

ALTERNATIVE HIGH PERFORMANCE COMPUTING BENCHMARKS

KURT ROBERT RUDOLPH (RUDOLPH9)

ABSTRACT. Benchmarks for HPC systems generally focus on floating-point intensive calculations. Few address data intensive graph operations, a class of computation greatly growing in demand. As an alternative to the standard floating-point benchmarks a new benchmark, the Graph 500, has been proposed. Implementations are provided in MPI and OpenMP.

The focus of this project is exploring additional implementations of the Graph500 benchmark. Specifically the project is looking at the UPC programming model, a partitioned global address space language that extends C.

UPC is available on several HPC platforms and will be available on the Blue Waters system here at the University of Illinois Urbana-Champaign.

The variety of parallel programming models utilized in the Graph500 benchmark provide valuable insight into discrete performance aspects of these HPC systems. The UPC implementation of this project offers the additional perspective of a partitioned global address space model and further identifies discrepancies in performance of HPC systems.

1. GRAPH 500 BENCHMARK

Unique from other available benchmarks, the Graph 500 benchmark focuses on graph operation rather than floating point operation. Additionally it employs multiple implementations of the data intensive graph operations performed.

Benchmarks which employ focus on

2. PARALLEL PROGRAMMING MODELS

2.1. Unified Parallel C (UPC). The focus of this project has been exploring a Unified Parallel C implementation of the Graph500 benchmark. Throughout the duration of the project the various aspects of the language have been explored.

Unified Parallel C is an extension of C which employs the partitioned global address space (PGAS) model. Specifically the language employs what are referred to as *shared pointer*. These shared pointers are the center point of the language and are utilized to employ the PGAS model of programming.

2.2. Message Passing Interface (MPI). Currently the first for implementing distributed parallel programs is generally MPI. Rarely are other models used. Highly tested, proven results and widely adopted MPI is simply the practical choice over other parallel models. While PGAS languages such as UPC have potential they currently are not developed enough to contend with MPI. The Graph 500 V1.2 employs multiple implementations in

MPI. The "simple" MPI implementation achieves two-sided communication through standard *send* and *receive* communicators. The "one_{sided}" MPI implementation achieves one-sided communication through MPI windows, a feature built into MPICH2 API. By creating an MPI window

2.3. OpenMP.