Efficient Multi-Processor Scheduling in Increasingly Realistic Models

P. A. Papp, G. Anegg, A. Karanasiou and A. N. Yzelman, with ongoing work by T. Böhnlein, R. S. Steiner and B. Lozes

Huawei Zurich Research Center

{pal.andras.papp, albertjan.yzelman}@huawei.com

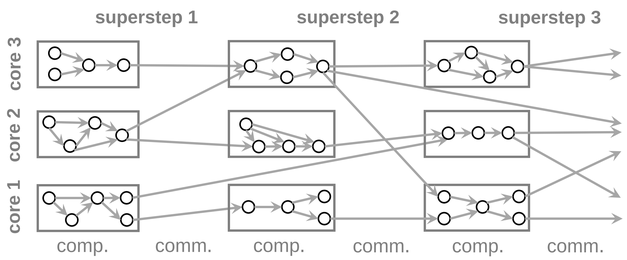
Main Objective

Designing optimal scheduling tools in realistic models

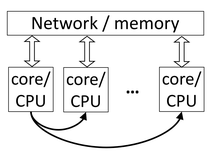
Our work focuses on
- scheduling algorithms in **more realistic models**
- algorithms to schedule **any computational DAG**
- finding **optimal or close-to-optimal schedules**

BSP model for DAG scheduling

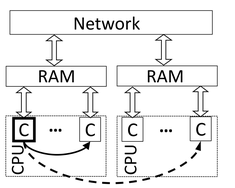
- Supersteps with computation and communication phases



- Offers a **realistic** cost for any computational DAG
  - extended with **hierarchical NUMA effects**



uniform send costs



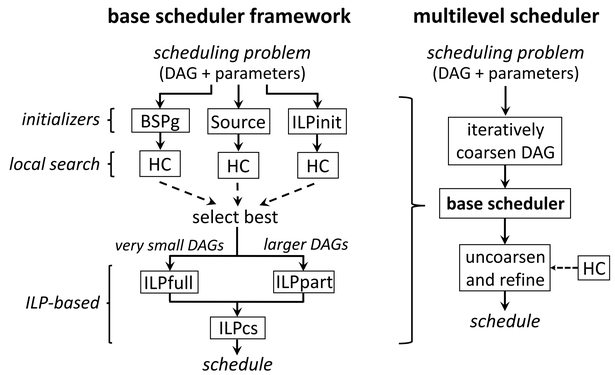
non-uniform (hierarchical)

Database of computational DAGs

- DAGs from various areas: algebraic programming applications, machine learning, graph algorithms
- Fine-grained DAGs of sparse computations
- For our experiments, and as a future benchmark

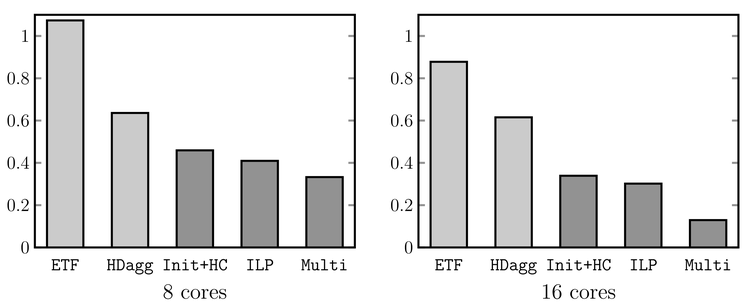
Scheduling algorithms

- Several initializers for initial schedule
- Hill climbing local search (`HC`) to improve solutions
- **ILP-based** schedulers:
  - `ILPfull`: entire problem as ILP (limited to small DAGs)
  - `ILPpart`: iteratively improve smaller parts of schedule
  - `ILPcs`: capture subproblem of communication scheduling
- **Multilevel** scheduling algorithm:
  - coarsen into much smaller DAGs, retaining structure
  - find BSP schedule for coarse DAG
  - iteratively uncoarsen, and refine in each step with `HC`



Model-based evaluation

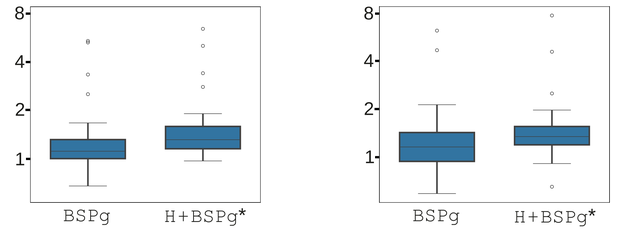
- Extensive comparison to **Cilk**, **ETF** and **HDagg** baselines



improvement ratio on 66 DAGs from our database, normalized to Cilk

Run-time evaluation

- Speeding up **sparse triangular solvers** via scheduling
- Baseline: **HDagg**-accelerated SpTRSV
- Using faster & improved versions of our schedulers
  - time complexity identical to HDagg



speedup on 33 DAGs from HDagg evaluation dataset, on Xeon Gold 6238T, 44 cores, 192 GB memory, and theoretical peak memory throughput 262.2 GB/s

**BSPg with the HDagg coarsener*

Results

Model-based evaluation (BSP):
- **24%** (UMA) and **43%** (NUMA) faster than HDagg
- **up to 4.5x** faster for some machine parameters

Run-time evaluation (SpTRSV):
- **48%** (22 cores) and **45%** (44 cores) faster than HDagg
- **up to 6.5x** (22 cores) and **7.8x** (44 cores) faster

Links and resources

Full paper
https://arxiv.org/abs/2404.15246

Our schedulers
https://github.com/AlgebraicProgramming/OneStopParallel

DAG database
https://github.com/AlgebraicProgramming/HyperDAG_DB