1. Record Nr. UNINA9910293150303321

Autore	Supalov Alexander
Titolo	Optimizing HPC Applications with Intel Cluster Tools [[electronic resource]] : Hunting Petaflops / / by Alexander Supalov, Andrey Semin, Christopher Dahnken, Michael Klemm
Pubbl/distr/stampa	Springer Nature, 2014 Berkeley, CA : , : Apress : , : Imprint : Apress, , 2014
ISBN	1-4302-6497-7
Edizione	[1st ed. 2014.]
Descrizione fisica	1 online resource (291 pages) : illustrations
Collana	The expert's voice in software engineering
Disciplina	004.11
Soggetti	Programming languages (Electronic computers) Software engineering Programming Languages, Compilers, Interpreters Software Engineering/Programming and Operating Systems
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Bibliographic Level Mode of Issuance: Monograph
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	Intro -- Contents at a Glance -- Contents -- About the Authors -- About the Technical Reviewers -- Acknowledgments -- Foreword -- Introduction -- Chapter 1: No Time to Read This Book? -- Using Intel MPI Library -- Using Intel Composer XE -- Tuning Intel MPI Library -- Gather Built-in Statistics -- Optimize Process Placement -- Optimize Thread Placement -- Tuning Intel Composer XE -- Analyze Optimization and Vectorization Reports -- Use Interprocedural Optimization -- Summary -- References -- Chapter 2: Overview of Platform Architectures -- Performance Metrics and Targets -- Latency, Throughput, Energy, and Power -- Peak Performance as the Ultimate Limit -- Scalability and Maximum Parallel Speedup -- Bottlenecks and a Bit of Queuing Theory -- Roofline Model -- Performance Features of Computer Architectures -- Increasing Single-Threaded Performance: Where You Can and Cannot Help -- Process More Data with SIMD Parallelism -- Distributed and Shared Memory Systems -- Use More Independent Threads on the Same Node -- Don't Limit Yourself to a Single Server -- HPC Hardware Architecture Overview -- A Multicore Workstation or a Server Compute Node -- Coprocessor for Highly

Parallel Applications -- Group of Similar Nodes Form an HPC Cluster -- Other Important Components of HPC Systems -- Summary -- References -- Chapter 3: Top-Down Software Optimization -- The Three Levels and Their Impact on Performance -- System Level -- Application Level -- Working Against the Memory Wall -- The Magic of Vectors -- Distributed Memory Parallelization -- Shared Memory Parallelization -- Other Existing Approaches and Methods -- Microarchitecture Level -- Addressing Pipelines and Execution -- Closed-Loop Methodology -- Workload, Application, and Baseline -- Iterating the Optimization Process -- Summary -- References -- Chapter 4: Addressing System Bottlenecks.
Classifying System-Level Bottlenecks -- Identifying Issues Related to System Condition -- Characterizing Problems Caused by System Configuration -- Understanding System-Level Performance Limits -- Checking General Compute Subsystem Performance -- Testing Memory Subsystem Performance -- Testing I/O Subsystem Performance -- Characterizing Application System-Level Issues -- Selecting Performance Characterization Tools -- Monitoring the I/O Utilization -- Analyzing Memory Bandwidth -- Summary -- References -- Chapter 5: Addressing Application Bottlenecks: Distributed Memory -- Algorithm for Optimizing MPI Performance -- Comprehending the Underlying MPI Performance -- Recalling Some Benchmarking Basics -- Gauging Default Intranode Communication Performance -- Gauging Default Internode Communication Performance -- Discovering Default Process Layout and Pinning Details -- Gauging Physical Core Performance -- Doing Initial Performance Analysis -- Is It Worth the Trouble? -- Example 1: Initial HPL Performance Investigation -- Getting an Overview of Scalability and Performance -- Learning Application Behavior -- Example 2: MiniFE Performance Investigation -- Choosing Representative Workload(s) -- Example 2 (cont.): MiniFE Performance Investigation -- Balancing Process and Thread Parallelism -- Example 2 (cont.): MiniFE Performance Investigation -- Doing a Scalability Review -- Example 2 (cont.): MiniFE Performance Investigation -- Analyzing the Details of the Application Behavior -- Example 2 (cont.): MiniFE Performance Investigation -- Choosing the Optimization Objective -- Detecting Load Imbalance -- Example 2 (cont.): MiniFE Performance Investigation -- Dealing with Load Imbalance -- Classifying Load Imbalance -- Addressing Load Imbalance -- Example 2 (cont.): MiniFE Performance Investigation -- Example 3: MiniMD Performance Investigation.
Optimizing MPI Performance -- Classifying the MPI Performance Issues -- Addressing MPI Performance Issues -- Mapping Application onto the Platform -- Understanding Communication Paths -- Selecting Proper Communication Fabrics -- Using Scalable Datagrams -- Specifying a Network Provider -- Using IP over IB -- Controlling the Fabric Fallback Mechanism -- Using Multirail Capabilities -- Detecting and Classifying Improper Process Layout and Pinning Issues -- Controlling Process Layout -- Controlling the Global Process Layout -- Controlling the Detailed Process Layout -- Setting the Environment Variables at All Levels -- Controlling the Process Pinning -- Controlling Memory and Network Affinity -- Example 4: MiniMD Performance Investigation on Xeon Phi -- Example 5: MiniGhost Performance Investigation -- Tuning the Intel MPI Library -- Tuning Intel MPI for the Platform -- Tuning Point-to-Point Settings -- Adjusting the Eager and Rendezvous Protocol Thresholds -- Changing DAPL and DAPL UD Eager Protocol Threshold -- Bypassing Shared Memory for Intranode Communication -- Bypassing the Cache for Intranode Communication -- Choosing the Best Collective Algorithms -- Tuning Intel MPI Library for the

Application -- Using Magical Tips and Tricks -- Disabling the Dynamic Connection Mode -- Applying the Wait Mode to Oversubscribed Jobs -- Fine-Tuning the Message-Passing Progress Engine -- Reducing the Pre-reserved DAPL Memory Size -- What Else? -- Example 5 (cont.): MiniGhost Performance Investigation -- Optimizing Application for Intel MPI -- Avoiding MPI_ANY_SOURCE -- Avoiding Superfluous Synchronization -- Using Derived Datatypes -- Using Collective Operations -- Betting on the Computation/Communication Overlap -- Replacing Blocking Collective Operations by MPI-3 Nonblocking Ones -- Using Accelerated MPI File I/O.
Example 5 (cont.): MiniGhost Performance Investigation -- Using Advanced Analysis Techniques -- Automatically Checking MPI Program Correctness -- Comparing Application Traces -- Instrumenting Application Code -- Correlating MPI and Hardware Events -- Collecting and Analyzing Hardware Counter Information in ITAC -- Collecting and Analyzing Hardware Counter Information in VTune -- Summary -- References -- Chapter 6: Addressing Application Bottlenecks: Shared Memory -- Profiling Your Application -- Using VTune Amplifier XE for Hotspots Profiling -- Hotspots for the HPCG Benchmark -- Compiler-Assisted Loop/Function Profiling -- Sequential Code and Detecting Load Imbalances -- Thread Synchronization and Locking -- Dealing with Memory Locality and NUMA Effects -- Thread and Process Pinning -- Controlling OpenMP Thread Placement -- Thread Placement in Hybrid Applications -- Summary -- References -- Chapter 7: Addressing Application Bottlenecks: Microarchitecture -- Overview of a Modern Processor Pipeline -- Pipelined Execution -- Data Conflicts -- Control Conflicts -- Structural Conflicts -- Out-of-order vs. In-order Execution -- Superscalar Pipelines -- SIMD Execution -- Speculative Execution: Branch Prediction -- Memory Subsystem -- Putting It All Together: A Final Look at the Sandy Bridge Pipeline -- A Top-down Method for Categorizing the Pipeline Performance -- Intel Composer XE Usage for Microarchitecture Optimizations -- Basic Compiler Usage and Optimization -- Using Optimization and Vectorization Reports to Read the Compiler's Mind -- Optimizing for Vectorization -- The AVX Instruction Set -- Why Doesn't My Code Vectorize in the First Place? -- Data Dependences -- Data Aliasing -- Array Notations -- Vectorization Directives -- ivdep -- vector -- simd -- Understanding AVX: Intrinsic Programming -- What Are Intrinsics?.
First Steps: Loading and Storing -- Arithmetic -- Data Rearrangement -- Dealing with Disambiguation -- Dealing with Branches -- __builtin_expect -- Profile-Guided Optimization -- Pragmas for Unrolling Loops and Inlining -- unroll/nounroll -- unroll_and_jam/nounroll_and_jam -- inline, noinline, forceinline -- Specialized Routines: How to Exploit the Branch Prediction for Maximal Performance -- When Optimization Leads to Wrong Results -- Using a Standard Library Method -- Using a Manual Implementation in C -- Vectorization with Directives -- Analyzing Pipeline Performance with Intel VTune Amplifier XE -- Summary -- References -- Chapter 8: Application Design Considerations -- Abstraction and Generalization of the Platform Architecture -- Types of Abstractions -- Levels of Abstraction and Complexities -- Raw Hardware vs. Virtualized Hardware in the Cloud -- Questions about Application Design -- Designing for Performance and Scaling -- Designing for Flexibility and Performance Portability -- Data Layout -- Structured Approach to Express Parallelism -- Understanding Bounds and Projecting Bottlenecks -- Data Storage or Transfer vs. Recalculation -- Total Productivity Assessment -- Summary -- References -- Index.

Sommario/riassunto

Optimizing HPC Applications with Intel® Cluster Tools takes the reader

on a tour of the fast-growing area of high performance computing and the optimization of hybrid programs. These programs typically combine distributed memory and shared memory programming models and use the Message Passing Interface (MPI) and OpenMP for multi-threading to achieve the ultimate goal of high performance at low power consumption on enterprise-class workstations and compute clusters. The book focuses on optimization for clusters consisting of the Intel® Xeon processor, but the optimization methodologies also apply to the Intel® Xeon Phi™ coprocessor and heterogeneous clusters mixing both architectures. Besides the tutorial and reference content, the authors address and refute many myths and misconceptions surrounding the topic. The text is augmented and enriched by descriptions of real-life situations.