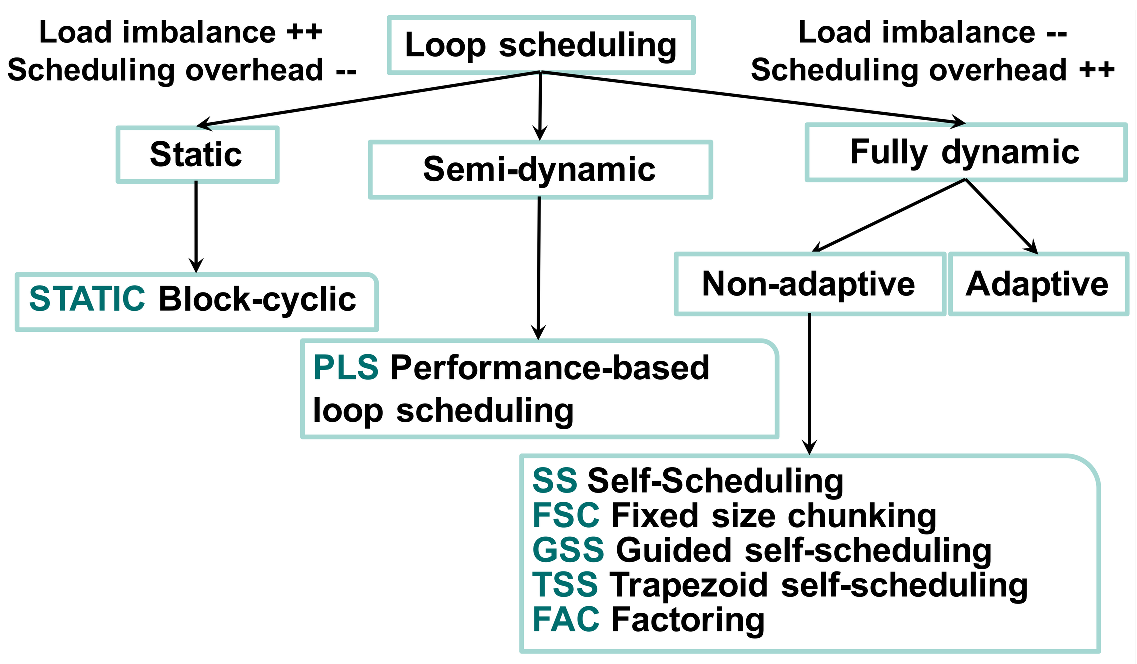
Dynamic Loop Scheduling Using the MPI Passive-Target Remote Memory Access Model

University of Basel

Ahmed Eleliemy and Florina. M. Ciorba
{firstname.lastname}@unibas.ch

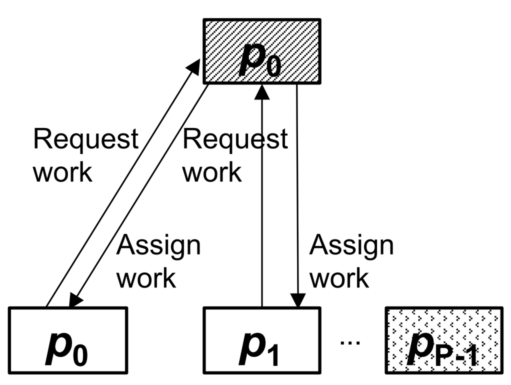
1. Problem Statement

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.

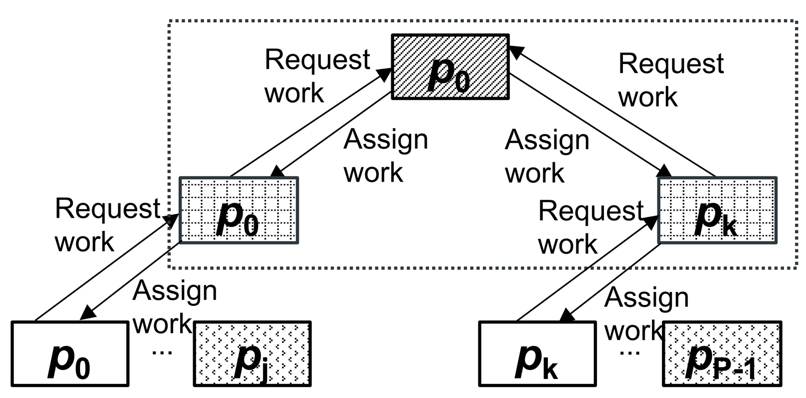


Taxonomy of loop scheduling techniques.

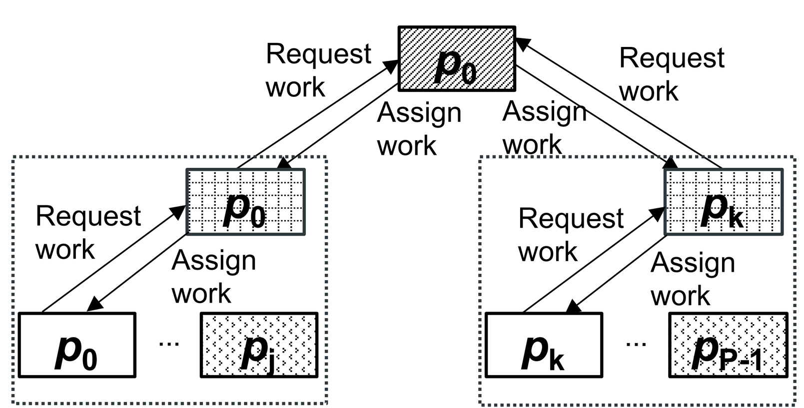
2. Existing Approaches



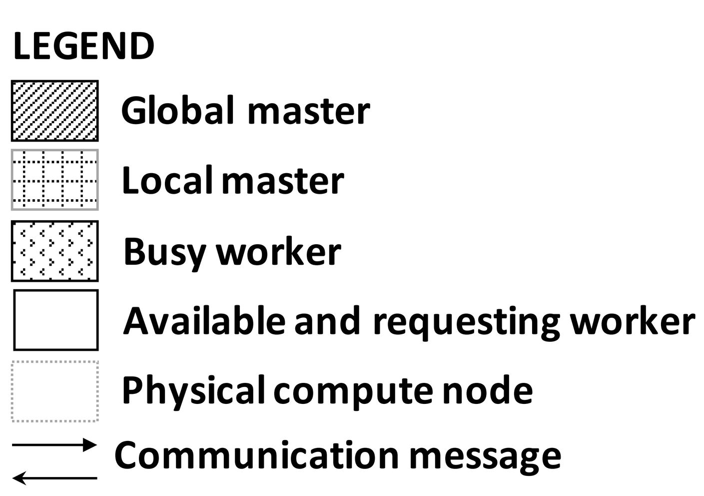
Conventional master-worker execution model using MPI two-sided communications [1, 3].



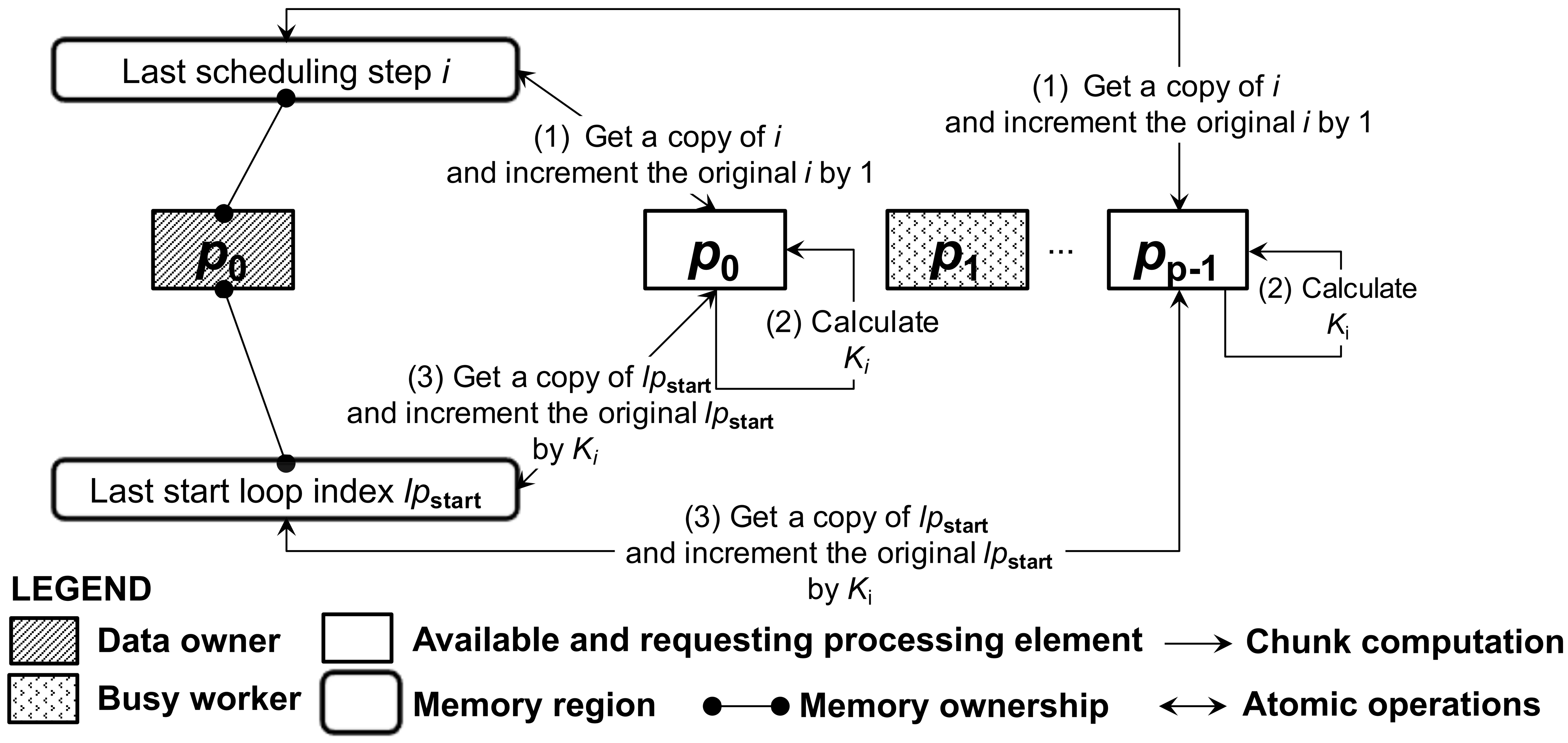
Hierarchical master-worker model using MPI two-sided communications. Global and local masters are located on a single physical compute node [2].



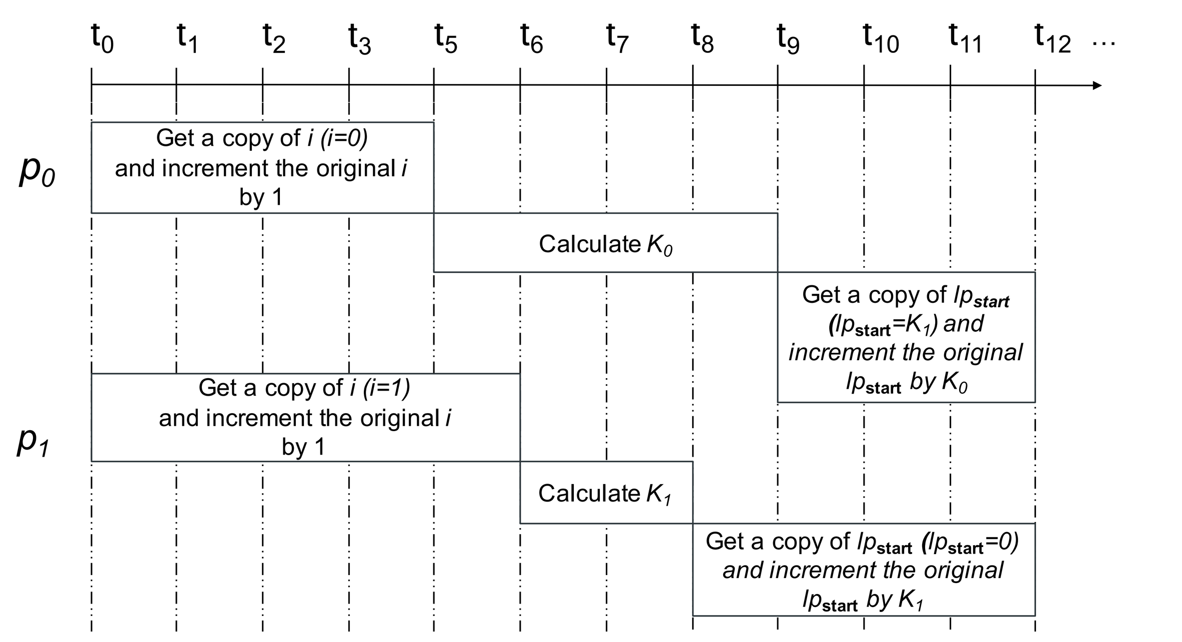
Hierarchical master-worker model using hybrid MPI two-sided communications and OpenMP. Local masters are distributed across multiple physical compute nodes [4].



3. Novel Distributed Chunk Calculation Approach



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

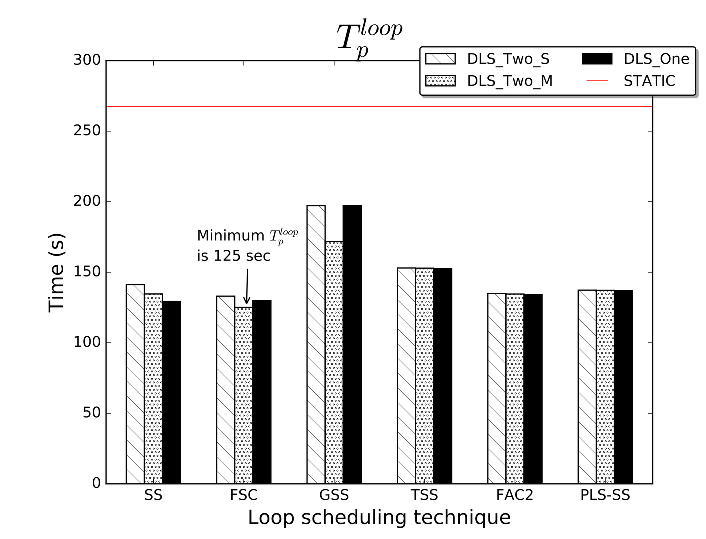


DLS execution with the proposed distributed chunk calculation approach.

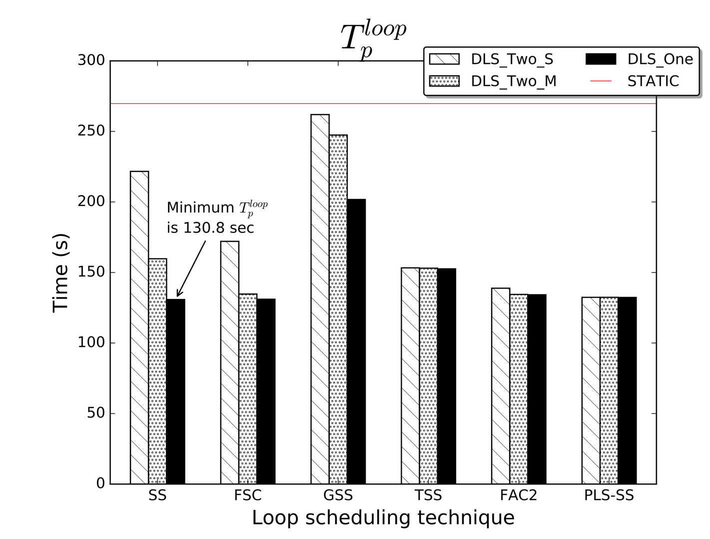
1. A processing element p_j obtains a copy of the last scheduling step i and atomically increments i by one.
2. p_j only uses its local pre-increment copy of i to calculate K_i with the selected DLS technique.
3. p_j obtains a copy of the last start loop index lp_{start} and atomically accumulates the size of the calculated chunk K_i to it.
4. p_j executes loop iterations between lp_{start} (before accumulation) and $lp_{start} + min(K_i, N)$.

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*.



DATAOWNER is pinned to a Xeon core



DATAOWNER is pinned to a KNL core.

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.

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.

References

[1] Chronopoulos, A.T., Andonie, R., Benche, M., and Grosu, D. "A class of loop self-scheduling for heterogeneous clusters", International Conference on Cluster Computing, 2001.

[2] Chronopoulos, A.T., Penmatsa, S., Yu, N., and Yu, D. "Scalable Loop Self-Scheduling Schemes for Heterogeneous Clusters", International Journal of Computational Science and Engineering, 2005.

[3] Cariñ̃o, R.L. and Banicescu, I. "A load balancing tool for distributed parallel loops", Journal of Cluster Computing, 2005.

[4] Wu, C.C., Yang, C.T., Lai, K.C., and Chiu, P.H. "Designing parallel loop self-scheduling schemes using the hybrid MPI and OpenMP programming model for multi-core grid systems", Journal of Supercomputing, 2012.