PDP 2020
28th EuroMicro International Conference on Parallel, Distributed, and Network-Based Processing. March 11-13, 2020

Evaluating the Energy Efficiency of OpenCL-accelerated AutoDock Molecular Docking

Leonardo Solis-Vasquez *, Diogo Santos-Martins +, Andreas Koch *, Stefano Forli +

* Embedded Systems and Applications Group
Technical University of Darmstadt, Germany

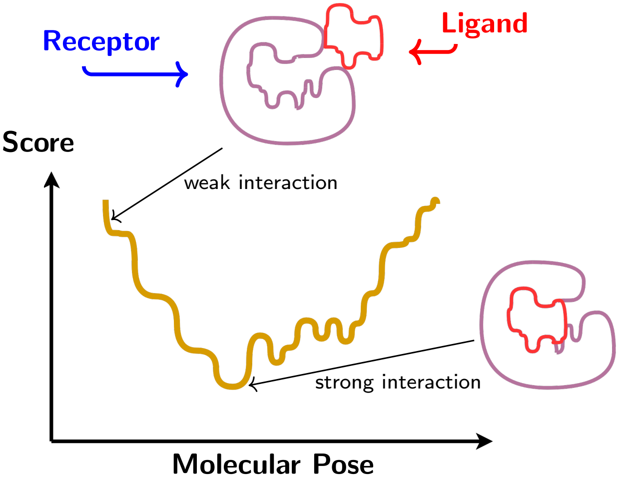
+ Center for Computational Structural Biology
The Scripps Research Institute, United States

Introduction

- Energy efficiency
  - Important in design of computer systems
  - Future systems will be constrained by their power consumption
- Upcoming trend
  - Replacement of homogeneous with heterogeneous accelerators
  - Top eight systems in the *Green500* list use GPUs
- Scientific applications used at scale
  - Can profit from HPC systems
  - Efficient deployment: performance-/energy-wise

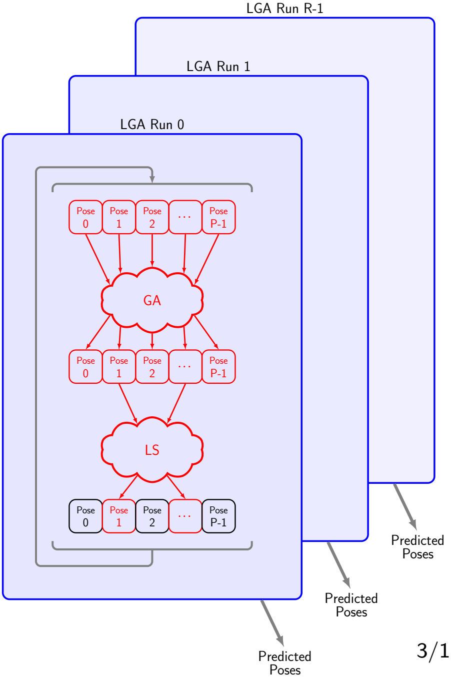
Molecular Docking (MD)

MD aims to find poses of strong interaction



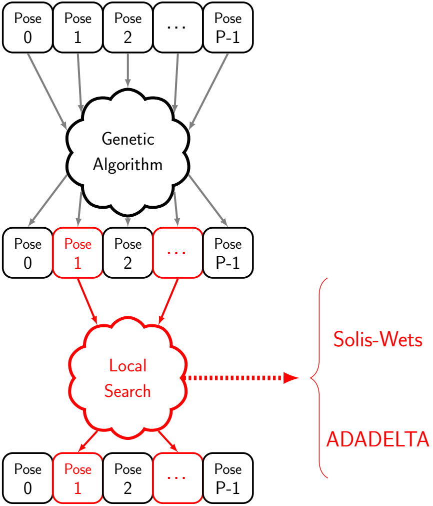
AutoDock

- One of the most cited MD tools
- Lamarckian Genetic Algorithm (LGA)
- LGA = GA + LS
  - Genetic Algorithm (GA)
  - Local Search (LS)



Our previous work on AutoDock

- AutoDock-OpenCL
  - Data-parallel approach for GPUs/CPUs
  - Fine-grained & multi-level parallelization
- Enhanced search
  - More accurate pose prediction
  - Available LS methods
    - Solis-Wets (legacy)
    - ADADELTA (new, gradient-based)



Our Contribution

Energy efficiency analysis of OpenCL-accelerated AutoDock

1. Correlation between performance and power consumption
   - Solis-Wets vs. ADADELTA

2. Impact of molecular complexity on
   - Execution performance
   - Energy efficiency
   - Multi-core CPUs & many-core GPUs

Execution profiling: Vega 56 GPU

- OpenCL configuration
  - 56 compute units
  - 64 work-items / work-group
- MD setup
  - # LGA runs: $R = 100$
  - Input ID: 3s8o
- Profiling is focused on the local-search kernel (Krnl_LS)

Krnl_LS	**Solis-Wets**	**ADADELTA**
Total time (%)	99	99
# Calls	120	46
Avg. time (ms)*	225	5526
Occupancy (%)	20	10

LS is the bottleneck

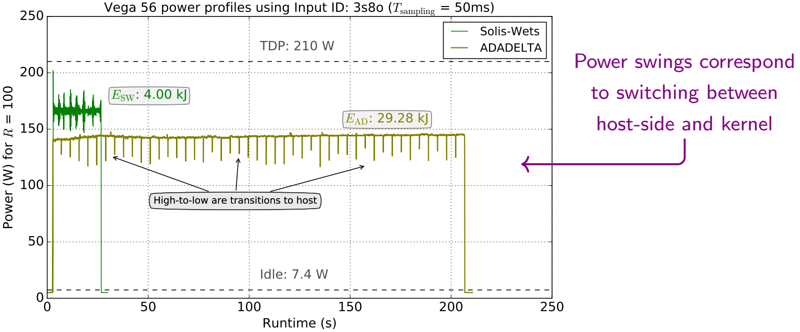
Solis-Wets requires more kernel enqueues ($\# \ Calls$) than ADADELTA

Overall duration: $\# \ Calls \times Avg. \ time$

In both cases, GPU utilization is low

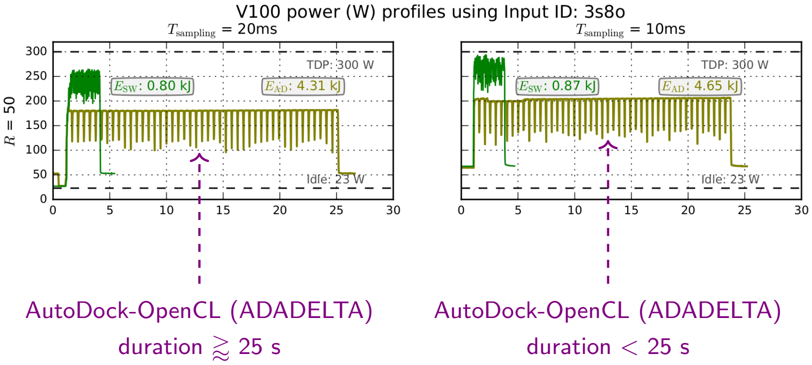
*Avg. time measured per kernel enqueue

Preliminary power profiling: Vega 56 GPU



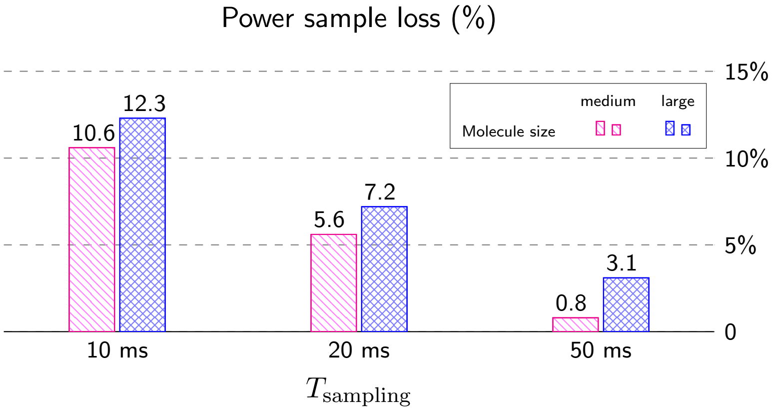
- Internal sensors accessed through software-based meters
  - $T_{\text{sampling}} = 50$ ms (max. supported on Vega 56)
- Energy consumption (power integrated over time)
  - Solis-Wets: $E_{\text{SW}} = 4.0$ kJ
  - ADADELTA: $E_{\text{AD}} = 29.3$ kJ

Power profiling: V100 GPU



- Sampling period does not affect profile shape ...
- ... but the shorter T_{sampling}, the more samples are lost
- Energies at $T_{\text{sampling}} = \{20, 10\}$ ms are slightly different

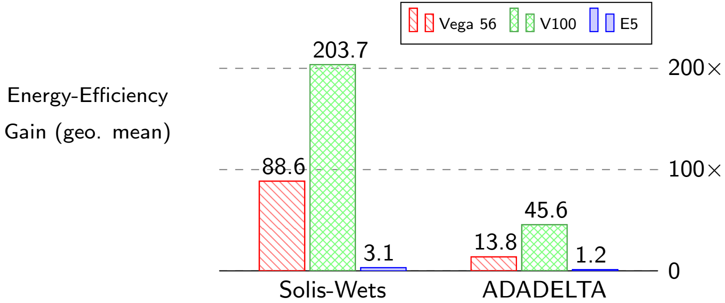
Impact of T_{sampling} on V100 power



- More samples are lost with $T_{\text{sampling}} = 10$ ms
- Next experiments are performed using $T_{\text{sampling}} = 50$ ms

Energy-Efficiency Gain

- Accelerator devices
  - GPUs: Vega 56 (on-premise), V100 (AWS p3.2xlarge)
  - CPUs: E5-2666 v3 CPU (18 cores, AWS c4.8xlarge)
- Baseline: original AutoDock
  - Implements only Solis-Wets LS method
  - Does not support multithreading



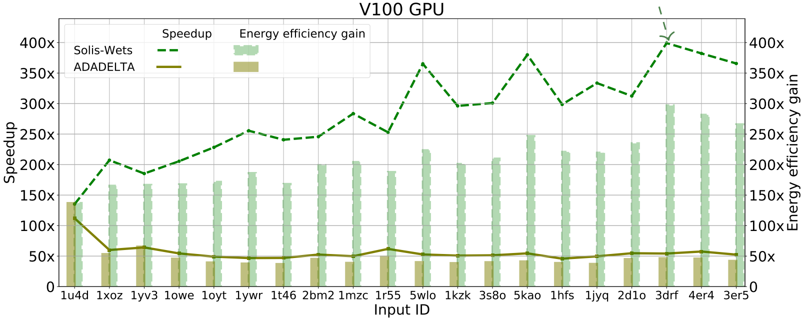
Performance & Energy Efficiency

Efficiency:
Solis-Wets > ADADELTA

GPU: highest gains on 3drf

$\text{Speedup}_{\text{SW}} \sim 400\times$

$\text{Energy}_{\text{SW}} \sim 297\times$



Growing molecular complexity

Higher Solis-Wets efficiency

Higher ADADELTA efficiency

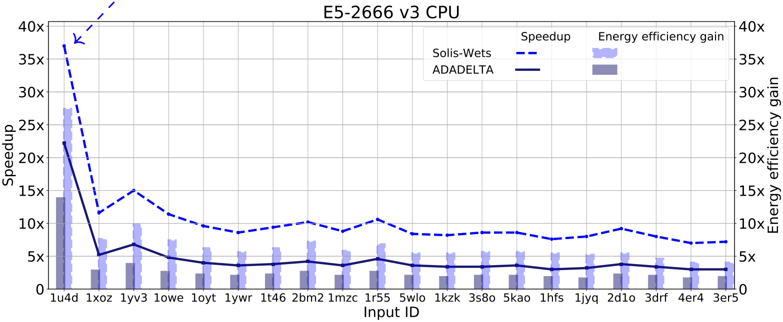
Performance & Energy Efficiency

CPU: highest gains on 1u4d

$\text{Speedup}_{\text{SW}} \sim 18\times$

$\text{Energy}_{\text{SW}} \sim 14\times$

GPUs faster than CPUs: more suitable mapping of AutoDock-OpenCL onto GPU hardware



Growing molecular complexity

Higher Solis-Wets efficiency

Higher ADADELTA efficiency

Final Remarks

1. AutoDock-OpenCL: comparing LS methods
   - ADADELTA
     - Lower speedups ☹
     - Lower energy efficiencies ☹
     - Higher quality of dockings ☺ (for complex molecules)
   - Solis-Wets
     - More efficient for small molecules
2. Energy gains
   - V100 GPU: *most* efficient
     - Solis-Wets: $\sim$297$\times$, ADADELTA: $\sim$137$\times$
   - E5 CPU: *least* efficient
     - Solis-Wets: $\sim$13$\times$, ADADELTA: $\sim$7$\times$
3. GPUs more efficient than CPUs
   - Fine-grained parallelization more suitable for GPUs

Evaluating the Energy Efficiency of OpenCL-accelerated AutoDock Molecular Docking

https://github.com/ccsb-scripps/AutoDock-GPU

Leonardo Solis-Vasquez

solis@esa.tu-darmstadt.de

https://www.esa.tu-darmstadt.de