

1. Record Nr.	UNINA9910825288003321
Autore	Mullen-Schultz Gary L
Titolo	Blue Gene/L : application development / / Gary L. Mullen-Schultz
Pubbl/distr/stampa	Rochester, MN, : IBM, International Technical Support Organization, c2005
Descrizione fisica	1 online resource (180 p.)
Collana	Redbooks
Soggetti	Supercomputers IBM computers
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	"December 2005."
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	Front cover -- Contents -- Notices -- Trademarks -- Preface -- The team that wrote this redbook -- Become a published author -- Comments welcome -- Summary of changes -- December 2005, Second Edition -- Part 1 MPI application information -- Chapter 1. Application development overview -- 1.1 MPI on Blue Gene/L -- 1.2 Memory considerations -- 1.2.1 Memory leaks -- 1.2.2 Memory management -- 1.2.3 Uninitialized pointers -- 1.2.4 Forcing MPI to allocate too much memory -- 1.2.5 Not waiting for MPI_Test -- 1.2.6 Flooding of messages -- 1.2.7 Poor choice of programming mode -- 1.3 Other considerations -- 1.3.1 Input/output -- 1.3.2 Miscellaneous -- 1.4 Include and link files -- 1.4.1 Include files -- 1.4.2 Static link files -- 1.5 Compilers overview -- 1.5.1 Programming environment overview -- 1.5.2 GNU -- 1.5.3 IBM XL compilers -- Chapter 2. Programming modes -- 2.1 Communication Coprocessor Mode -- 2.2 Virtual Node Mode -- 2.3 Which mode to use -- 2.4 Choosing modes -- Chapter 3. System calls supported by Compute Node Kernel -- 3.1 Introduction to the Compute Node Kernel -- 3.2 System calls -- 3.2.1 Return codes -- 3.2.2 List of supported system calls -- 3.3 Unsupported calls -- Chapter 4. Developing applications with IBM XL compilers -- 4.1 Compiling and linking applications on Blue Gene/L -- 4.2 Default compiler options -- 4.3 Unsupported options -- 4.4 Tuning your code for Blue Gene/L -- 4.5 Using the compiler optimization options -- 4.6 Structuring data in adjacent pairs -- 4.7 Using

vectorizable basic blocks -- 4.8 Using inline functions -- 4.9 Removing possibilities for aliasing (C/C++) -- 4.10 Structure computations in batches of five or ten -- 4.11 Checking for data alignment -- 4.12 Using XL built-in floating-point functions for Blue Gene/L -- 4.13 Complex type manipulation functions -- 4.14 Load and store functions -- 4.15 Move functions.

4.16 Arithmetic functions -- 4.16.1 Unary functions -- 4.16.2 Binary functions -- 4.16.3 Multiply-add functions -- 4.17 Select functions -- 4.18 Examples of built-in functions usage -- Chapter 5. Running and debugging -- 5.1 Running applications -- 5.1.1 mmcs_db_console -- 5.1.2 mpirun -- 5.1.3 LoadLeveler -- 5.1.4 Other scheduler products -- 5.2 Debugging applications -- 5.2.1 General debugging architecture -- 5.2.2 GDB -- 5.2.3 TotalView -- Chapter 6. Checkpoint and restart support -- 6.1 Why use checkpoint and restart -- 6.2 Technical overview -- 6.2.1 Input/output considerations -- 6.2.2 Signal considerations -- 6.3 Checkpoint API -- 6.3.1 Checkpoint library API -- 6.4 Directory and file naming conventions -- 6.5 Restart -- 6.5.1 Determining latest consistent global checkpoint -- 6.5.2 Checkpoint and restart functionality -- Part 2 System application information -- Chapter 7. Control system (Bridge) APIs -- 7.1 API support overview -- 7.1.1 Requirements -- 7.1.2 General comments -- 7.1.3 Memory allocation and deallocation -- 7.2 APIs -- 7.2.1 API to the MMCS Resource Manager -- 7.2.2 Resource Manager Memory Allocators API -- 7.2.3 Resource Manager Memory Deallocators API -- 7.2.4 Messaging API -- 7.2.5 API to the MMCS job manager -- 7.2.6 API to the MMCS partition manager -- 7.2.7 State diagrams for jobs and partitions -- 7.3 Control system API return codes -- 7.3.1 Return codes specification -- 7.4 Small partition allocation -- 7.4.1 BP definitions -- 7.4.2 API examples -- 7.4.3 Allocating a new small partition -- 7.4.4 Querying a small partition -- Part 3 Performance analysis -- Chapter 8. Performance guidelines and tools -- 8.1 Tooling overview -- 8.1.1 IBM High Performance Computing Toolkit -- 8.2 General performance testing -- 8.2.1 Overview of the tools that are available on pSeries -- 8.2.2 Overview of tools ported to Blue Gene/L.

8.3 Message passing performance -- 8.3.1 MPI Tracer and Profiler -- 8.4 CPU performance -- 8.4.1 Hardware performance monitor -- 8.4.2 Xprofiler -- 8.5 I/O performance -- 8.5.1 Modular I/O -- 8.6 Visualization and analysis -- 8.6.1 PeekPerf -- 8.7 MASS and MASSV libraries -- Chapter 9. Performance counters and PAPI -- 9.1 Introduction to the performance counter interface -- 9.2 bgl_perfctr library API -- 9.2.1 API details -- 9.2.2 Ways to access the counters -- 9.2.3 Available counter events -- 9.2.4 Correct API usage -- 9.3 PAPI implementation -- 9.3.1 linux-bgl PAPI substrate -- 9.3.2 PAPI event mapping for Blue Gene/L -- 9.3.3 Modifications to PAPI -- 9.4 Examples of using HPM libraries for Blue Gene/L -- 9.4.1 PAPI library usage examples -- 9.4.2 bgl_perfctr usage example -- 9.5 Conclusion -- Appendix A. Statement of completion -- Appendix B. Electromagnetic compatibility -- Appendix C. Blue Gene/L safety considerations -- Important safety notices -- Stability and weight -- Circuit breakers -- Ac terminal blocks -- Line cord retention -- Bulk power module bay -- Cover access -- Fan assembly/cards -- Appendix D. MPI environment variables -- Setting environment variables -- BGLMPI_COLLECTIVE_DISABLE -- BGLMPI_EAGER, BGLMPI_RVZ, and BGLMPI_RZV -- Glossary -- Related publications -- IBM Redbooks -- Other publications -- Online resources -- How to get IBM Redbooks -- Help from IBM -- Index -- Back cover.
