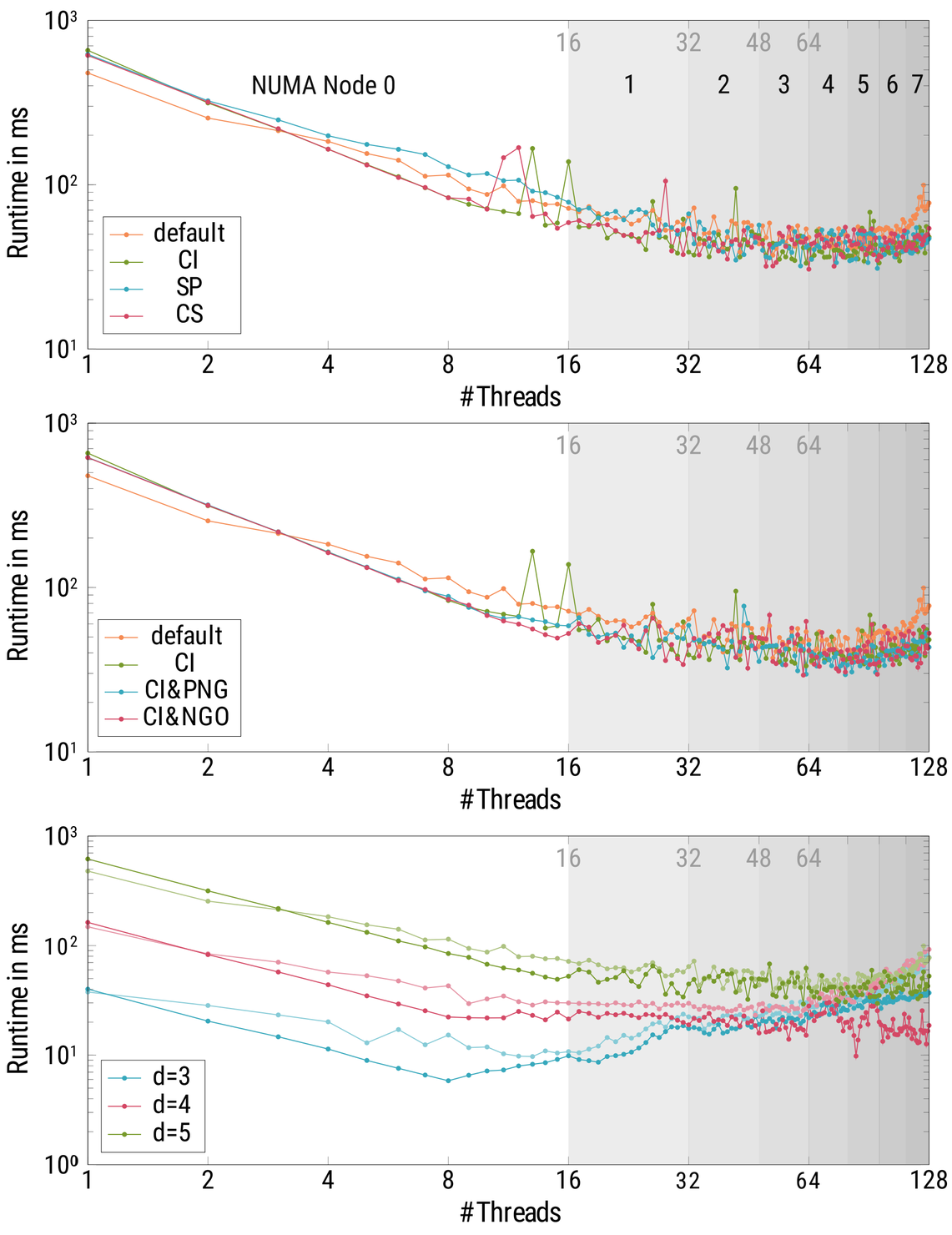
TECHNISCHE UNIVERSITÄT
CHEMNITZ

Fakultät für Informatik
Professur Betriebssysteme
Laura Morgenstern

Towards a NUMA-aware Task-based Fast Multipole Method

Project Idea

Molecular dynamics (MD) simulations have become a vital research method in biochemistry and materials science. In MD, the fast multipole method (FMM) is used to reduce the complexity of the computation of pairwise long-range interactions. Since the computation of long-range interactions is the most expensive part of MD, its efficiency and scalability is decisive for the whole simulation. The goal of this project is to analyze and improve the scalability of the FMM on modern multicore systems. Today's multicore systems are based on NUMA (Non-Uniform Memory Access) architectures for reasons of scalability. To exploit the scalability of the hardware, algorithms have to take data locality into account. Regarding the FMM, we present the NUMA-aware data and thread placement policies Scatter Principally (SP), Compact Ideally (CI) and Compact Scatter (CS) as well as the NUMA-aware load balancing policies Prefer NUMA-Group (PNG) and NUMA-Group Only (NGO) in [1]. In this project, we examine the performance of those policies on one of the Future SOC Lab's multicore servers with eight NUMA nodes, with each NUMA node being an Intel Xeon X7560 with 8 cores and 2-way SMT (Simultaneous Multithreading).



Data Locality Speeds Up Molecular Dynamics

- 1000 particles, tree depth d=5, multipoleorder p=3
- FMM with NUMA-aware data placement is up to **2.14 times faster** than default FMM
- Unfavorable runtime peaks of CI and CS are due to SMT

NUMA-aware Load-Balancing for the Win

- 1000 particles, tree depth d=5, multipoleorder p=3
- FMM with NUMA-aware data placement and NUMA-aware load balancing is up to **2.43 times faster** than default FMM
- Load balancing only pays off between threads running on the same NUMA node

Upside Down: Doing Less in More Time

- 1000 particles, multipoleorder p=3, NUMA CI and NGO
- If the computational effort is too low (see d=3), using cores located on remote NUMA nodes increases runtime
- However, computational effort per compute node will decrease further

Conclusion and Future Work

- NUMA-awareness is crucial for performance
- Don't load balance between NUMA nodes
- For very low computational effort, we still don't scale
- Develop a less strict thread pinning policy
- Develop a generic NUMA-aware pool allocator
- Reduce data transfer between NUMA nodes further

HPI Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

Bibliography

[1] Laura Morgenstern. A NUMA-Aware Task-Based Load-Balancing Scheme for the Fast Multipole Method. Master's thesis, Chemnitz University of Technology, 2017.