Overview of MPI Macrobenchmarks

Jiri Simsa

July 9, 2013

1 ASC Sequoia Benchmark Codes [4]

The ASC Sequoia benchmark codes are used by the Lawrence Livermore National Lab to test the performance and functionality of the Sequoia supercomputer.

1.1 AMG

Summary: AMG is a parallel algebraic multigrid solver for linear systems arising from problems on unstructured grids. The driver provided for this benchmark builds linear systems for various 3D problems.

Language: C

Parallelism: MPI, OpenMP

MPI APIs: Blocking Collectives, Blocking P2P, Communicator Management, Datatypes, Non-blocking P2P, Persistent P2P

1.2 IRS

Summary: The implicit radiation solver code (IRS) solves the radiation transport equation by the flux-limited diffusion approximation using an implicit matrix solution. In fact, IRS is a general diffusion equation solver, but its flux limiter imposes the speed of light as the maximum signal speed, hence making it a radiation solver.

Language: C

Parallelism: MPI, OpenMP

MPI APIs: Blocking Collectives, Blocking P2P, Communicator Management, Non-blocking P2P

1.3 LAMMPS

Summary: LAMMPS is a well-known open source code that is capable of simulating a wide variety of particle-like systems. For the purposes of the Sequoia procurement, LAMMPS is used to run a simple classical molecular dynamics problem. Parallelization is by spatial discretization and the vast bulk of communication is nearest neighbor. This makes LAMMPS scale well, and, more importantly, it makes it possible to predict with confidence the parallel efficiency for large systems.

Language: C++

Parallelism: MPI

MPI APIs: Blocking Collectives, Blocking P2P, Communicator Management, Non-blocking P2P

1.4 SPHOT

Summary: SPhot is a 2D photon transport code. Photons are born in hot matter and tracked through a spherical domain that is cylindrically symmetric on a logically rectilinear, 2D mesh.

Language: Fortran

Parallelism: MPI, OpenMP

MPI APIs: Blocking Collectives, Blocking P2P, Datatypes, Non-blocking P2P

1.5 UMT

Summary: The UMT benchmark is a 3D, deterministic, multigroup, photon transport code for unstructured meshes. The transport code solves the rst-order form of the steady-state Boltzmann transport equation. The equation's energy dependence is modeled using multiple photon energy groups. The angular dependence is modeled using a collocation of discrete directions, or "ordinates." The spatial variable is modeled with an "upstream corner balance" nite volume dierencing technique. The solution proceeds by tracking through the mesh in the direction of each ordinate. For each ordinate direction all energy groups are transported, accumulating the desired solution on each zone in the mesh.

Parallelism: MPI, OpenMP

MPI APIs: Blocking Collectives, Communicator Management, Non-blocking P2P, Persistent P2P

2 The Mantevo Project [3]

The Mantevo project is an effort to provide open-source software packages for the analysis, prediction and improvement of high performance computing applications.

2.1 CloverLeaf

Summary: CloverLeaf is a miniapp that solves the compressible Euler equations on a Cartesian grid, using an explicit, second-order accurate method.

Language: C, Fortran

Parallelism: MPI, OpenMP

MPI APIs: Blocking Collectives, Non-blocking P2P

2.2 CoMD

Summary: A simple proxy for the computations in a typical molecular dynamics application. The reference implementation mimics that of SPaSM. In addition, an OpenCL implementation which allows testing on multicore and GPU architectures, is provided.

Language: C

Parallelism: MPI

MPI APIs: Blocking Collectives, Blocking P2P

2.3 HPCCG

Summary: HPCCG is intended to be the best approximation to an unstructured implicit finite element or finite volume application in 800 lines or fewer.

Language: C++

Parallelism: MPI, OpenMP

MPI APIs: Blocking Collectives, Blocking P2P, Communicator Management, Non-Blocking P2P

2.4 MiniFE

Summary: MiniFE is an proxy application for unstructured implicit finite element codes. It is similar to HPCCG and pHPCCG but provides a much more complete vertical covering of the steps in this class of applications. MiniFE also provides support for computation on multicore nodes, including pthreads and Intel Threading Building Blocks (TBB) for homogeneous multicore and CUDA for GPUs.

Language: C++

Parallelism: MPI

MPI APIs: Blocking Collectives, Blocking P2P, Non-Blocking P2P

2.5 MiniGhost

Summary: A finite difference proxy application which implements a difference stencil across a homogenous three dimensional domain.

Language: Fortran

Parallelism: MPI, OpenMP

MPI APIs: Blocking Collectives, Communicator Management, Datatypes, File Manipulation, Non-Blocking P2P

2.6 MiniMD

Summary: A simple proxy for the force computations in a typical molecular dynamics applications. The algorithms and implementation used closely mimics these same operations as performed in LAMMPS.

Language: C++

Parallelism: MPI, OpenMP

MPI APIs: Blocking Collectives, Blocking P2P, Communicator Management, Non-Blocking P2P

2.7 MiniXyce

Summary: A portable proxy of some of the key capabilities in the electrical modeling Xyce.

Language: C++

Parallelism: MPI

MPI APIs: Blocking Collectives, Blocking P2P

3 NAS Parallel Benchmark [2]

The NAS Parallel Benchmarks (NPB) are a small set of programs designed to help evaluate the performance of parallel supercomputers. The benchmarks are derived from computational fluid dynamics (CFD) applications and consist of five kernels and three pseudo-applications in the original "pencil-and-paper" specification (NPB 1). The benchmark suite has been extended to include new benchmarks for unstructured adaptive mesh, parallel I/O, multi-zone applications, and computational grids. Problem sizes in NPB are predefined and indicated as different classes. Reference implementations of NPB are available in commonly-used programming models like MPI and OpenMP (NPB 2 and NPB 3).

3.1 Integer Sort (IS)

Language: C

Parallelism: MPI

MPI APIs: Blocking Collectives, Blocking P2P, Non-blocking P2P

3.2 Embarassingly Parallel (EP)

Language: Fortran

Parallelism: MPI

MPI APIs: Blocking Collectives

3.3 Conjugate Gradient (CG)

Language: Fortran

Parallelism: MPI

MPI APIs: Blocking Collectives, Blocking P2P, Non-blocking P2P

3.4 Multi-Grid (MG)

Language: Fortran

Parallelism: MPI

MPI APIs: Blocking Collectives, Blocking P2P, Non-blocking P2P

3.5 Fourier Transform (FT)

Language: Fortran

Parallelism: MPI

MPI APIs: Blocking Collectives, Communicator Management

3.6 Block Tri-diagonal Solver (BT)

Language: Fortran

Parallelism: MPI

MPI APIs: Blocking Collectives, Communicator Management, Datatypes, File Manipulation, Non-blocking P2P

3.7 Scalar Penta-diagonal Solver (SP)

Language: Fortran

Parallelism: MPI

MPI APIs: Blocking Collectives, Communicator Management, Non-blocking P2P

3.8 Lower-Upper Gauess-Siedel Solver (LU)

Language: Fortran

Parallelism: MPI

MPI APIs: Blocking Collectives, Blocking P2P, Communicator Management, Non-blocking P2P

4 Miscellaneous

4.1 ARMCI-MPI [1]

Summary: The purpose of the Aggregate Remote Memory Copy (ARMCI) library is to provide a general-purpose, efficient, and widely portable remote memory access (RMA) operations (one-sided communication) optimized for contiguous and noncontiguous (strided, scatter/gather, I/O vector) data transfers. In addition, ARMCI includes a set of atomic and mutual exclusion operations.

Language: C

Parallelism: MPI

MPI APIs: Blocking Collectives, Blocking P2P, Communicator Management, Datatypes, One-sided Communication

4.2 CTF [7]

Summary: Cyclops (cyclic-operations) Tensor Framework (CTF) is a distributed library for tensor contractions. CTF aims to scale high-dimensional tensor contractions such as those required in the Coupled Cluster (CC) electronic structure method to massively-parallel supercomputers.

Language: C++

Parallelism: MPI, OpenMP

MPI API: Blocking Collectives, Blocking P2P, Communicator Management, Non-blocking P2P

4.3 Elemental [5]

Summary: Elemental is a framework for distributed-memory dense linear algebra that strives to be both fast and convenient. It combines ideas including: element-wise matrix distributions, object-oriented submatrix tracking, and first-class matrix distributions.

Language: C++

Parallelism: MPI, OpenMP

MPI API: Blocking Collectives, Blocking P2P, Communicator Management, Non-blocking Collectives, Non-blocking P2P

4.4 Graph500 Search Benchmark [6]

Summary: The intent of this benchmark problem is to develop a compact application that has multiple analysis techniques (multiple kernels) accessing a single data structure representing a weighted, undirected graph. In addition to a kernel to construct the graph from the input tuple list, there is one additional computational kernel to operate on the graph.

Language: C

Parallelism: MPI, OpenMP

MPI API: Blocking Collectives, Communicator Management, File Manipulation, Non-blocking P2P, One-sided Communication

4.5 MADNESS [8]

Summary: MADNESS provides a high-level environment for the solution of integral and differential equations in many dimensions using adaptive, fast methods with guaranteed precision based on multi-resolution analysis and novel separated representations. There are three main components to MADNESS. At the lowest level is a new petascale parallel programming environment that increases programmer productivity and code performance/scalability while maintaining backward compatibility with current programming tools such as MPI and Global Arrays. The numerical capabilities built upon the parallel tools provide a high-level environment for composing and solving numerical problems in many (1-6+) dimensions. Finally, built upon the numerical tools are new applications with initial focus upon chemistry, atomic and molecular physics, material science, and nuclear structure.

Language: C++

Parallelism: MPI, pthreads

MPI API: Blocking Collectives, Blocking P2P, Communicator Management, Non-blocking P2P

5 MPI API Usage Summary

- **Blocking Collectives**: 25/25
- **Blocking P2P**: 17/25
- **Communicator Management**: 16/25
- **Datatypes**: 5/25
- **File Manipulation**: 3/25
- **Non-blocking Collectives**: 1/25
- **Non-blocking P2P**: 20/25
- **One-sided Communication**: 2/25
- **Persistent Communication**: 2/25

References

[1] Jim Dinan, Pavan Balaji, and Jeff R. Hammond. Aggregate Remote Memory Copy (ARMCI) Library. `http://wiki.mpich.org/armci-mpi/index.php`, 2013.

[2] NASA Advanced Supercomputing Division. NAS Parallel Benchmarks. `http://www.mantevo.org/`, 2009.

[3] Michael Heroux and Richard Barrett. The Mantevo Project. `http://www.mantevo.org/`, 2013.

[4] Lawrence Livermore National Laboratories. ASC Sequoia Benchmark Codes. `https://asc.llnl.gov/sequoia/benchmarks/`, 2008.

[5] Jack Poulson, Bryan Marker, Robert A. van de Geijn, Jeff R. Hammond, and Nichols A. Romero. Elemental: A New Framework for Distributed Memory Dense Matrix Computations. *ACM Transactions on Mathematical Software*, 39(2):13:1–13:24, February 2013.

[6] Jason Riedy and Jeremiah Willcock. The Graph500 Search Benchmark. `http://www.graph500.org/`, 2013.

[7] Edgar Solomonik, Devin Matthews, Jeff R. Hammond, and James Demmel. Cyclops Tensor Framework: reducing communication and eliminating load imbalance in massively parallel contractions. *IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2013.

[8] Scott W. Thornton, Nicholas Vence, and Robert E. Harrison. Introducing the MADNESS numerical framework for petascale computing. *Proceedings of the Cray User Group Conference*, 2009.