of time it takes to prepare for and end the service action.

The Blue Gene Navigator has an interface with which you can perform service actions and see which jobs and blocks will be impacted by the service action that you are going to perform.

## 2.13 Diagnostics

Diagnostics on the Blue Gene/P system are similar to the set of tests that are available on the Blue Gene/L system with the addition of test cases added to exercise new features such as Direct Memory Access (DMA).

Diagnostics are still initiated from Blue Gene Navigator. You can use them to track the progress of, and cancel, a diagnostic run if necessary. Results are stored in the database and viewable from the Navigator. There are cross-reference links between RAS messages, failures, and hardware locations.

You can select between small, medium, or large sets of tests plus an additional test bucket is added that is labelled *complete*. The complete option includes all the tests in the large suite, plus a single node Linpack and a torus connectivity test.

Diagnostics are more efficient in the Blue Gene/P system because pipelining has been introduced into the harness. The harness runs tests on the hardware while compiling results from previously run tests.

RAS and error reporting are improved for both diagnostics and normal system usage. For instance, RAS events are now used to indicate failures as opposed to error messages being sprinkled throughout test output. This event-driven model provides many benefits, which include:

- ▶ Reduced time spent parsing code.
- ▶ Improved diagnostics runtime.
- ▶ Decreased network utilization.

The new RAS subsystem provides not only low-level error reporting, but decoding facilities in the control system to enhance error messages on the way back up to the user. The diagnostics can take an overview of all the RAS events that are posted during a test. They can also make decisions based on all the information presented.



**A**

## **Statement of completion**

IBM considers installation to be complete when the following activities have taken place:

- ▶ The Blue Gene/P rack or racks have been physically placed in position.
- ▶ The cabling is complete, including power, ethernet, and torus cables.
- ▶ The Blue Gene/P racks can be powered on.
- ▶ All hardware is displayed in the Blue Gene Navigator and is available.



# Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this Redpaper.

## IBM Redbooks

For information about ordering these publications, see “How to get IBM Redbooks” on page 21. Note that some of the documents referenced here may be available in softcopy only.

- ▶ *Blue Gene Safety Considerations*, REDP-4257
- ▶ *Blue Gene System Administration*, SG24-7417
- ▶ *Blue Gene/L: Hardware Overview and Planning*, SG24-6796
- ▶ *Blue Gene/L: Performance Analysis Tools*, SG24-7278
- ▶ *Blue Gene/L: Safety Considerations*, REDP-3983
- ▶ *IBM System Blue Gene Solution: Application Development*, SG24-7179
- ▶ *IBM System Blue Gene Solution: Blue Gene/P Application Development*, SG24-7287
- ▶ *IBM System Blue Gene Solution: High Performance Computing Toolkit for Blue Gene/P*, REDP-4256
- ▶ *IBM System Blue Gene Solution: System Administration*, SG24-7178
- ▶ *Unfolding the IBM eServer Blue Gene Solution*, SG24-6686

## How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

[ibm.com/redbooks](http://ibm.com/redbooks)

## Help from IBM

IBM Support and downloads

[ibm.com/support](http://ibm.com/support)

IBM Global Services

[ibm.com/services](http://ibm.com/services)









# Evolution of the IBM System Blue Gene Solution



**Redpaper**

## **A new generation of hardware**

## **Additional software functionality**

## **Enhanced control system software**

In this IBM Redpaper publication, we discuss the evolution of the IBM System Blue Gene Solution to the new generation IBM System Blue Gene/P Solution. This paper is intended for those who are familiar with IBM System Blue Gene/L Solution and are interested in the improvements made to the Blue Gene/P Solution.

This paper outlines many of the features that are useful to users, administrators, and facility planners. We present an overview of hardware changes and software improvements at both the user and administrative levels. On the hardware level, we discuss such topics as increased size, power consumption, and airflow. On the software level, we discuss the improvements to the control system and job submission process.

## **INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION**

## **BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE**

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

**For more information:**  
[ibm.com/redbooks](http://ibm.com/redbooks)