Orchid: Towards Heterogeneous Batched Eigenvalue Solvers

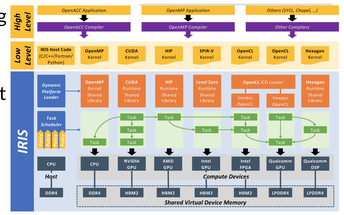
Matthew Chung[1], Mohammad Alaul Haque Monil (Advisor)[2], Keita Teranishi (Advisor)[2], Narasinga Rao Miniskar (Advisor)[2]
[1]University of California, Riverside [2]Oak Ridge National Laboratory

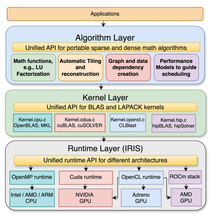
Abstract

There is a growing need for the efficient solution of many small eigenvalue problems (up to N = 1500) that arise in emerging scientific applications. These small-to-medium sized problems present unique computational challenges, particularly when thousands or millions of such problems must be solved repeatedly. This work presents Orchid, a novel distributed, heterogeneous, batched eigenvalue solver based on the IRIS runtime. Orchid can utilize all compute platforms in both heterogeneous nodes and clusters by harnessing the capabilities of the IRIS architecture. Orchid leverages heterogeneous architectures across multiple nodes by partitioning the application task DAG intelligently and orchestrates multiple instances of the IRIS runtime via MPI. We evaluate our proposal against two heterogeneous hardware configurations and Frontier, demonstrating Orchid's performance utilizing both intra-node and inter-node heterogeneity.

Background

- IRIS[3] provides a task-based programming model and runtime environment for extremely heterogeneous systems
- Supports heterogeneous execution against all major compute platforms, including CUDA, HIP, OpenCL, and OpenMP
- Provides automated data movement and DAG creation/fusion





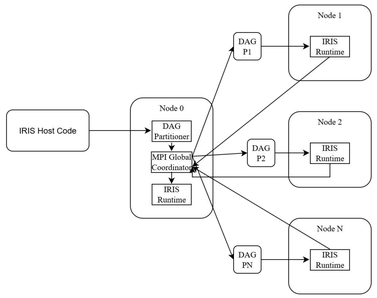
- MatRIS[2] is a performance portable math library for IRIS
- Sits on top of IRIS, provides a collection of key BLAS/ LAPACK routines (GETRF, POTRF, TRSM, GEMM, etc)
- Unified interface for invoking vendor specific and open source kernels
- Provides core features for extending MatRIS to include additional algorithms (i.e. eigenvalue solvers)

Goals

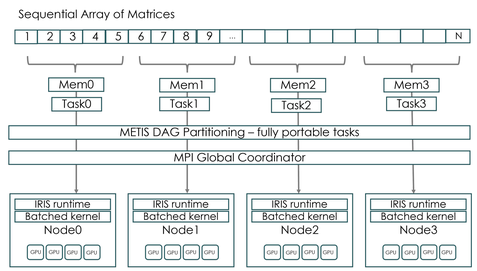
- We focus on developing a **productive, portable, and heterogeneous** framework for a batched eigenvalue solver
- **Productive:** Batched eigenvalue solver is practical and simple to apply to scientific applications. Provided through IRIS/MatRIS programming model - "code once, run anywhere".
- **Portable:** Solver can be performantly executed on a diverse set of compute platforms. EigenG-Batched[4] backend provides competitive performance than cuSolver while supporting execution on both NVIDIA and AMD GPUs.
- **Heterogeneous:** Solver concurrently executes on heterogeneous, distributed architectures, fully utilizing all compute resources. IRIS runtime system provides intra-node heterogeneity, our proposal provides features for distributed execution.

Acknowledgements | This material is based upon work are supported by the DOE ASCR Fairbanks project.

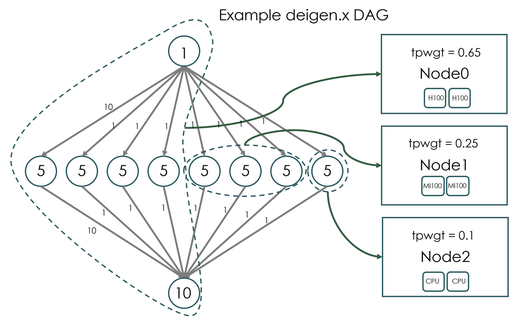
Design/Approach



- **DAG partitioning:** multi-constraint task DAG partitioning using METIS[1] enables capability-aware workload distribution
  - Objective to minimize overall cut weight, results in lower communication overhead and data movement
  - Edge weights model data movement, vertex weights model task metadata, target weights model node capacity
  - METIS accounts for differing capabilities between nodes, ensuring maximum total device utilization and minimal makespan
  - Highly customizable constraints



- Address two challenges for extending IRIS to support a distributed eigenvalue solver: multi-node orchestration and task DAG partitioning.
- **MPI-based Global Coordinator:** Supports running one IRIS runtime per node
  - Handles overall control flow of distributed execution - instantiating all runtimes, distributing workload between nodes, collecting/aggregating results
  - Rank 0 handles initial task DAG construction and partitions DAG according to node capabilities

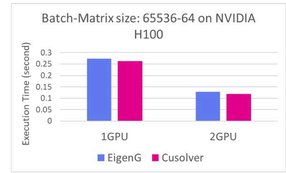


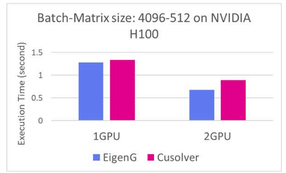
- **MatRIS Integration:** productive implementation of a heterogeneous batched eigenvalue solver
  - Implementation unifies calls to vendor specific libraries
  - We provide two kernels: a CUDA and HIP kernel, enabling compatibility with IRIS
  - IRIS dynamic hardware discovery automates kernel selection, enabling tasks to be sent to multiple heterogeneous devices
  - Kernels build task graphs and handle vendor specific streams, integrate 1D tiling - enables distributed execution

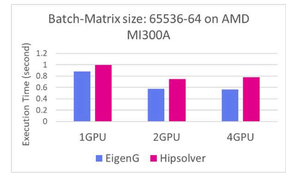
Hardware & Experiment Setup

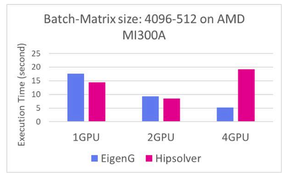
- Hardware Configuration A: **Intra**-node heterogeneity
  - One node of 1x AMD CPU and 1x RTX 3090 + 1x Radeon RX 6900XT
- Hardware Configuration B: **NVIDIA** GPUs
  - One node of 2x AMD CPU and 2x H100 NVL
- Hardware Configuration C: **AMD** GPUs
  - One node of 4x MI300A (CPU + GPU)
- **Experiment purpose**: Evaluate solver against multiple architectures and compare performance to reference vendor solvers
- Hardware Configuration A: batch sizes 4096, 8192 and matrix sizes 64, 256
- Hardware Configuration B/C: batch sizes 4096, 65536 and matrix sizes 64, 512

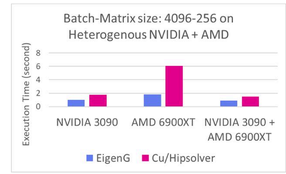
Results

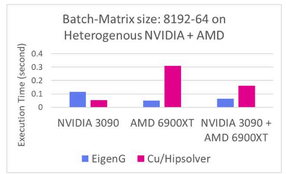












- Examine cases with small batch size, but large matrices and large batch size, but small matrices
- Orchid provides competitive performance to vendor solvers, while providing significant performance gains on heterogeneous nodes
- EigenG-Batched provides scalability in multi-GPU scenarios

Conclusion and Future Work

- Orchid provides a portable and heterogeneous batched eigenvalue solver built on the IRIS runtime. Orchid is capable of operating on top of intra-node and inter-node environments while retaining its portability and a productive programming model
- Move towards enabling batched symmetric eigenvalue solvers to leverage exascale computing resources
- Next steps: Extending IRIS memory model across nodes, adding UCX-based communication backend to allow for cross-node device-to-device communication, performing runtime optimizations

References

[1] Karypis, George & Kumar, Vipin. (1997). METIS—A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes and Computing Fill-Reducing Ordering of Sparse Matrices.
[2] Mohammad Alaul Haque Monil, Narasinga Rao Miniskar, Keita Teranishi, Jeffrey S. Vetter, and Pedro Valero-Lara. 2023. MatRIS: Multi-level Math Library Abstraction for Heterogeneity and Performance Portability using IRIS Runtime. In Proceedings of the SC '23 Workshops of the International Conference on High Performance Computing, Network, Storage, and Analysis (SC-W '23). Association for Computing Machinery, New York, NY, USA, 1081–1092. https://doi.org/10.1145/3624062.3624184
[3] J. Kim, S. Lee, B. Johnston and J. S. Vetter, "IRIS: A Portable Runtime System Exploiting Multiple Heterogeneous Programming Systems," 2021 IEEE High Performance Extreme Computing Conference (HPEC), Waltham, MA, USA, 2021, pp. 1-8, doi: 10.1109/HPEC49654.2021.9622873. keywords: {Performance evaluation;Iris;Runtime;Graphics processing units;Prototypes;Computer architecture;Programming;heterogeneous architectures;runtime systems;compilers},
[4] Y. Hasegawa, T. Imamura, T. Ina, N. Onodera, Y. Asahi and Y. Idomura, "GPU Optimization of Lattice Boltzmann Method with Local Ensemble Transform Kalman Filter," 2022 IEEE/ACM Workshop on Latest Advances in Scalable Algorithms for Large-Scale Heterogeneous Systems (ScalAH), Dallas, TX, USA, 2022, pp. 10-17, doi: 10.1109/ScalAH56622.2022.00007. keywords: {Symmetric matrices;Computational modeling;Graphics processing units;Transforms;Supercomputers;Matrix decomposition;Kalman filters;Ensemble data assimilation;Local ensemble transform Kalman filter;Eigenvalue decomposition},