Investigating Portability in Chapel for Tree-based Optimization on GPU-powered Clusters

Tiago Carneiro[1], Engin Kayraklioglu[2], Guillaume Helbecque[3] and Nouredine Melab[4]

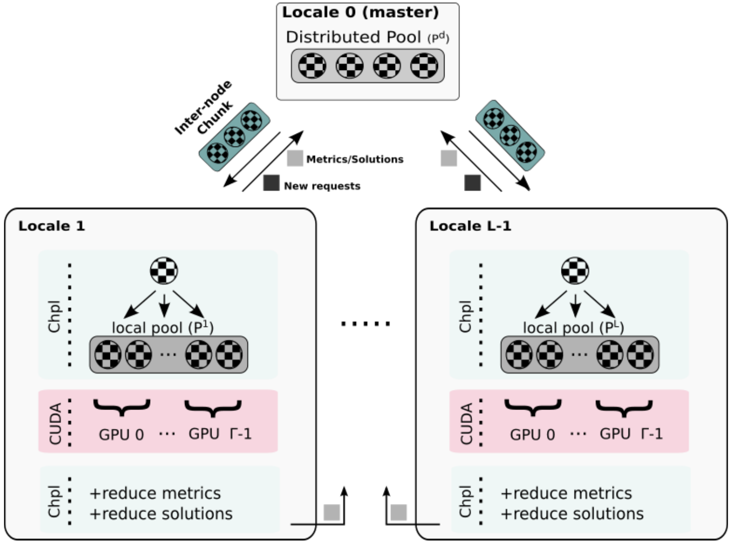
IMEC Leuven – Belgium[1], Hewlett Packard Enterprise - USA[2], University of Luxembourg[3], University of Lille - France[4]

Chapel-Con 2024
June 5–7, 2024

Chapel-based Optimization (ChOp)

A Distributed Tree-based Search in Chapel

- Chapel-based Optimization (ChOp)
  - The `DistributedIters` module, encapsulates a complex distributed master-worker load balancing scheme.

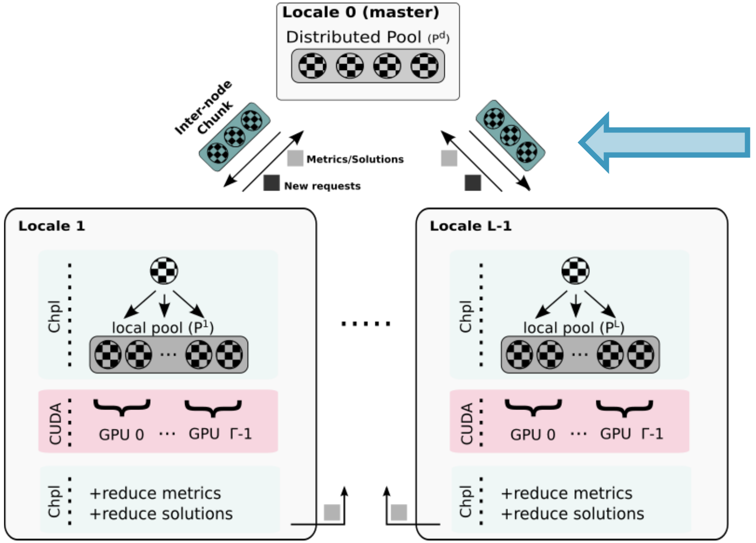


Chapel-based Optimization (ChOp)

A Distributed Tree-based Search in Chapel

- Chapel-based Optimization (ChOp)
  - The `DistributedIters` module, encapsulates a complex distributed master-worker load balancing scheme.

Branch-and-Bound/Backtracking
To solve big COP



DistributedIters:
distributed load balancing/work distribution
Metrics reduction
Termination criteria

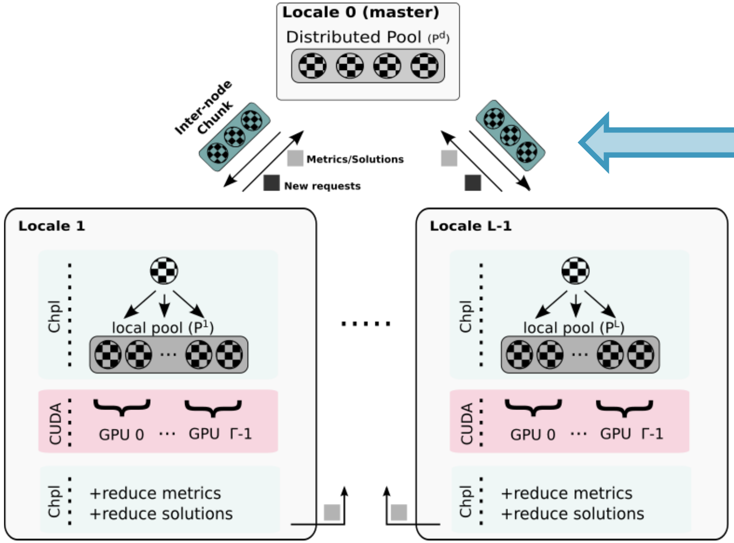
imec

Chapel-based Optimization (ChOp)

A Distributed Tree-based Search in Chapel

- Chapel-based Optimization (ChOp)
  - The `DistributedIters` module, encapsulates a complex distributed master-worker load balancing scheme.

Branch-and-Bound/Backtracking
To solve big COP



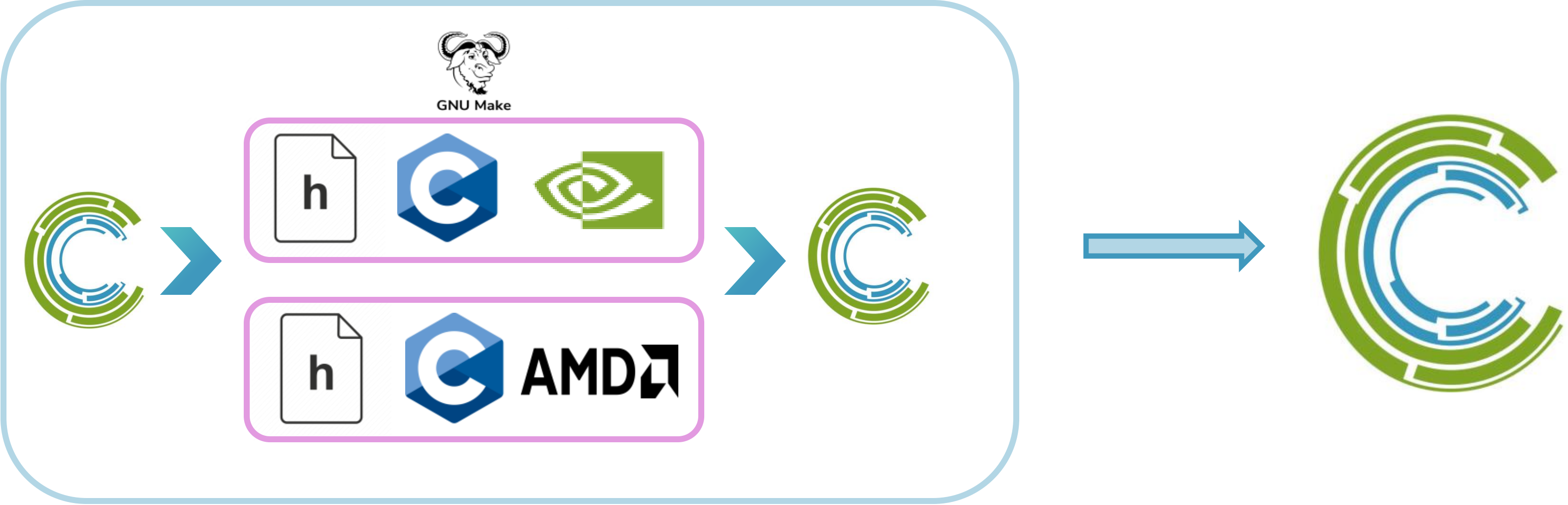
DistributedIters:
distributed load balancing/work distribution
Metrics reduction
Termination criteria

Intra-node:
Chapel+CUDA

The Complexity of Using C-Interoperability

Redundant and Difficult to Maintain

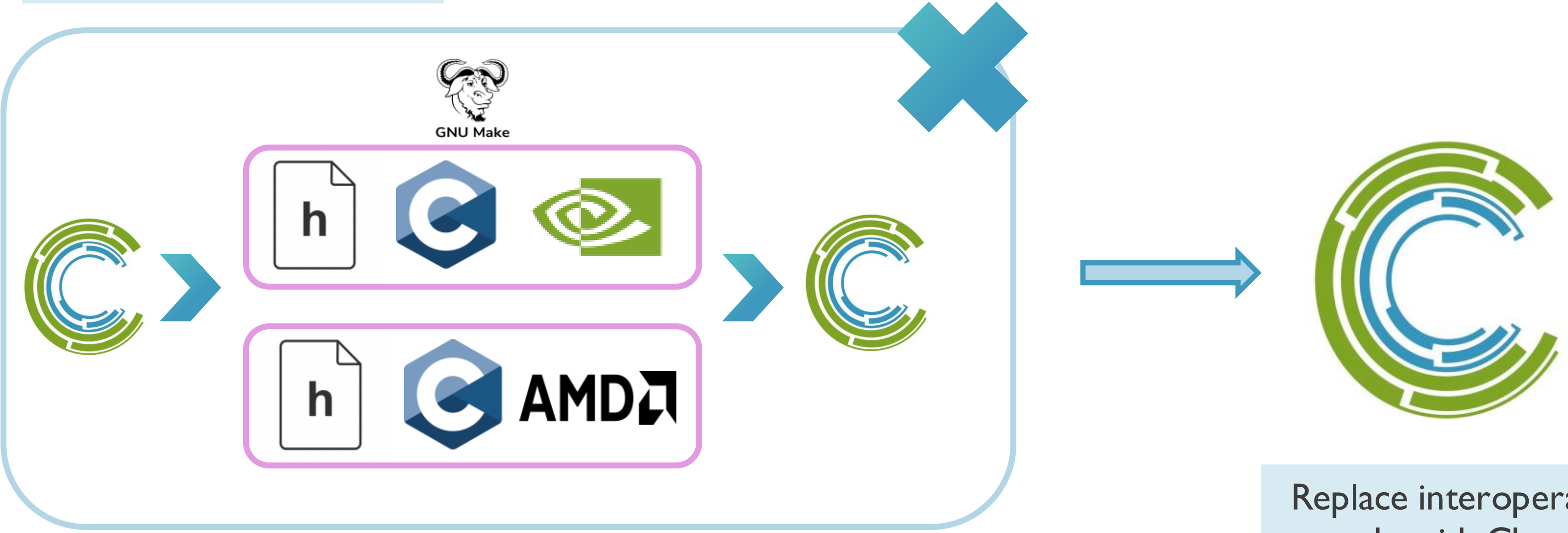
Intra-node Parallelism



The Complexity of Using C-Interoperability

Redundant and Difficult to Maintain

Intra-node Parallelism



Replace interoperability code with Chapel's native GPU support

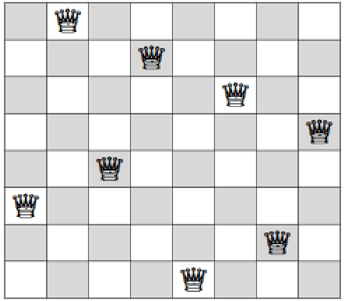
Objectives

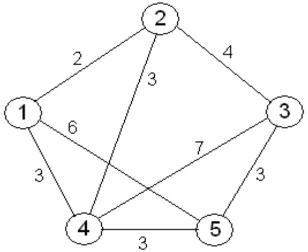
1. **Objective:**
   1. to verify whether it is possible to achieve both code portability and performance portability in distributed exact optimization with Chapel's native GPU support.
      - Is it worth in terms of programming effort?

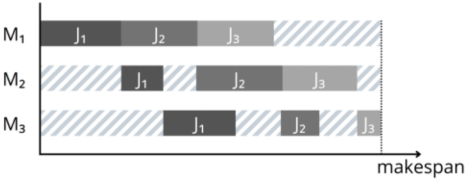
Performance Experiments

Protocol

- The following applications for enumerating all valid complete solutions of the N-Queens problem are considered: (Backtracking)
    - **Single-node Multi-GPU *baseline*:** HIP+OpenMP and CUDA+OpenMP
    - **Single-node Multi-GPU:** Chapel-GPU
    - **Distributed Hybrid:** Chapel+CUDA/Chapel+HIP
    - **Distributed Chapel-GPU:** Chape's native GPU support



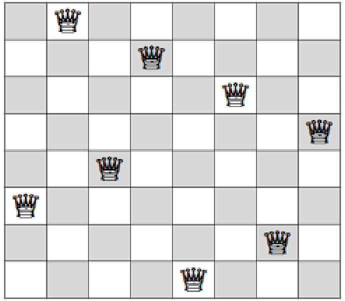




Performance Experiments

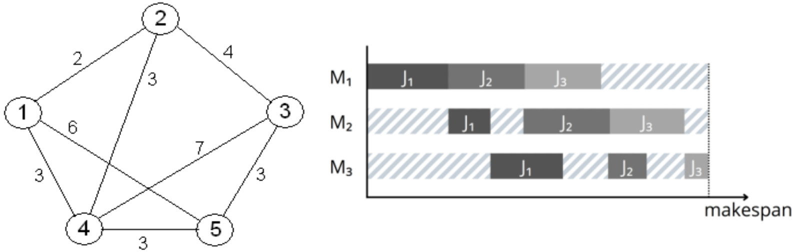
Protocol

- The following applications for enumerating all valid complete solutions of the N-Queens problem are considered: (Backtracking)
  - **Single-node Multi-GPU *baseline*:** HIP+OpenMP and CUDA+OpenMP
  - **Single-node Multi-GPU:** Chapel-GPU
  - **Distributed Hybrid:** Chapel+CUDA/Chapel+HIP
  - **Distributed Chapel-GPU:** Chape's native GPU support



N-Queens – proof of concept

C. Optimization Problems – B&B



Performance Experiments

Testbeds and Parameters

- **N-Queens:**
  - Queens of sizes ranging from N=18 to N=22
- **Computers:**
  - NVIDIA System:
    - Perlmutter supercomputer (#12 of TOP500)
    - 1 to 128 computer nodes
    - 4 to 512 GPUs (A100 SXM4)
  - AMD System:
    - Frontier supercomputer (#1 of TOP500)
    - 1 to 128 computer nodes
    - 8 to 1024 GPUs (MI250As)
    - (only two hours of execution)

Performance Experiments

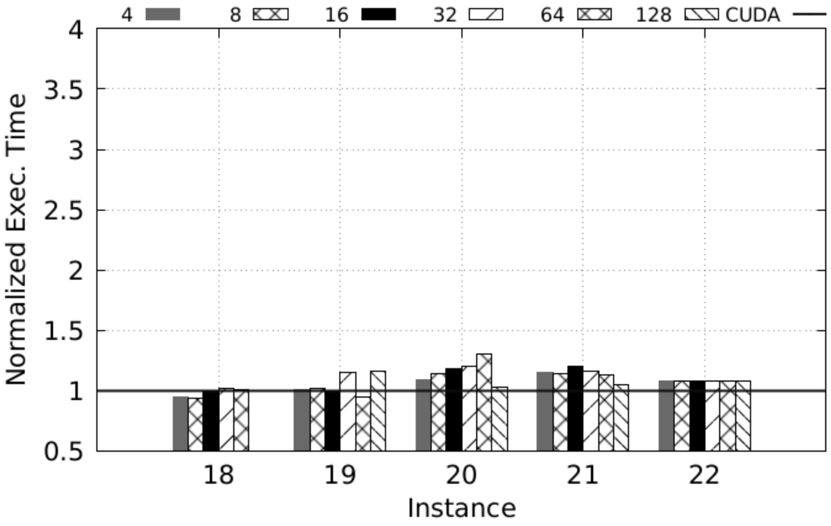
Testbeds and Parameters

- **N-Queens:**
  - Queens of sizes ranging from N=18 to N=22
- **Computers:**
  - NVIDIA System:
    - Perlmutter supercomputer (#12 of TOP500)
    - 1 to 128 computer nodes
    - 4 to 512 GPUs (A100 SXM4)
  - AMD System:
    - Frontier supercomputer (#1 of TOP500)
    - 1 to 128 computer nodes
    - 8 to 1024 GPUs (MI250As)
    - **(only two hours of execution)**

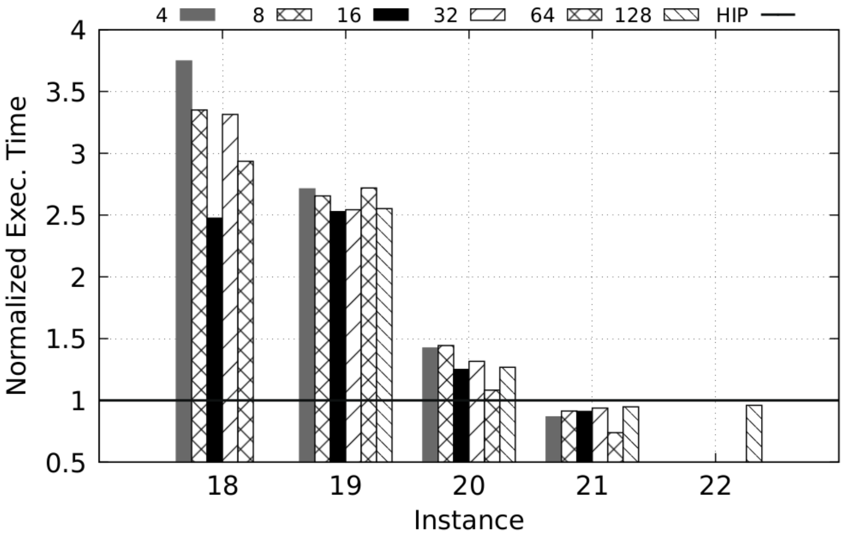
Performance Results

Distributed Application

- N-Queens of sizes ranging from N=18 to N=22
- 4 to 128 computer nodes, Chapel's native GPU support *vs.* Hybrid



(a) vs. CUDA - NVIDIA-based System

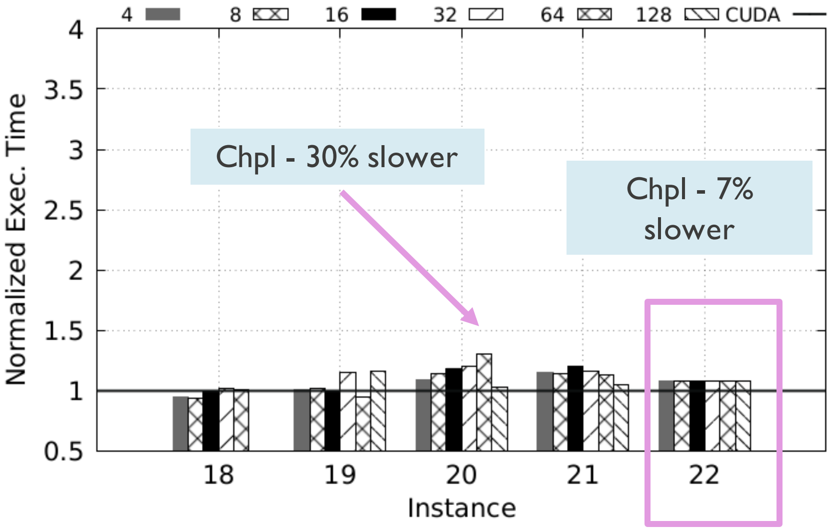


(b) vs. HIP - AMD-based System

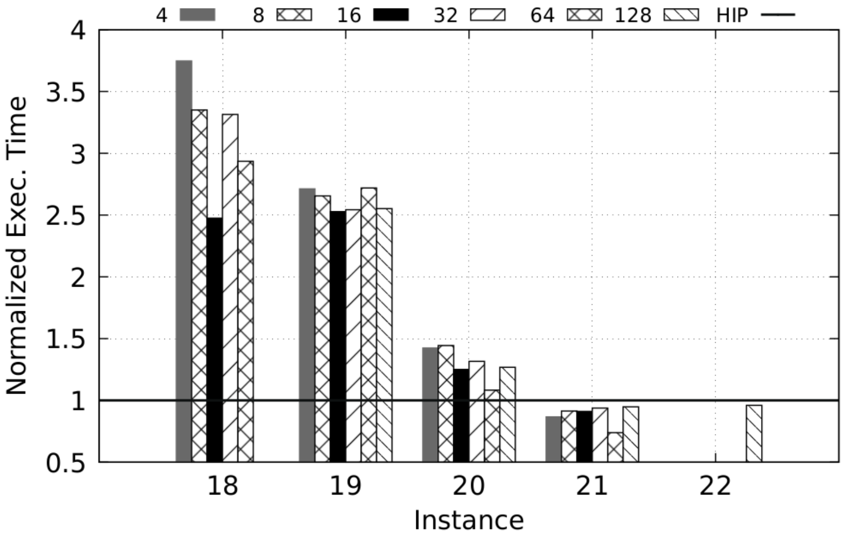
Performance Results

Distributed Application

- N-Queens of sizes ranging from N=18 to N=22
- 4 to 128 computer nodes, Chapel's native GPU support *vs.* Hybrid



(a) vs. CUDA - NVIDIA-based System

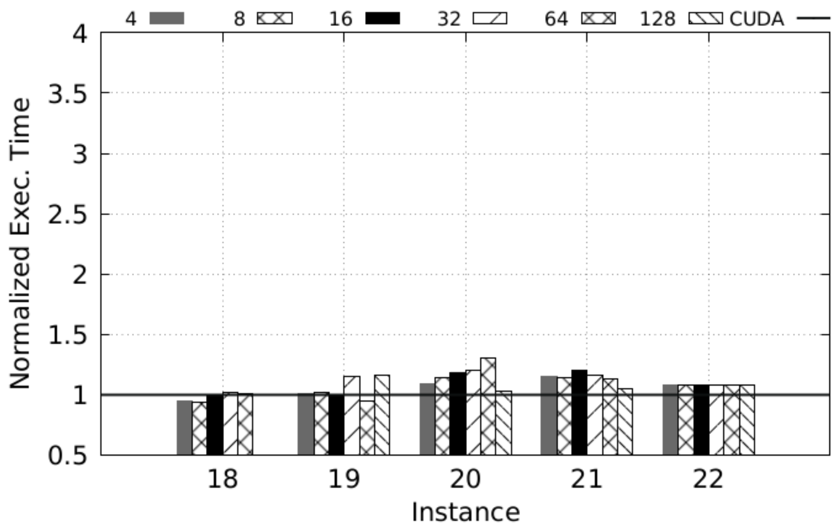


(b) vs. HIP - AMD-based System

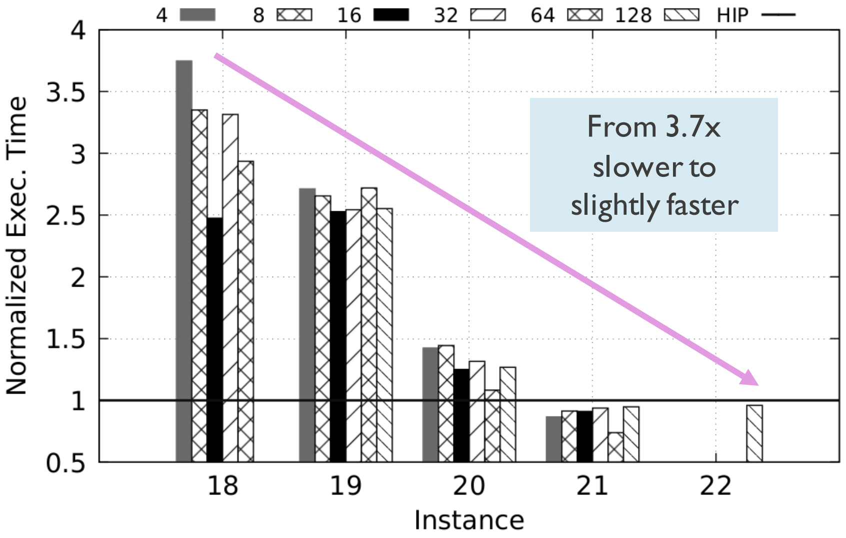
Performance Results

Distributed Application

- N-Queens of sizes ranging from N=18 to N=22
- 4 to 128 computer nodes, Chapel's native GPU support *vs.* Hybrid



(a) vs. CUDA - NVIDIA-based System

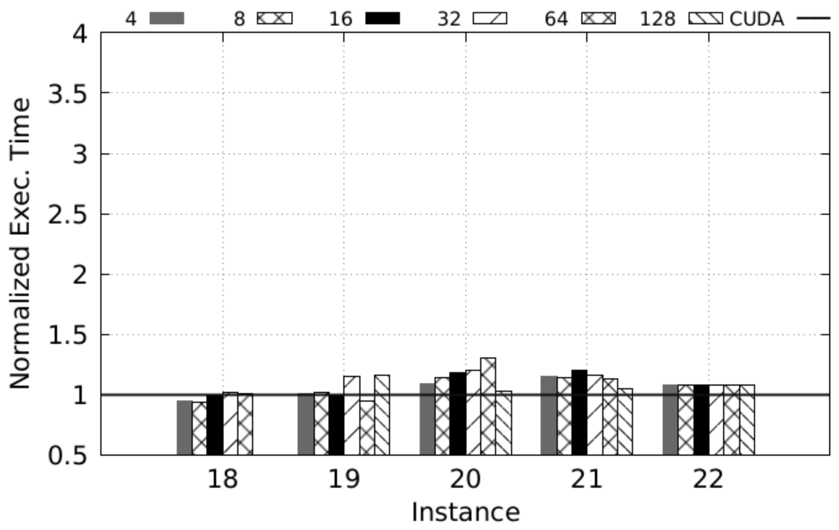


(b) vs. HIP - AMD-based System

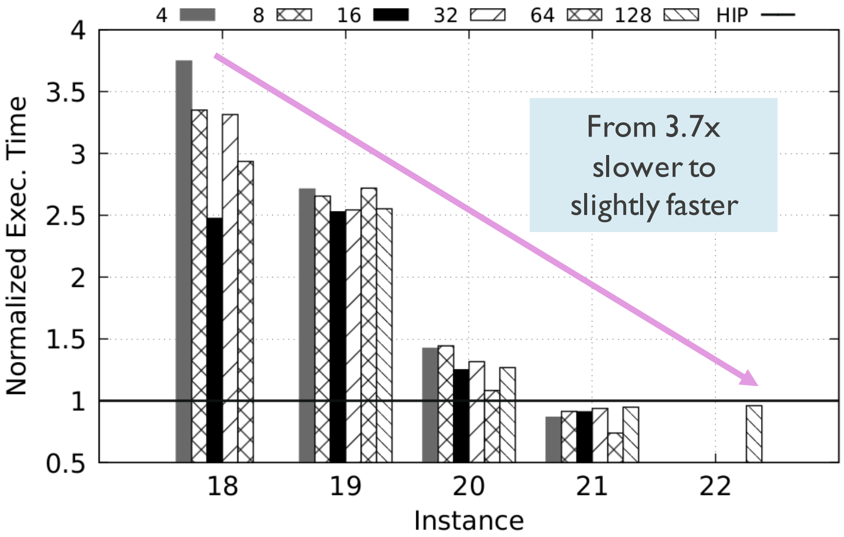
Performance Results

Distributed Application

- N-Queens of sizes ranging from N=18 to N=22
- 4 to 128 computer nodes, Chapel's native GPU support *vs.* Hybrid



(a) vs. CUDA - NVIDIA-based System

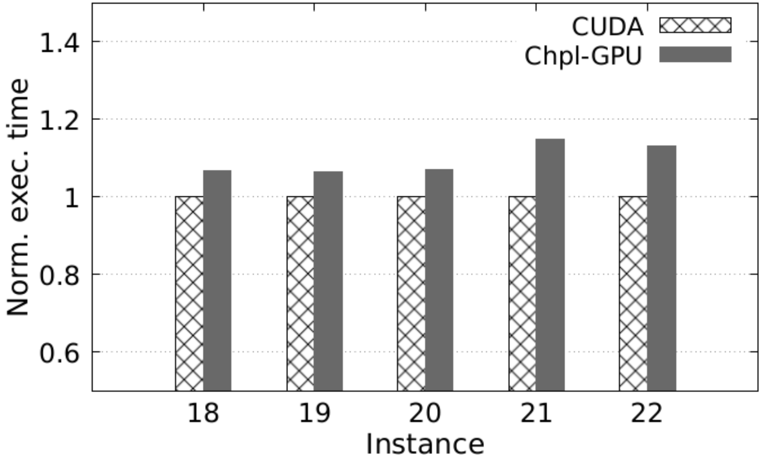


(b) vs. HIP - AMD-based System

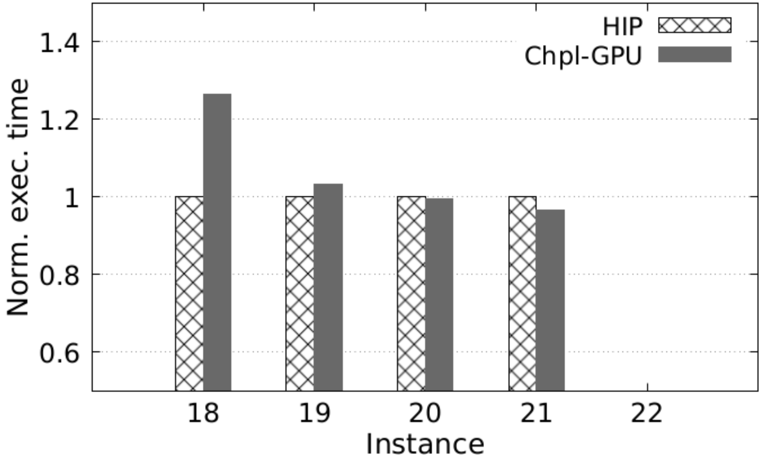
Performance Results

Single-node Application

- N-Queens of sizes ranging from N=18 to N=22
- 4 GPUS – NVIDIA, 8 GPUs – AMD, Chapel's native GPU support *vs.* Baseline



(a) vs. CUDA - NVIDIA-based System



(b) vs. HIP - AMD-based System

imec

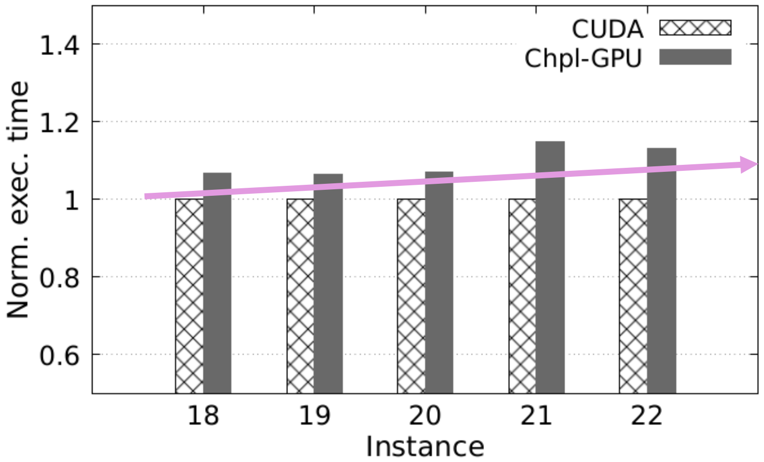
Performance Results

Single-node Application

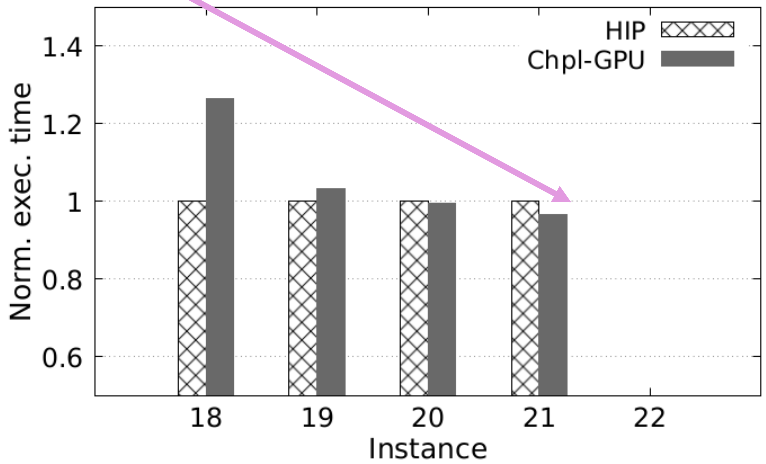
- N-Queens of sizes ranging from N=18 to N=22
- 4 GPUS – NVIDIA, 8 GPUs - AMD

Similar behavior on single-node

The lower the load - the worse the Chapel performance



(a) vs. CUDA - NVIDIA-based System

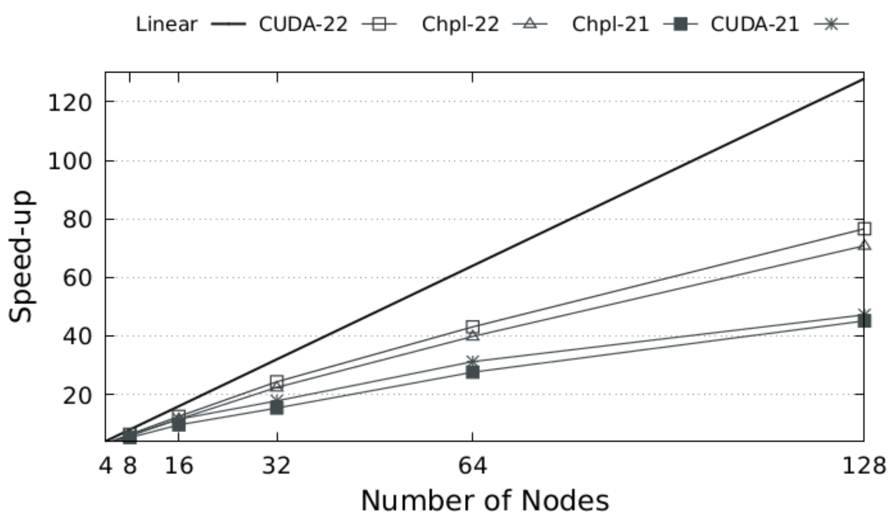


(b) vs. HIP - AMD-based System

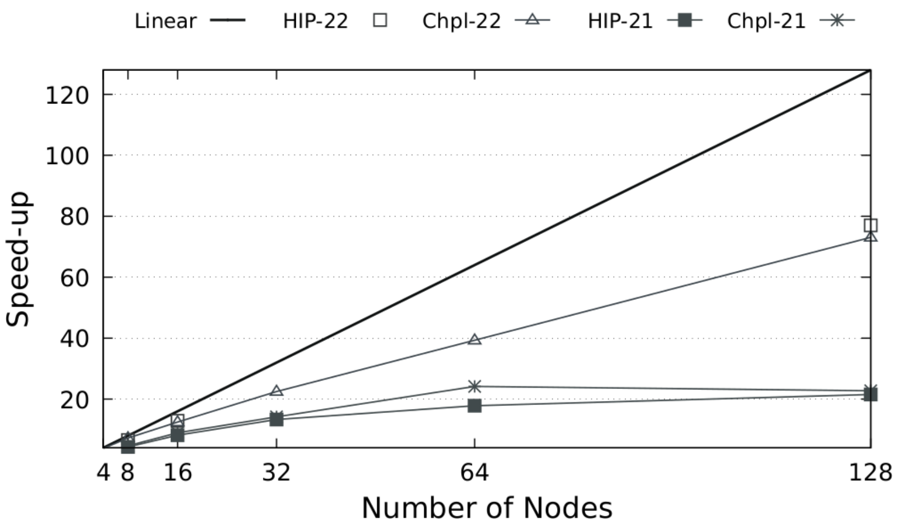
Performance Results

Strong Scaling

- N-Queens of sizes ranging from N=18 to N=22
- 4-128 computer nodes, speedup compared to the Baseline



(a) NVIDIA-based System



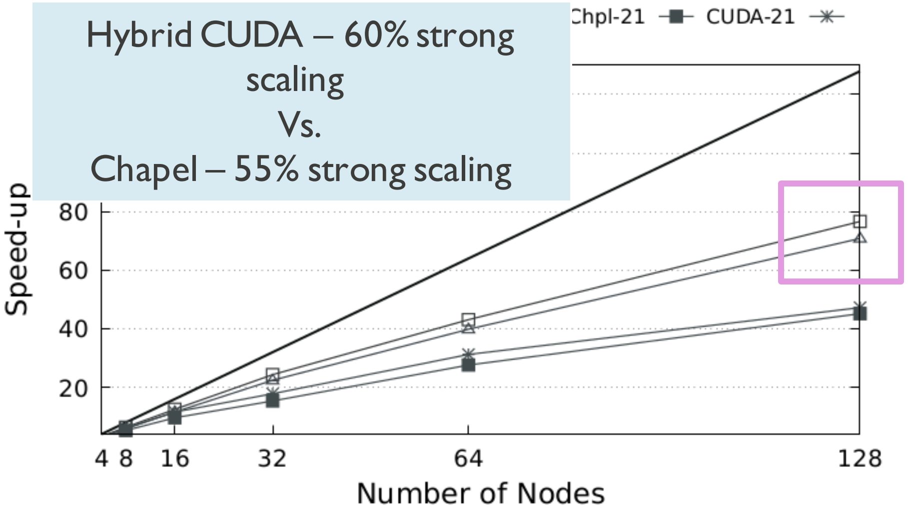
(b) AMD-based system

Performance Results

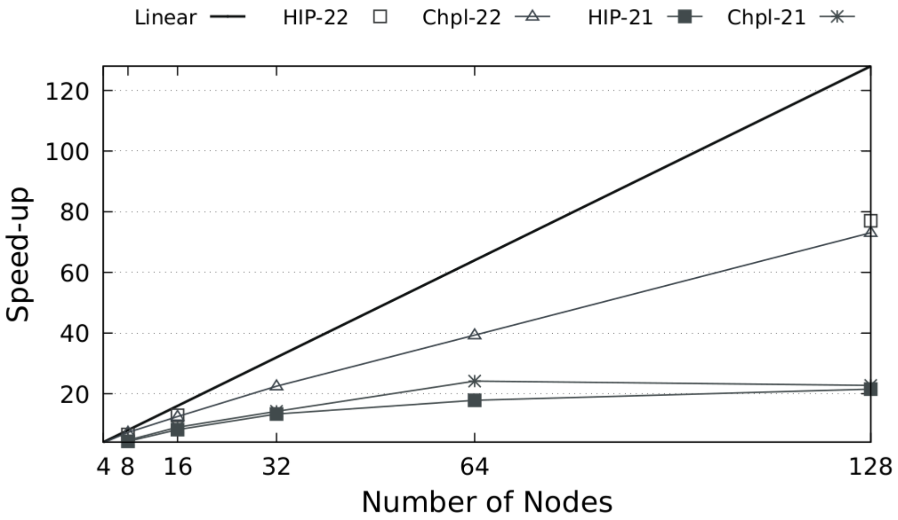
Strong Scaling

- N-Queens of sizes ranging from N=18 to N=22
- 4-128 computer nodes, speedup compared to the Baseline

Hybrid CUDA – 60% strong scaling
Vs.
Chapel – 55% strong scaling



(a) NVIDIA-based System



(b) AMD-based system

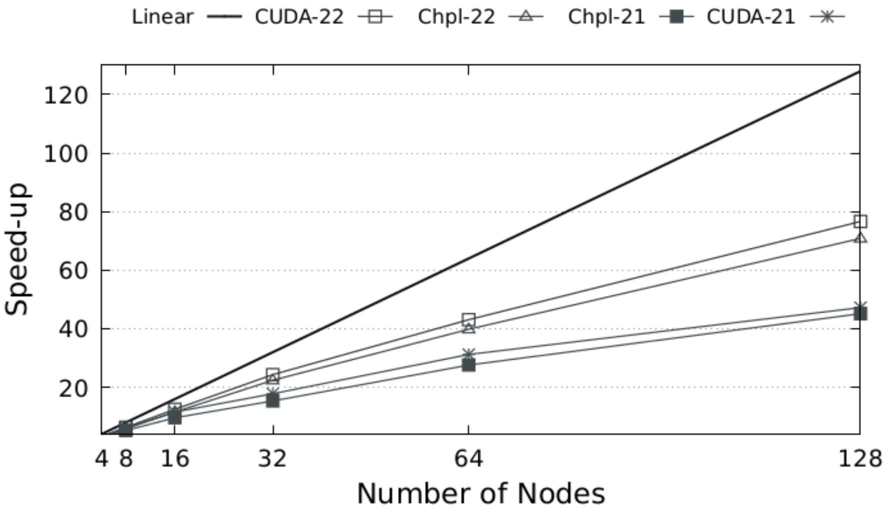
imec

Performance Results

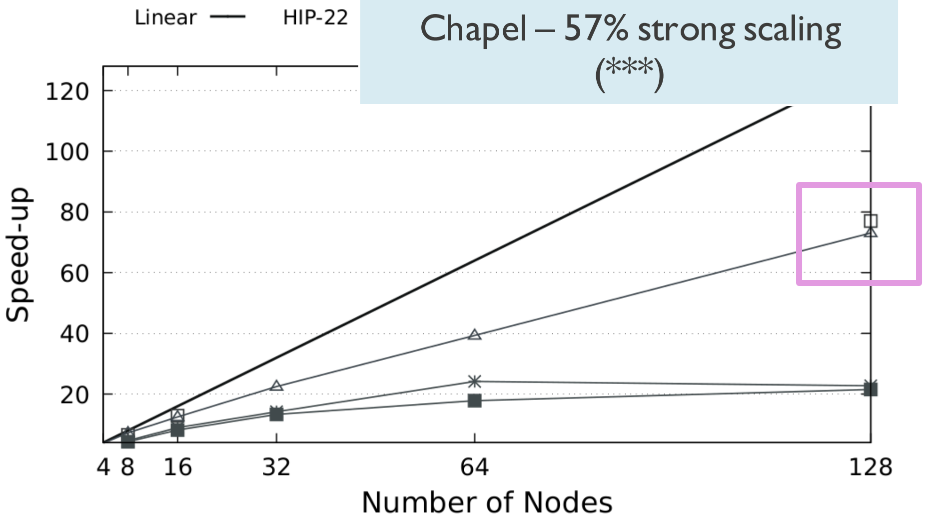
Strong Scaling

- N-Queens of sizes ranging from N=18 to N=22
- 4-128 computer nodes, speedup compared to the Baseline

Hybrid HIP – 61% strong scaling
Vs.
Chapel – 57% strong scaling
(***)



(a) NVIDIA-based System



(b) AMD-based system

Programming Effort

Single-node – Multi-GPU

- The HIP, CUDA and Chapel-GPU applications are equivalent in terms of SLOC
  - Single-node – Multi-GPU
  - One thread per GPU
  - N-Queens kernel
- The benefit is to get rid of both HIP and CUDA applications
  - Low/no performance loss for the biggest loads

Programming Effort

Distributed

- Replacing one interoperability code (CUDA or HIP) with Chapel's native GPU support results in an application 30% shorter.
  - Getting rid of both interoperability codes (CUDA and HIP) results in a final code 65% shorter.
  - Almost no performance loss in distributed execution - biggest loads

Conclusion

- It is possible to use Chapel to program all levels of parallelism of a large-scale cluster.
  - Both the Chapel-GPU implementation and its hybrid counterpart achieved similar strong scaling efficiency on 128 nodes.
  - Using Chapel's Native GPU support instead of interoperability results in a distributed application 65% shorter.
- It is possible to achieve both code portability and performance portability in distributed exact optimization with Chapel's native GPU support.

- ** Results are from a paper of the same title that will be presented in **Europar 2024**

Acknowledgments

- This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic through the e-INFRA CZ (ID:90254) -- Project EU2022D08-197.
- It is also supported by the Agence Nationale de la Recherche (ref.ANR-22-CE46-0011) and the Luxembourg National Research Fund (ref. INTER/ANR/22/ 17133848), under the UltraBO project.
- This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility located at Lawrence Berkeley National Laboratory, operated under Contract No. DE-AC02-05CH11231 using NERSC award ASCR-ERCAP-mp215. In addition, this research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

Thank you!

Questions?

https://github.com/tcarneirop/ChOp

Chapel-based Optimization on Github

tcarneiropessoa@gmail.com