1. Problem Statement

Existing dynamic loop scheduling (DLS) techniques for distributed-memory systems employ a master-worker execution model which has a limited performance on large-scale and heterogeneous computing resources.

2. Existing Approaches

Conventional master-worker execution model using MPI two-sided communications [1, 5].

Hierarchical master-worker model using MPI two-sided communications. Global and local masters are located on a single physical compute node [2].

Hierarchical master-worker model using hybrid MPI two-sided communications and OpenMP. Local masters are distributed across multiple physical compute nodes [4].

3. Novel Distributed Chunk Calculation Approach

Novel distributed chunk calculation approach using MPI one-sided communication and passive-target synchronization.

4. Experimental Setup and Results

- Parallel spin-image generation.
- Three two-socket Intel Xeon E5-2640 processors with a total of 20 cores per node, denoted Xeon.
- Three Intel Xeon Phi 7210 manycore processors with a total of 64 cores per node, denoted KNL.

Implementation approaches

- **One_DLS** proposed distributed chunk calculation using one-sided MPI communication and passive-target synchronization.
- **Two_DLS_S** single-thread master-worker using two-sided MPI communication.
- **Two_DLS_M** multi-thread master-worker using two-sided MPI communication.

5. Take Home Messages

- The proposed approach, **DLS_One**, employs MPI passive-target synchronization and delivers a competitive performance against existing approaches, **DLS_Two_S**, and **DLS_Two_M**, that use MPI two-sided communication and employ the conventional master-worker execution model.

- Using **DLS_One**, the performance of DLS techniques were almost unaffected by the arbitrary mapping of the DATAOWNER to any processing element in the system.

References


Acknowledgment

This work was supported by the Swiss National Science Foundation in the context of the “Multi-level Scheduling in Large Scale High Performance Computers” (MLS) grant number 169123.