CHAPELCON'25

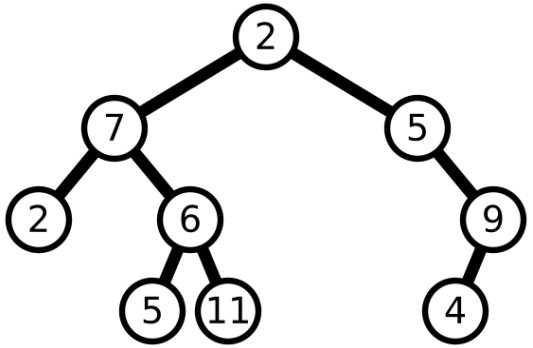
A PORTABLE LOW-LEVEL MULTI-GPU BRANCH-AND-BOUND: A COMPARISON AGAINST CHAPEL

IVAN TAGLIAFERRO[1 2], GUILLAUME HELBECQUE[1], EZHILMATHI KRISHNASAMY[2], NOUREDINE MELAB[1], AND GRÉGOIRE DANOY[2 3]

[1]UNIVERSITÉ DE LILLE, CNRS/CRISTAL, CENTRE INRIA DE L'UNIVERSITÉ DE LILLE, FRANCE

[2]UNIVERSITY OF LUXEMBOURG, FSTM-DCS, LUXEMBOURG

[3]UNIVERSITY OF LUXEMBOURG, SNT, LUXEMBOURG

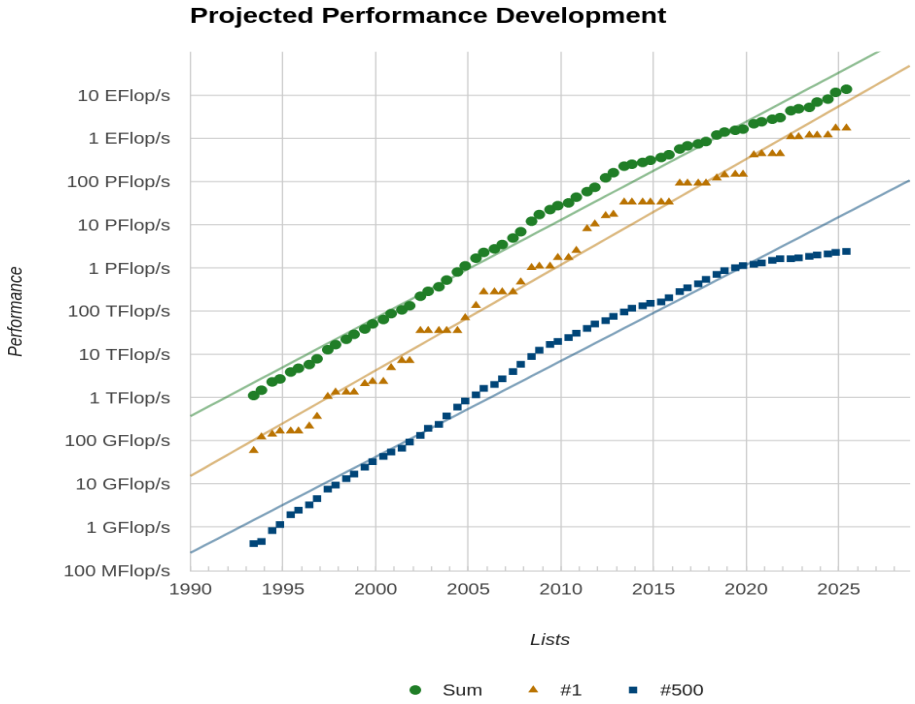


MOTIVATIONS AND CONTEXT

MOTIVATIONS

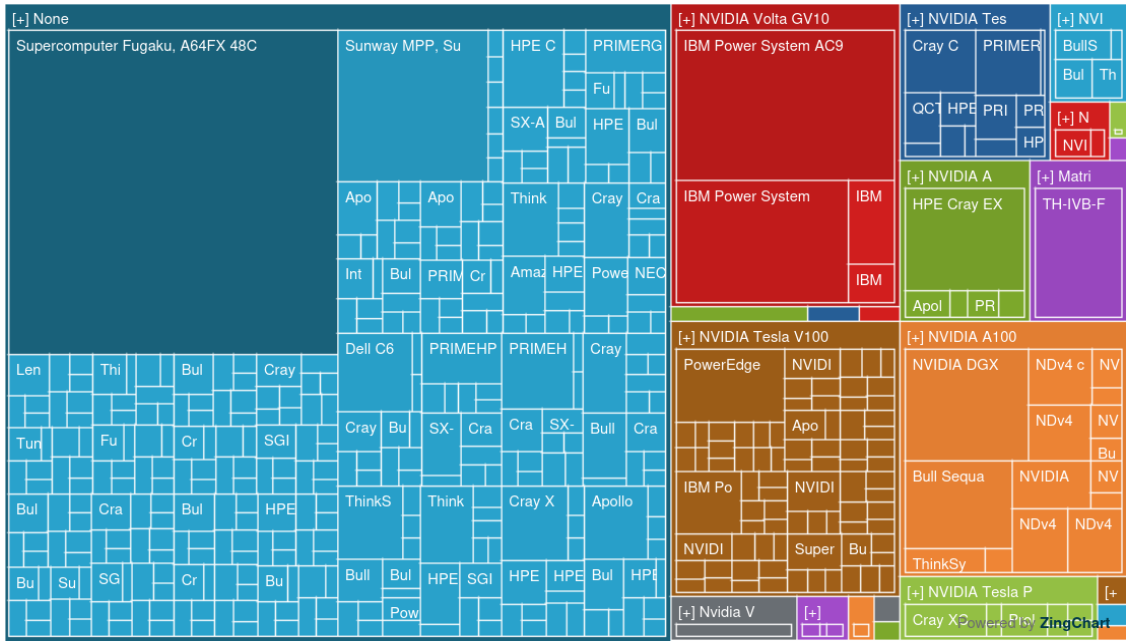
Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
1	**El Capitan** - HPE Cray EX255a, AMD 4th Gen EPYC 24C 1.8GHz, AMD Instinct MI300A, Slingshot-11, TOSS, HPE DOE/NNSA/LLNL United States	11,039,616	1,742.00	2,746.38	29,581
2	**Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE Cray OS, HPE DOE/SC/Oak Ridge National Laboratory United States	9,066,176	1,353.00	2,055.72	24,607
3	**Aurora** - HPE Cray EX - Intel Exascale Compute Blade, Xeon CPU Max 9470 52C 2.4GHz, Intel Data Center GPU Max, Slingshot-11, Intel DOE/SC/Argonne National Laboratory United States	9,264,128	1,012.00	1,980.01	38,698
4	**JUPITER Booster** - BullSequana XH3000, GH Superchip 72C 3GHz, NVIDIA GH200 Superchip, Quad-Rail NVIDIA InfiniBand NDR200, RedHat Enterprise Linux, EVIDEN EuroHPC/FZJ Germany	4,801,344	793.40	930.00	13,088

[Top500]



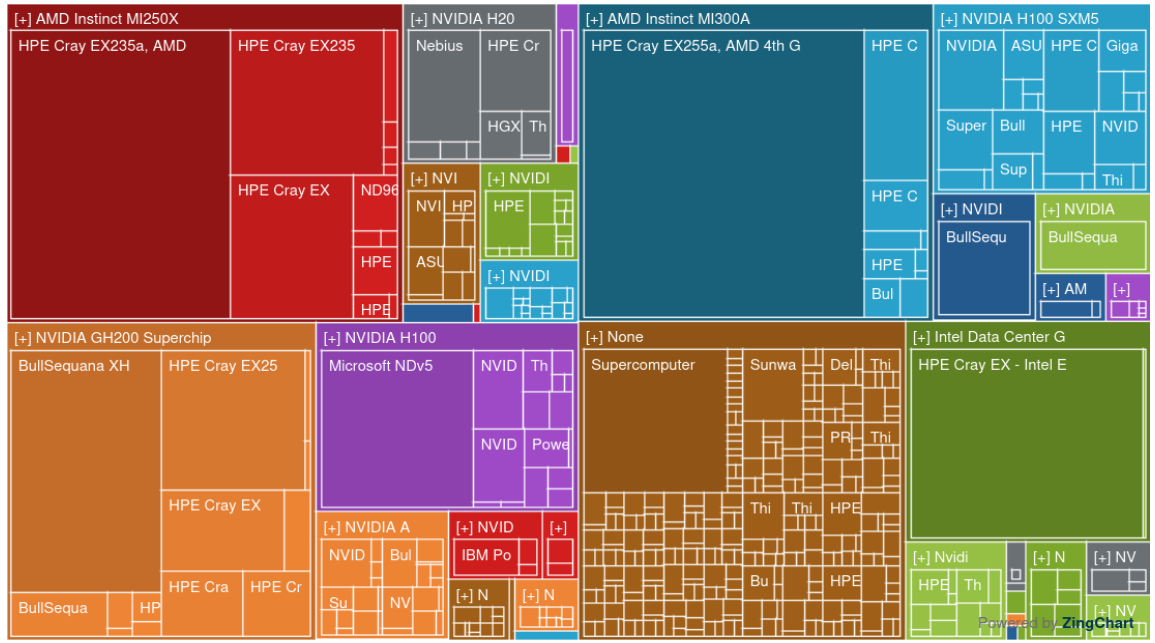
- Exascale *era:*
  - Increasingly large number of cores
  - More and more heterogeneous
  - Less reliable (Mean Time Between Failures < 1h)
- MPI+X *vs.* PGAS [UltraBO]

MOTIVATIONS



[Top500] June 2021

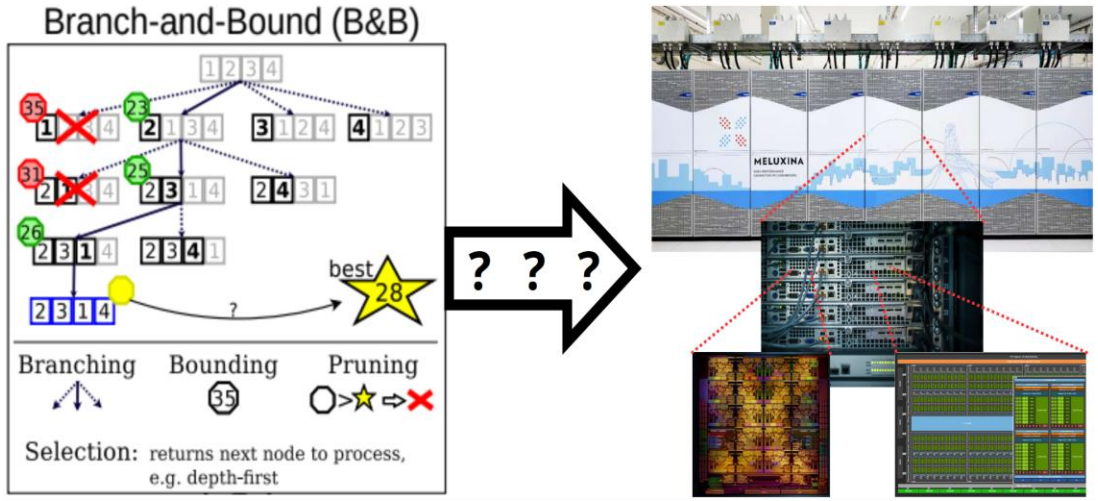
4 Years!!!



[Top500] June 2025

- Increasing usage of accelerators
  - Variety of GPU architectures: Nvidia and AMD

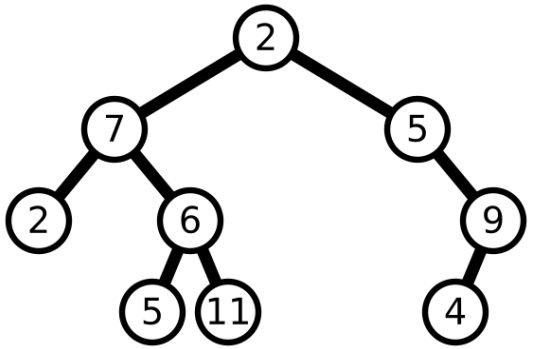
CONTEXT



[UltraBO]

- Distributed GPU-accelerated tree search methods (*e.g.*, B&B) for COPs
  - Large and Irregular search trees
  - Dynamic Load Balancing
  - Efficient Data Structs
  - Low-level Portable Solutions

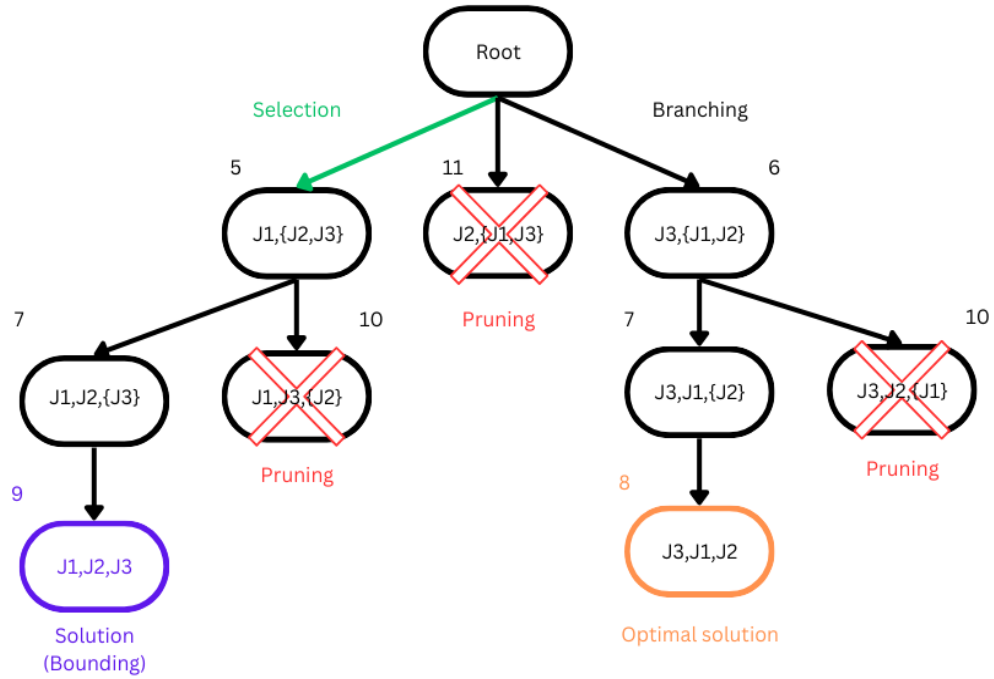
- Motivating example: Permutation Flowshop Scheduling Problem (PFSP)
  - Search trees for hard instances contain up to 10^{15} explored nodes



BACKGROUND

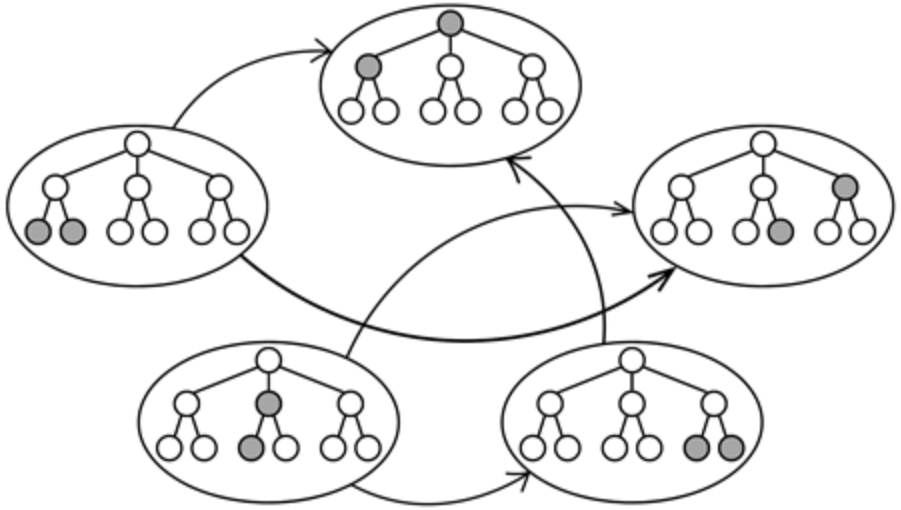
BRANCH-AND-BOUND (B&B)

- Divide a Combinatorial Optimization Problem (COP) into partial sub-problems [Gendron1994]
- Search space is a tree
- Four operators:
  - Branching
  - Selection, *e.g.*:
    - Breadth-first search (FIFO)
    - Depth-first search (LIFO)
  - Bounding
  - Pruning



PARALLEL B&B

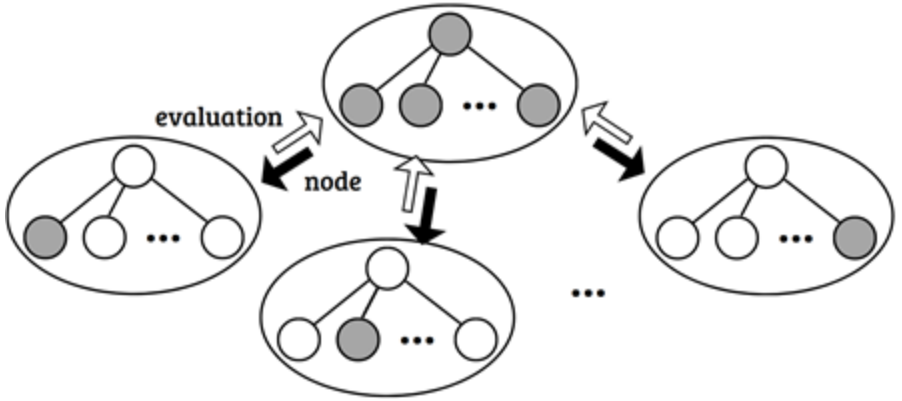
Parallel Tree Exploration (PTE)



- Done on CPU
- High degree of parallelism

+

Parallel Evaluation of Bounds (PEB)



- Done on GPU
- Suited for higher evaluation costs

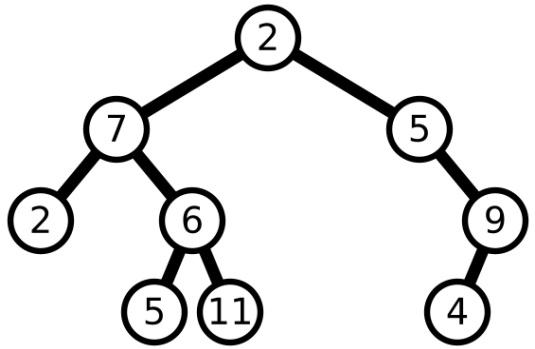
CHAPEL LANGUAGE[1]

- PGAS - "Partitioned Global Address Space" Language [Chamberlain2018]
  - Contrasts with MPI communications model
- Native vendor-neutral GPU support [Chapel2024]
  - Nvidia Architectures (2022)
  - AMD Architectures (2023)
- Native shared-memory multiprocessing directives

[1]The Chapel Language: https://chapel-lang.org/.

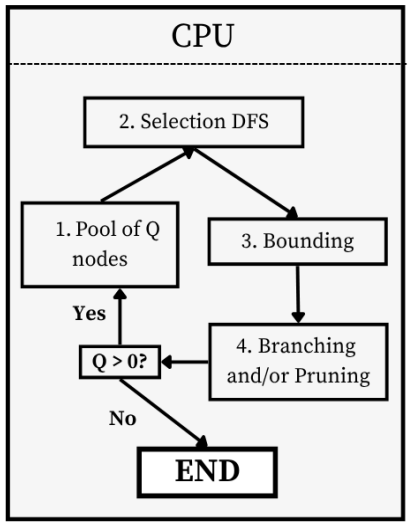
OBJECTIVES

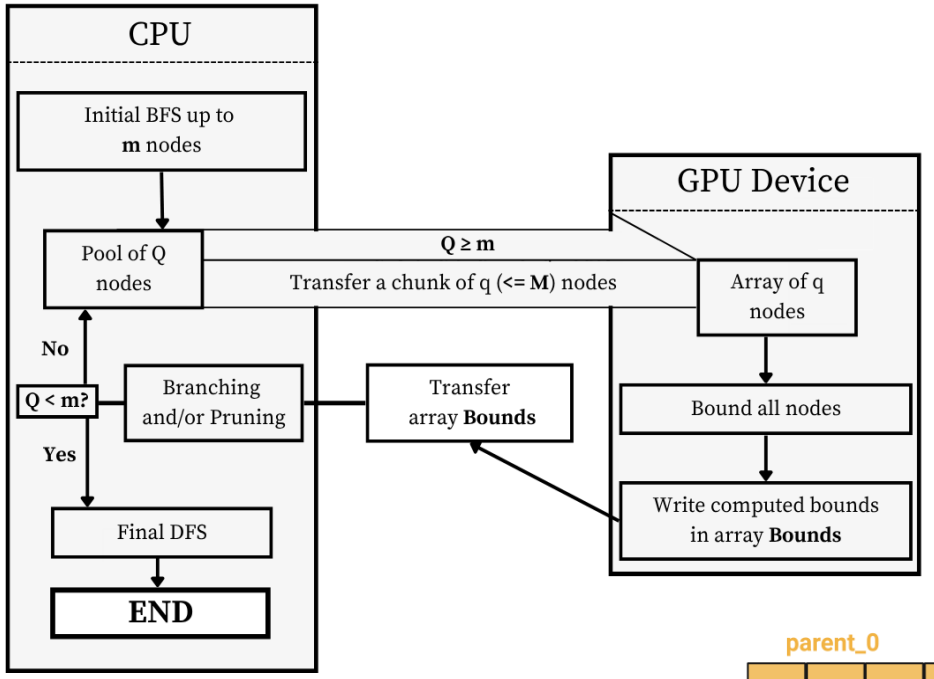
- Low-level (C Language) GPU-portable parallel B&B:
  - Generic processing of tree nodes [Gmys2017, Gmys2022]
  - Dynamic load balancing (OpenMP, atomic spin-lock)
  - CUDA / HIP for Nvidia / AMD architectures [Chakroun2013, Vu2016]
- Towards distributed generic GPU-accelerated parallel B&B:
  - MPI+X *vs.* PGAS (Chapel) at the intra-node level [Carneiro2021, Helbecque2024, Helbecque2025]

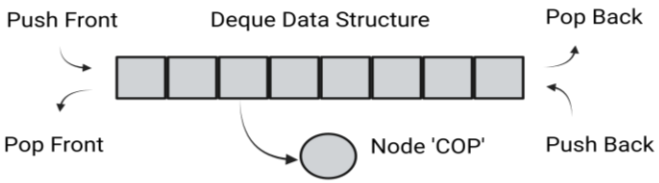


DESIGN AND IMPLEMENTATION

CPU TO SINGLE-GPU (PEB)



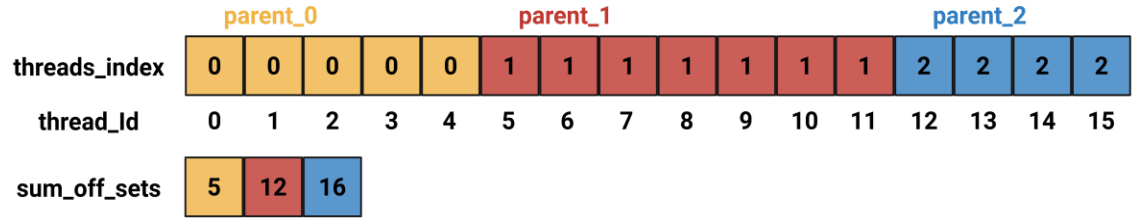




[Helbecque2025]

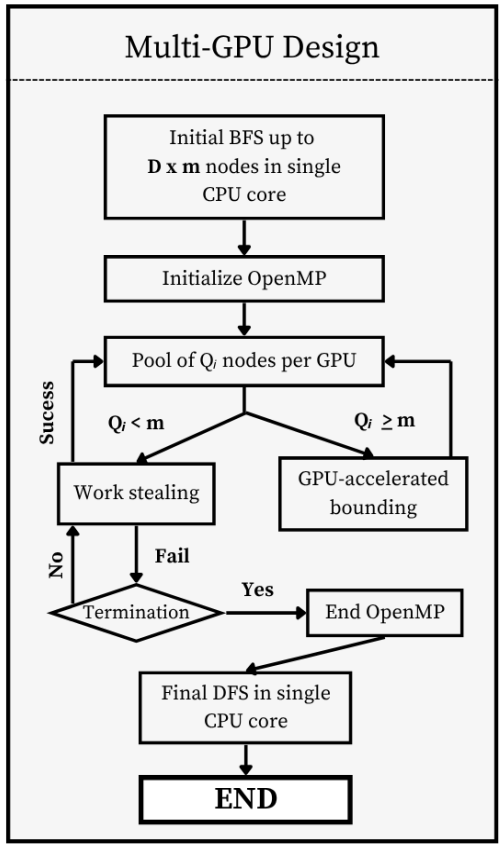
GPU Thread Indexing:

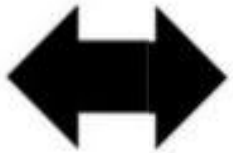
1. Compute amount of child nodes per parent
2. Affect 1 child per thread

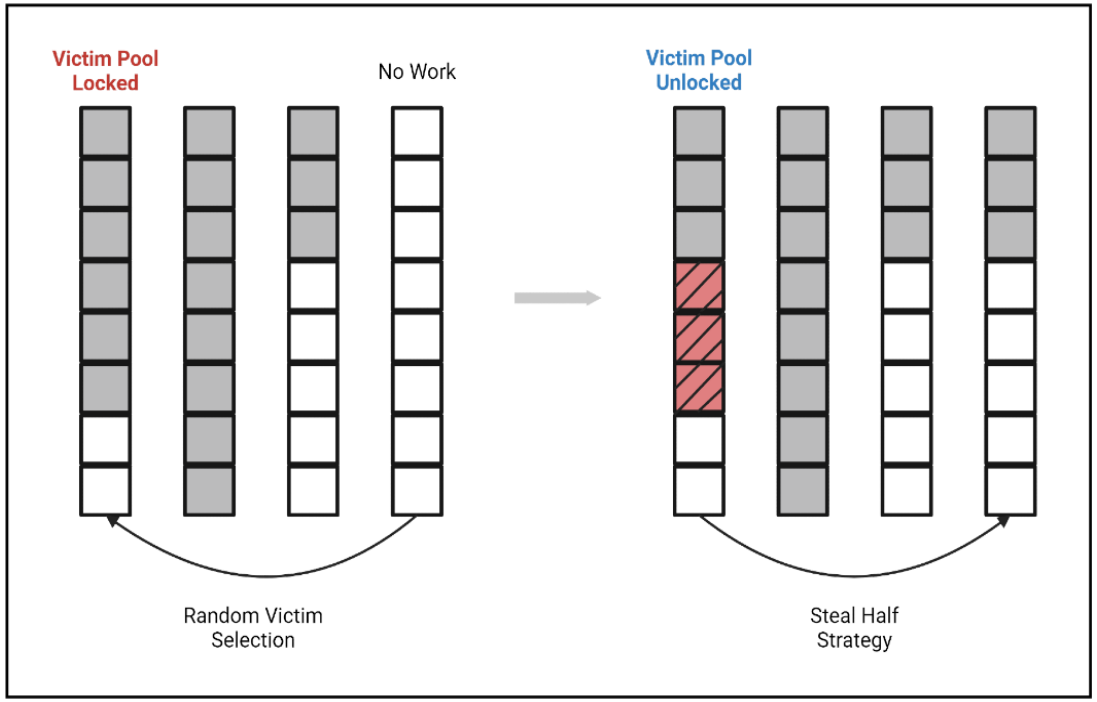


[Chakroun2013]

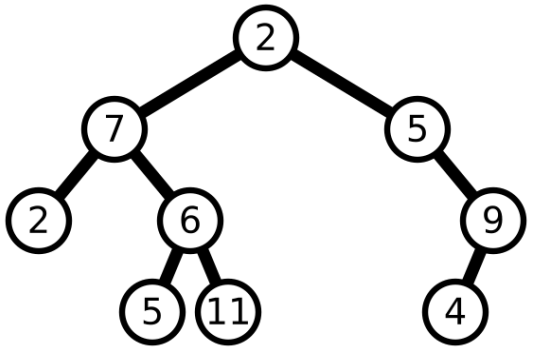
MULTI-GPU DESIGN (PTE + PEB)







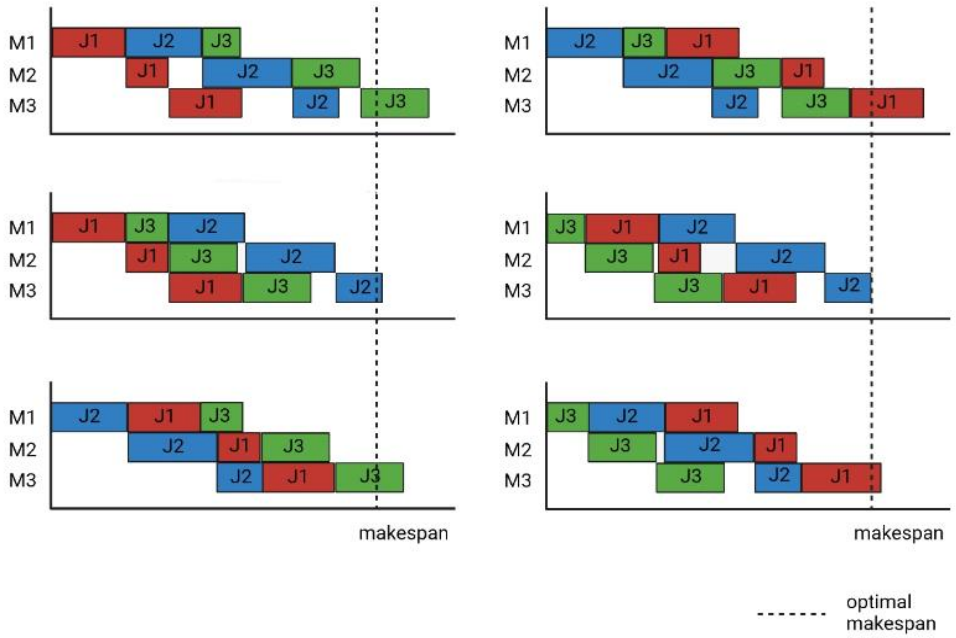
Intra-node Work Stealing Strategy
based on OpenMP and atomics



EXPERIMENTAL SETTINGS AND RESULTS

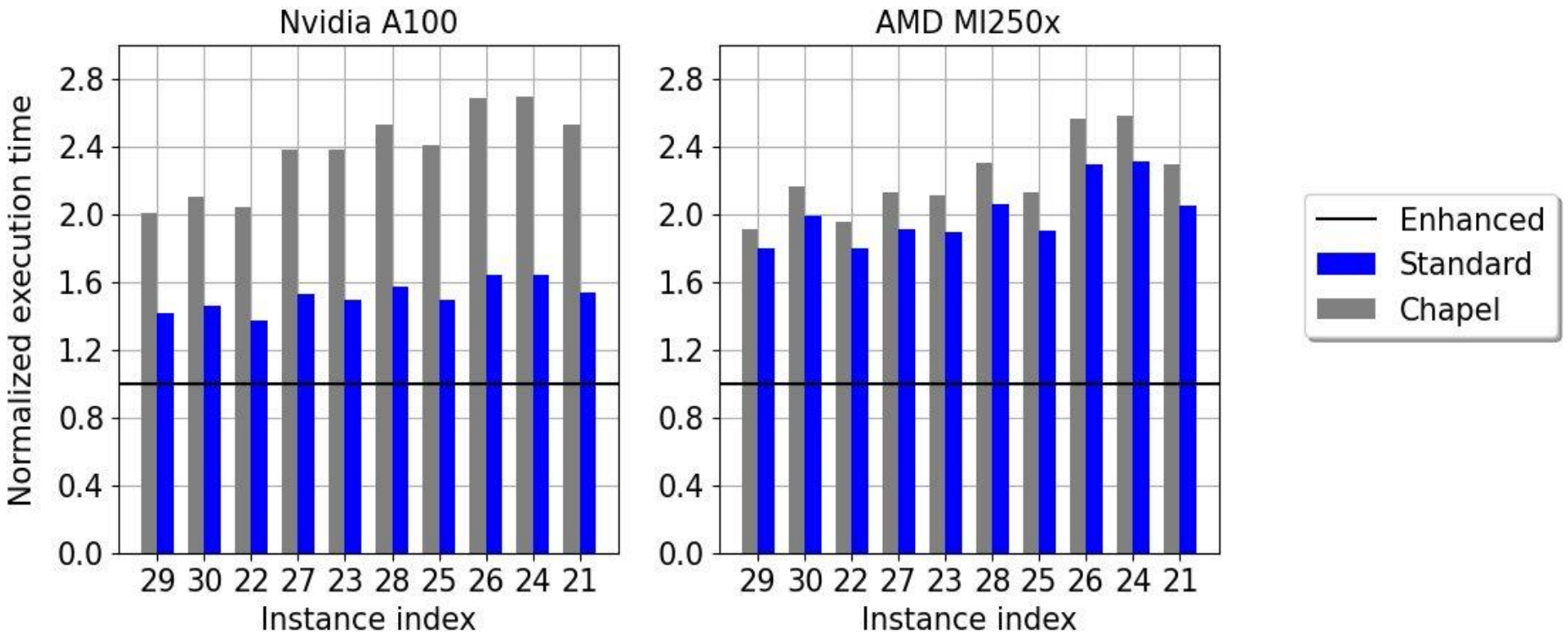
EXPERIMENTAL SETTINGS

- **LUMI supercomputer** (ranked 9th in the TOP500): AMD MI250x GPUs
- **Grid'5000** testbed: Nvidia A100-SXM4-40GB GPUs



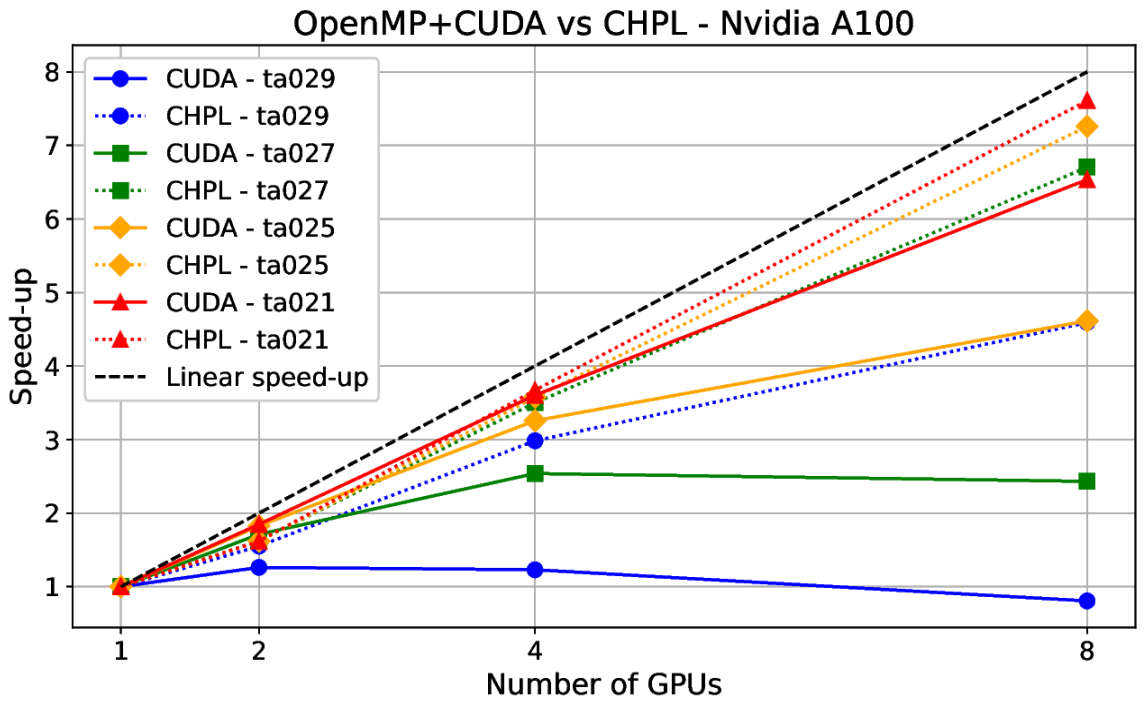
- Permutation Flowshop Scheduling Problem [Garey1976]
  - Highly irregular
  - Big search space: **n!**
- Taillard's 20 × 20 instances, *i.e.*, ta021 to ta030 [Taillard1993]
  - ta29, **ta30**, **ta22**, ta27, **ta23**, **ta28**, ta25, **ta26**, **ta24**, ta21
- "Two-machine bound" B&B lower bound function (**LB2**) [Lageweg1978]
- Initial B&B upper bound = **best optimal solution known**
- Implementations tested:
  - Single-GPU (**SG-B&B**): CUDA, HIP, Chapel
  - Multi-GPU (**MG-B&B**): CUDA, HIP, Chapel

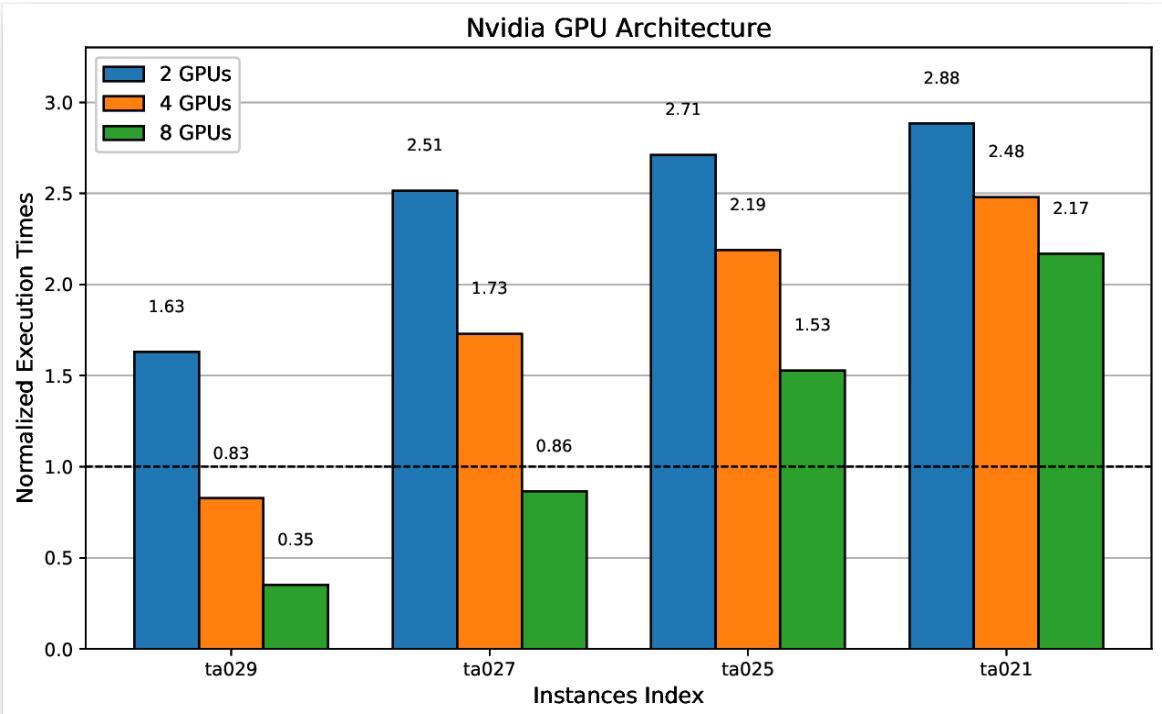
SG-B&B *VS.* CHAPEL SG-B&B



Lower execution times in comparison to Chapel baselines

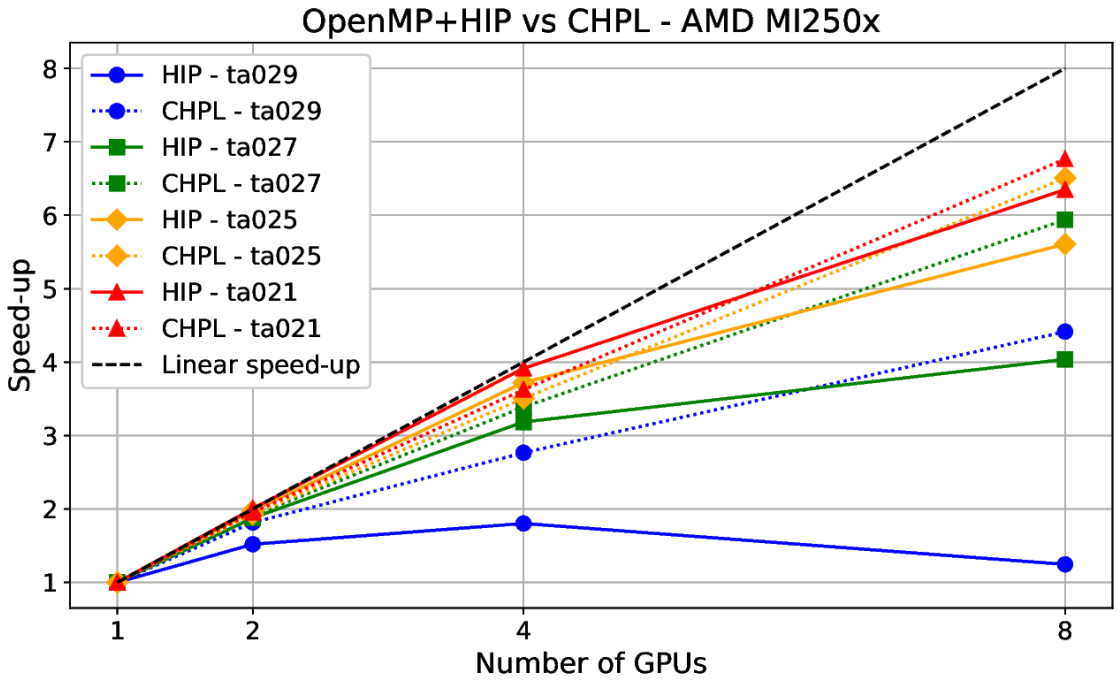
MG-B&B *VS.* CHAPEL MG-B&B NVIDIA

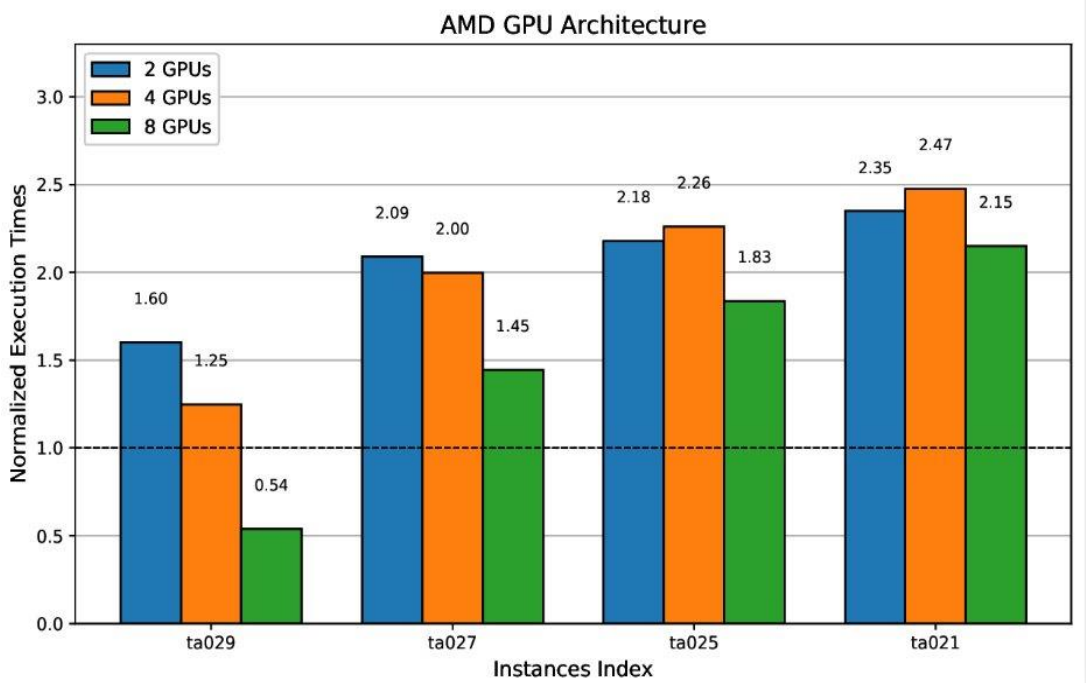




Good strong scalability for bigger instances when computing the relative speedup in relation to CUDA / Chapel SG-B&B implementation

MG-B&B *VS.* CHAPEL MG-B&B AMD

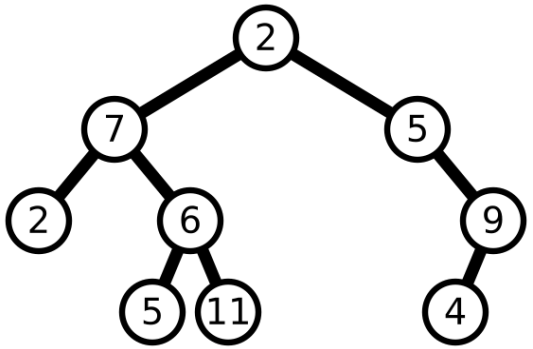




Good strong scalability for bigger instances when computing the relative speedup in relation to HIP / Chapel SG-B&B implementation

IN CONCLUSION...

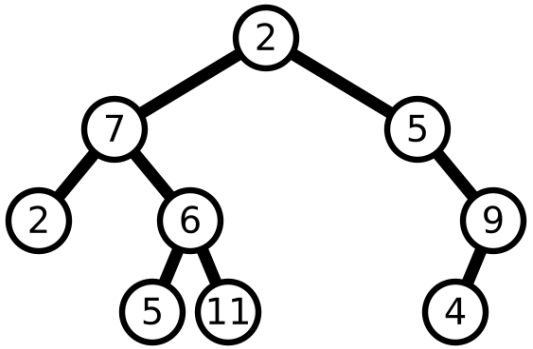
- Single-GPU:
  - Efficient performance and code portable GPU thread Indexing
    - And for Chapel? 30% Speedup!
  - More performant when compared to Chapel baselines
- Multi-GPU:
  - Efficient implementation of the 'atomic' multi-pool approach
  - Good strong scalability
    - Preliminary Results for the inter-node settings



FUTURE WORKS

FUTURE WORKS

- Improve intra-node implementation:
  - Other performance / code portable GPU optimizations
  - Memory contention strategies for **generic** pool-based B&B
  - One-sided shared-memory MPI *vs.* OpenMP
- MPI layer for inter-node distributed implementation
  - Distributed dynamic load balancing
- (More) Feedback on MPI+X *vs.* PGAS and solving hard pending benchmarks (*i.e.*, PFSP)



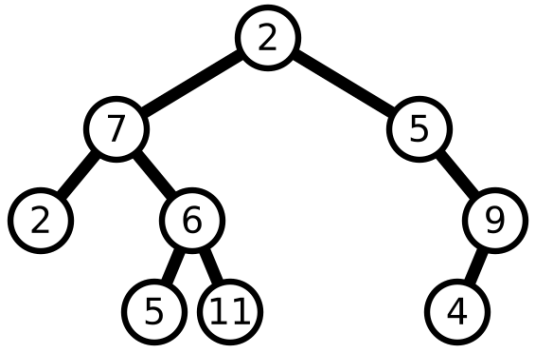
REFERENCES

REFERENCES

- [Top500] **Top500 Project.** *Top500 International Ranking*, https://www.top500.org/.
- [UltraBO] **UltraBO Project**, https://sites.google.com/view/ultrabo/.
- [Garey1976] **M. R. Garey, D. S. Johnson, and R. Sethi**. *The Complexity of Flowshop and Jobshop Scheduling*. Math. Oper. Res., 1:117–129, 1976. https://doi.org/10.1287/moor.1.2.117
- [Lageweg1978] **B. J. Lageweg, J. K. Lenstra, and A. H. G. R. Kan**. *A General Bounding Scheme for the Permutation Flow-Shop Problem*. Operations Research, 26(1):53–67, 1978. https://doi.org/10.1287/opre.26.1.53
- [Taillard1993] **E. Taillard**. *Benchmarks for basic scheduling problems*. European Journal of Operational Research, 64(2):278–285, 1993. Project Management and Scheduling. https://doi.org/10.1016/0377-2217(93)90182-M
- [Gendron1994] **B. Gendron and T. G. Crainic.** *Parallel Branch-And-Bound Algorithms: Survey and Synthesis*. Operations Research, vol. 42, no. 6, pp. 1042–1066, 1994. https://doi.org/10.1287/opre.42.6.1042
- [Chakroun2013] **I. Chakroun, N. Melab, M. Mezmaz, and D. Tuyttens.** *Combining multi-core and GPU computing for solving combinatorial optimization problems.* Journal of Parallel and Distributed Computing, vol. 73, no. 12, pp. 1563–1577, 2013. https://doi.org/10.1016/j.jpdc.2013.07.023
- [Vu2016] **T.-T. Vu and B. Derbel.** *Parallel Branch-and-Bound in multi-core multi-CPU multi-GPU heterogeneous environments*. Future Generation Computer Systems, vol. 56, pp. 95–109, 2016. https://doi.org/10.1016/j.future.2015.10.009
- [Gmys2017] **Gmys J., Mezmaz M., Melab N., Tuyttens D**. I*VM-based parallel branch-and-bound using hierarchical work stealing on multi-GPU systems.* Concurrency and Computation: Practice and Experience, 29(9): e4019, 2017. https://doi.org/10.1002/cpe.4019

REFERENCES

- [Chamberlain2018] B. Chamberlain, E. Ronaghan, B. Albrecht, L. Duncan, M. Ferguson, B. Harshbarger, D. Iten, D. Keaton, V. Litvinov, P. Sahabu, , and G. Titus. *Chapel Comes of Age : Making Scalable Programming Productive*. Proceedings of CUG 2018, 2018. Available at: https://chapel-lang.org/publications/cug2018-chapel.pdf
- [Carneiro2021] T. Carneiro, N. Melab, A. Hayashi, and V. Sarkar. *Towards Chapel-based Exascale Tree Search Algorithms: dealing with multiple GPU accelerators*. Proceedings of HPCS 2020 - The 18th International Conference on High Performance Computing & Simulation, Barcelona / Virtual, Spain, March 2021. https://hal.science/hal-03149394
- [Gmys2022] J. Gmys. *Exactly Solving Hard Permutation Flowshop Scheduling Problems on Peta-Scale GPU-Accelerated Supercomputers*. INFORMS Journal on Computing, vol. 34, pp. 2502–2522, 9 2022. https://doi.org/10.1287/ijoc.2022.1193
- [Helbecque2023] G. Helbecque, J. Gmys, N. Melab, T. Carneiro, P. Bouvry. *Parallel distributed productivity-aware tree-search using Chapel*. Concurrency Computation Practice Experience, 35(27):e7874, 2023. https://doi.org/10.1002/cpe.7874
- [Chapel2024] J. Milthorpe, X. Wang, and A. Azizi. *Performance portability of the chapel language on heterogeneous architectures*. 2024 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), 2024, pp. 6–13. https://doi.org/10.1109/IPDPSW63119.2024.00011
- [Helbecque2024] G. Helbecque, E. Krishnasamy, T. Carneiro, N. Melab, and P. Bouvry. *A Chapel-based Multi-GPU Branch-and-Bound Algorithm*. Proceedings of Euro-Par Workshops, Madrid, Spain, Aug 2024. https://doi.org/10.1007/978-3-031-90200-0_37
- [Helbecque2025] G. Helbecque, E. Krishnasamy, N. Melab, and P. Bouvry. *GPU-Accelerated Tree-Search in Chapel Versus CUDA and HIP*. 2024 IEEE International Parallel and Distributed Processing Symposium Workshops, 05 2024, pp. 872–879. https://doi.org/10.1109/IPDPSW63119.2024.00156



QUESTIONS?

Acknowledgments: This work is supported by the Agence Nationale de la Recherche (ref. ANR-22-CE46-0011) and the Luxembourg National Research Fund (FNR) (ref. INTER /ANR/22/17133848), under the UltraBO project; and by the FNR POLLUX program under the SERENITY project (ref. C22/IS/17395419). We also acknowledge the EuroHPC Joint Undertaking for granting access to the EuroHPC supercomputer LUMI, hosted by CSC (Finland) and the LUMI consortium, through a EuroHPC Regular Access call. Some experiments were conducted using the Grid'5000 experimental testbed, developed under the INRIA ALADDIN development action with support from CNRS, RENATER, and several universities, as well as other funding bodies.

Data Availability Statement: All code written in support of this publication is publicly available at https://doi.org/10.5281/zenodo.15828954 and https://github.com/Guillaume-Helbecque/GPU-accelerated-tree-search-Chapel