Unlocking Performance Portability on LUMI-G Supercomputer: A Virtual Screening Case Study

Gianmarco Accordi, Davide Gadioli, Gianluca Palermo, Luigi Crisci, Lorenzo Carpentieri, Biagio Cosenza, Andrea R. Beccari
Overview

1. Drug Discovery and Virtual Screening
2. HPC for Urgent Computing
3. LiGen Batched GPU Acceleration
4. SYCL Porting
5. LUMI Benchmark Access
6. Results and Conclusions
Drug Discovery

- Identify chemicals that yield potential therapeutic effects
- It is a very long and costly process
  - Due to failure while finding drug candidates
Drug Discovery

- Identify chemicals that yield potential therapeutic effects
- It is a very long and costly process
  - Due to failure while finding drug candidates
Virtual Screening

● It's an early stage of the drug discovery process
● It screens a large database of known compounds
  ○ Looking for the most promising drug candidates
● *In-silico* filter of compounds
  ○ Generate feasible compounds poses
  ○ Evaluate each pose-protein interaction strength
Molecular Docking

- Structural-based virtual screening
- Given two molecules, it searches for feasible ligands poses
  - It binds ligands onto a target protein
Scoring Functions

● Predicts the interaction strength of each pose-protein pair
  ○ The output **SCORE** is a numerical value
  ○ It is used for ranking

● Consider different chemical interactions
  ○ Hydrogen bonding
  ○ Solvent
  ○ Buried surface
  ○ Van der Waals forces

\[ f \{ \text{protein}, \text{ligand} \} = \text{SCORE} \]
Overview

1. Drug Discovery and Virtual Screening
2. HPC for Urgent Computing
3. LiGen Batched GPU Acceleration
4. SYCL Porting
5. LUMI Benchmark Access
6. Results and Conclusions
Virtual Screening and HPC

- Virtual screening is complex
  - Virtual compound libraries are very large
- Each ligand-protein evaluation is independent
  - Embarrassing parallel problem
- Supercomputers are leveraged to perform virtual screening campaign
LiGen

- Virtual screening application owned by Dompé
- It is a component of the EXSCALATE drug discovery platform
- Design to hinge the modern supercomputer nodes
- Used to perform extreme-scale virtual screening campaign

References:
Urgent Computing

- Virtual screening campaign to fight back against pandemics
  - **ANTAREX**<sub>4</sub>ZIKA CPU only version
  - **EXSCALATE**<sub>4</sub>C0V GPU support (CUDA)
- The **LIGATE** European is developing a CADD workflow
  - Support for several EuroHPC supercomputers
  - LUMI deployment showed a new challenge

References:
Overview

1. Drug Discovery and Virtual Screening
2. HPC for Urgent Computing
3. LiGen Batched GPU Acceleration
4. SYCL Porting
5. LUMI Benchmark Access
6. Results and Conclusions
LiGen GPU Approach

- LiGen deploys a highly optimized CUDA version
  - GPU computational approach shifted from a latency to a throughput one
  - Thanks to a collaboration with NVIDIA's engineers

References:
LiGen Batched Approach - 1

- LiGen code makes extensive use of template meta-programming
  - Influence kernel's registers pressure
  - Batch's ligands properties are used to select a kernel implementation

- Batches can be tuned
  - Total number of batches used
  - Each batch dimension
    - Influenced by the hardware characteristics
    - Auto-tuning using CUDA runtime API

References:
G. Accordi et al., “Out of kernel tuning and optimizations for portable large-scale docking experiments on GPUs”, JoS, 2024
LiGen Latency Approach

- LiGen latency version process ligand in-order

References:
G. Accordi et al., “Out of kernel tuning and optimizations for portable large-scale docking experiments on GPUs”
LiGen Batched Approach - 2

- LiGen throughput version process ligand out-of-order
  - It packs ligands with the same expected execution time in batches

References:
G. Accordi et al., “Out of kernel tuning and optimizations for portable large-scale docking experiments on GPUs”, JoS, 2024
API Query & Formulas

CUDA

\[ l = b \times SM \times \frac{t}{ws} \]

- CUDA \( b \) obtained with
  - `cudaOccupancyMaxActiveBlocksPerMultiprocessor`
Overview

1. Drug Discovery and Virtual Screening
2. HPC for Urgent Computing
3. LiGen Batched GPU Acceleration
4. SYCL Porting
5. LUMI Benchmark Access
6. Results and Conclusions
LiGen SYCL porting

- SYCL porting has initially been focused on NVIDIA GPUs
  - We focused on maintaining LiGen functionality end-to-end
- Then, the LiGen SYCL version was extended to run on multi-GPU and multi-node architectures
  - The batched approach has been ported into SYCL

References:
L. Crisci et al., “Enabling Performance Portability on the LiGen Drug Discovery Pipeline”, FGCS, 2024
## API Query & Formulas

<table>
<thead>
<tr>
<th>CUDA</th>
<th>SYCL</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ l = b \times SM \times \frac{t}{ws} ]</td>
<td>[ l = \frac{wgs}{t} \times CU \times \frac{t}{sgs} ]</td>
</tr>
</tbody>
</table>

- **CUDA** \( b \) obtained with
  - `cudaOccupancyMaxActiveBlocksPerMultiprocessor`
- **SYCL** \( wgs \) obtained with
  - `kernel_device_specific::work_group_size`
  - **Part of the** `kernel_bundle` **API**
LiGen Number of Batches

(a) CUDA implementation

(b) SYCL implementation
LiGen Batch Dimension

- Batching showed a similar performance improvement

References:
G. Accordi et al., “Out of kernel tuning and optimizations for portable large-scale docking experiments on GPUs”
Batching Registers Pressure

- Ligands' sizes used as typed template parameter
- Template parameters impact loop unrolling
  - Increased register pressure
- Fix register number does not help
  - Slower kernel

<table>
<thead>
<tr>
<th>Num Atoms</th>
<th>CUDA Registers</th>
<th>SYCL(oneAPI) Registers</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>102</td>
<td>158</td>
</tr>
<tr>
<td>64</td>
<td>103</td>
<td>176</td>
</tr>
<tr>
<td>96</td>
<td>98</td>
<td>176</td>
</tr>
<tr>
<td>128</td>
<td>102</td>
<td>178</td>
</tr>
<tr>
<td>160</td>
<td>112</td>
<td>176</td>
</tr>
<tr>
<td>192</td>
<td>124</td>
<td>182</td>
</tr>
</tbody>
</table>
Overview

1. Drug Discovery and Virtual Screening
2. HPC for Urgent Computing
3. LiGen Batched GPU Acceleration
4. SYCL Porting
5. LUMI Benchmark Access
6. Results and Conclusions
Benchmark Access

- SYCL support has been extended to AMD GPUs
  - Thanks to a Benchmark Access on the LUMI-G partition
- SYCL Porting tested and tuned for AMD architecture
- On AMD GPUs, no LiGen reference is available
  - HIP porting using HIPIFY
  - The HIP version has been tested but not tuned
LUMI

- Located at CSC Data Center in Finland, co-founded by EuroHPC
- It has a speed of 550 petaFLOPS
  - 5th supercomputer in the world, according to November 2023 Top500
  - 1st supercomputer in Europe, according to the same ranking
- Based on HPE Cray EX architecture
- No official SYCL support when we started collecting data
  - We have been in contact with the CSC support team
  - We preferred to compile everything from scratch
  - Some technical problems to get a working SYCL toolchain
Overview

1. Drug Discovery and Virtual Screening
2. HPC for Urgent Computing
3. LiGen Batched GPU Acceleration
4. SYCL Porting
5. LUMI Benchmark Access
6. Results and Conclusions
Experimental Setup

● Software-stack
  ○ GCC 11.3 and LLVM 15.0.6
  ○ AdaptiveCpp 0.9.4
  ○ oneAPI DPC++ 2022-12
  ○ NVCC 11.7
  ○ HIP 5.3

● Hardware
  ○ AMD MI250X on LUMI-G nodes
  ○ AMD MI100 on E4 cluster
  ○ NVIDIA A100 on Karolina nodes
LiGen LUMI GPUs Performance

Throughput [lgs/sec]

- AdaptiveCPP
- oneAPI
- CUDA
- HIP

Comparative chart showing performance of different GPUs.
LiGen Scaling on LUMI

NOTE: due to a technical problem with the FS, we used only 1 GPU per node
Conclusions

● We are now able to support several EuroHPC supercomputers
  ○ Performance portability unlocked

● There is still room for improvement
  ○ On NVIDIA, we cannot go fully SYCL
    ■ High register pressure
  ○ Support now for AMD systems
    ■ SYCL compiler performs differently
    ■ HIP requires some tuning

Urgent computing scenarios can perform future virtual screening campaigns on more supercomputing architectures
Thank you for your attention!

- **Acknowledgment**
  - EuroHPC JU for awarding this project access to
    - LUMI under project EHPC-BEN-2022B12-001
    - Karolina with grant EHPC-DEV-2021D02-049
  - European Union’s Horizon 2020 research and innovation program
    - Under grant agreement No 95613 (LIGATE)

Contact reference:
Gianmarco Accordi
gianmarco.accordi@polimi.it