Application Specific Computing

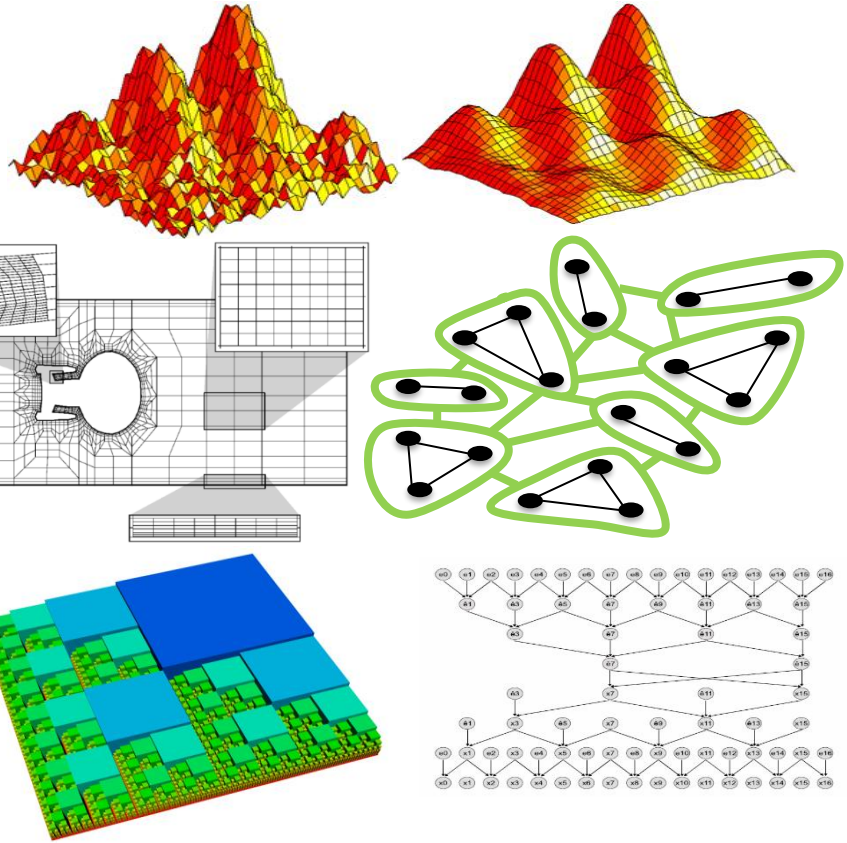
Robert Strzodka
Application Specific Computing
ZITI, Heidelberg University
http://asc.ziti.uni-heidelberg.de

Application Specific Computing

Prof. Dr. Robert Strzodka, ZITI

Parallel Algorithms and Hardware

- Ultimate efficiency and performance
  - GPU, many-core CPU (≥100th)
- Data representation
  - mixed-precision, compression, redundancy
- Data access
  - layout, spatial and temporal locality
- Data structure
  - unstructured grids, graphs, adaptivity
- Numerical methods
  - GMG, AMG, Krylov, ILU, FMM, MC
- Programming abstractions
  - CUDA, C++2x, PSTL, TBB, own libs
- https://asc.ziti.uni-heidelberg.de/



C++23 Ranges and Views

- What are the advantages over iterators?
- How do they help composability?
  - Ranges library (since C++20) - cppreference.com
  - C++20 Ranges — Complete Guide | ITNEXT
  - Understanding Ranges Views and View Adaptors
  - A Plan for C++26 Ranges (open-std.org)
  - P3351R2 views::scan

Supervisor: Wilfredo Colmenares



In the field of HPC

Most of the code in the field of HPC was written in C, C++ or Fortran. Julia has seen a surge in adoption due to its user-friendly syntax, LLVM support and wide hardware interoperability. How good of an alternative is Julia to other languages in the HPC space?

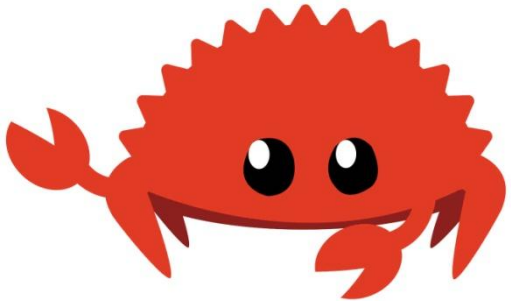
- https://dl.acm.org/doi/10.1145/3624062.3624278
- https://dl.acm.org/doi/10.1109/SCW63240.2024.00245
- https://arxiv.org/abs/2309.17309
- https://benlauwens.github.io/ThinkJulia.jl/latest/book.html

Supervisor: Alessandro Biagiotti

Rust - Future in HPC ?

Rust is a new systems-programming language with ever-increasing adoption.
The language offers
- Memory safety
- Fearless concurrency
- Better Developer Experience
- Zero Cost Abstractions

Making it a very promising language for HPC



- The Rust Programming Language
- Introduction - The Rust Performance Book
- Towards Safe HPC: Productivity and Performance via Rust
- Performance vs Programming Effort between Rust and C - Case Study

Supervisor: Fabian Osyguß

Interface and Performance of  Ginkgo

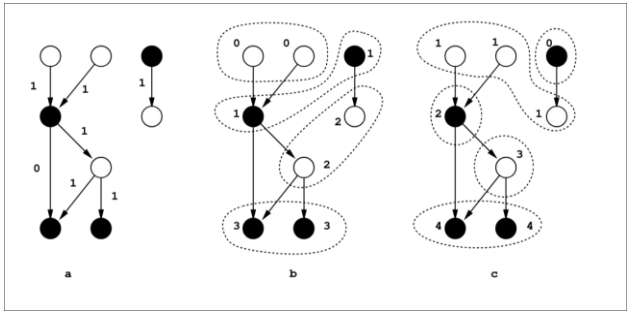
- Ginkgo is a high-performance linear algebra library
- Focus on solution of sparse linear systems
- Sophisticated separation of core algorithms from architecture-specific backends

- [2006.16852] Ginkgo: A Modern Linear Operator Algebra Framework for High Performance Computing
- Ginkgo - A math library designed to accelerate Exascale Computing Project science applications

Supervisor: Fabian Osyguß

The problem of loop fusion

Given a scientific code, finding the reordering that fuses the most loops is an NP-hard problem, what are the strategies to solve it out there? What is their complexity? What structures do they use?

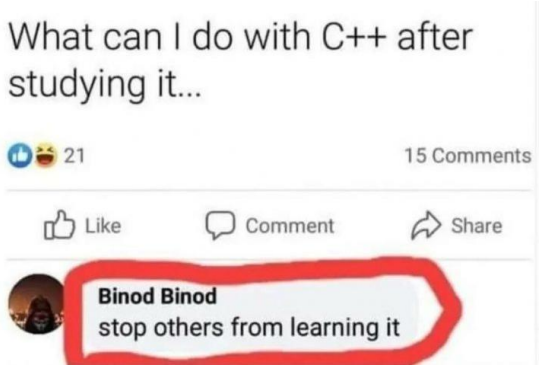


- https://ieeexplore.ieee.org/abstract/document/807510
- https://dl.acm.org/doi/10.1145/3416510

Supervisor: Alessandro Biagiotti

Frontend for scientific computing DSLs

We want to explore the possibilities of adding a frontend to a DSL for scientific codes, the objective is to remove any obstacles to widespread adoption (spoiler: we don't want the frontend to be C++).



- https://github.com/nushio3/formura
- https://dl.acm.org/doi/10.1145/3374916

Supervisor: Alessandro Biagiotti

Reflecting on reflection

Reflection is already available in the latest version of the C++ standard, we want to understand its advantages in complex and highly templated code, especially when it comes to compile-time calculations.

- https://en.cppreference.com/cpp/meta/reflection
- https://isocpp.org/files/papers/P2996R4.html
- Reflecting on reflection (iso cpp conference)
- Reflection Is Only Half the Story



Supervisor: Alessandro Biagiotti

Highway: A C++ library that provides portable SIMD/vector intrinsics.

- CPUs provide SIMD/vector instructions that apply the same operation to multiple data items. This can reduce energy usage e.g. *fivefold* because fewer instructions are executed.

- https://github.com/google/highway/tree/master
- https://www.mnm-team.org/pub/Fopras/rock23/PDF-Version/rock23.pdf

Supervisor: Xingjian Qi

Space Filling Curves is All You Need: Communication-Avoiding Matrix Multiplication Made Simple

- Revisiting Space Filling Curves (SFC) to alleviate complicated tuning like cache blocking. A Matrix Multiplication is partitioned using advancements in SFC, and obtain platform-oblivious and shape-oblivious Matrix Multiplication schemes with high degree of data locality.
- https://arxiv.org/abs/2601.16294

Supervisor: Wilfredo Colmenares

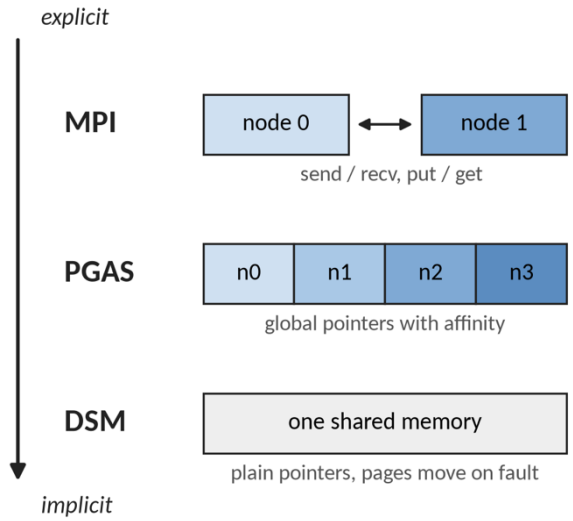
Emulating High-precision Matrix Operations on Low-precision Matrix Engines

- Nvidia H100 provide 30 TFLOPs for FP64 and 3341 TFLOPs for INT8(tensor cores), offering ~100x speedups.
- Emulating FP64 with INT8 using tensor cores
  - Ozaki Scheme:
    - 1. Error-free transformations of matrix multiplication by using fast routines of matrix multiplication and its applications.
    - 2. Generalization of error-free transformation for matrix multiplication and its application.
    - 3. Ozaki Scheme II: A GEMM-oriented emulation of floating-point matrix multiplication using an integer modular technique.
- Emulating FP32 with FP16 using tensor cores
  - Compensated summation + rounding with RN(2203.03341)

Supervisor: Xingjian Qi

MPI, PGAS, DSM: Programming Clusters

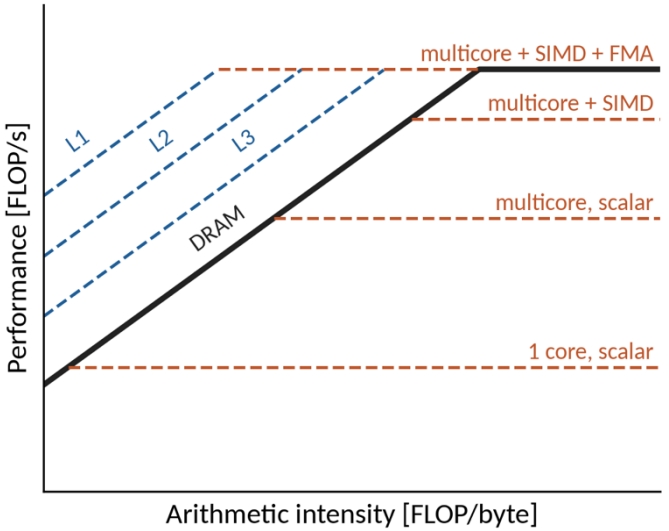
- How much must the programmer know about where data lives?
- Why, or why not Distributed Shared Memory?
- Starting points
  - Remote Memory Access Programming in MPI-3 (ACM TOPC 2015)
  - Partitioned Global Address Space Languages (ACM CSUR 2015)
  - ArgoDSM: Turning Centralized Coherence and Distributed Critical-Section Execution on their Head (HPDC 2015)
  - UPC++ (PGAS library, docs and tutorials)
  - …



Supervisor: Björn Steinorth

The Roofline Model and Its Extensions

- What limits a kernel: compute or memory – and which level of the memory hierarchy?
- How do SIMD, FMA and multiple cores change the achievable peak?
- Starting points
  - Roofline: An Insightful Visual Performance Model for Multicore Architectures (CACM 2009)
  - Cache-aware Roofline Model: Upgrading the Loft (IEEE CAL 2014)
  - LIKWID (tools to measure bandwidths and peak performance)
  - …



Supervisor: Björn Steinorth