Réseau de Contrainte Ternaire pour une Propagation Efficace de Bornes sur GPU

Journées Francophones de Programmation par Contraintes (JFPC 2025)

Pierre Talbot
pierre.talbot@uni.lu
https://ptal.github.io

2nd July 2025

University of Luxembourg

UNIVERSITÉ DU LUXEMBOURG

Motivation

- Machine learning (deep learning, reinforcement learning, ...) has seen tremendous speed-ups (e.g. 100x, 1000x) by using GPU.
- Some (sequential) optimizations on CPU are made irrelevant if we can explore huge state space faster.

Can we replicate the success of GPU on machine learning applications to combinatorial optimization?

State of the Art: Combinatorial Optimization on GPU

Very scarce literature, usually:

- **Heuristics**: often population-based algorithms[1].
- **Limited set of problems**[2]
- **Limited GPU parallelization**: offloading to GPU specialized filtering procedures[3,4].
- **cuOpt**: new MILP solver—relaxation on GPU, search on CPU[5].

No general-purpose constraint solver on GPU.

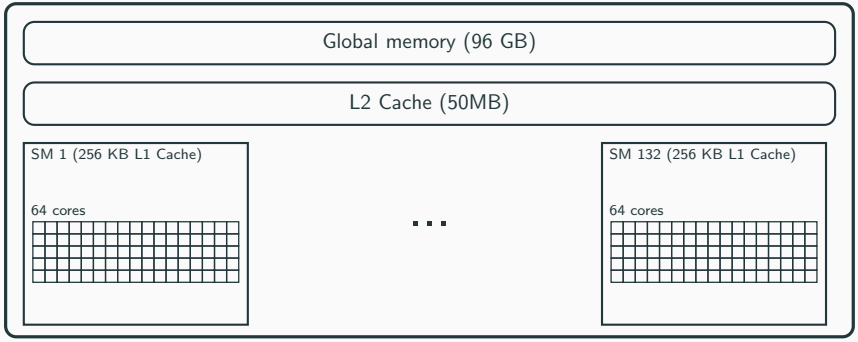
[1] A. Arbelaez and P. Codognet, *A GPU Implementation of Parallel Constraint-Based Local Search*, PDP, 2014.
[2] Jan Gmys. Exactly Solving Hard Permutation Flowshop Scheduling Problems on Peta-Scale GPU-Accelerated Supercomputers. INFORMS Journal on Computing, 2022.
[3] F. Campeotto et al., *Exploring the use of GPUs in constraint solving*, PADL, 2014
[4] F. Tardivo et al., *Constraint propagation on GPU: A case study for the AllDifferent constraint*, Journal of Logic and Computation, 2023.
[5] Using primal-dual linear programming (PDLP).

Our Contributions

- **First general constraint solver fully executing on GPU (propagation + search).**
  ⇒ **General**: Support MiniZinc and XCSP3 constraint models.
  ⇒ **Simple**: interval-based constraint solving + backtracking search (no global constraints, learning, restart, event-based propagation, ...).
  ⇒ **Efficient?**: Almost on-par with Choco (21% better, 30% worst, 49% equal).
  ⇒ **Open-source**: Publicly available on `https://github.com/ptal/turbo`.
- **Ternary constraint network**: representation of constraints suited for GPU architectures.

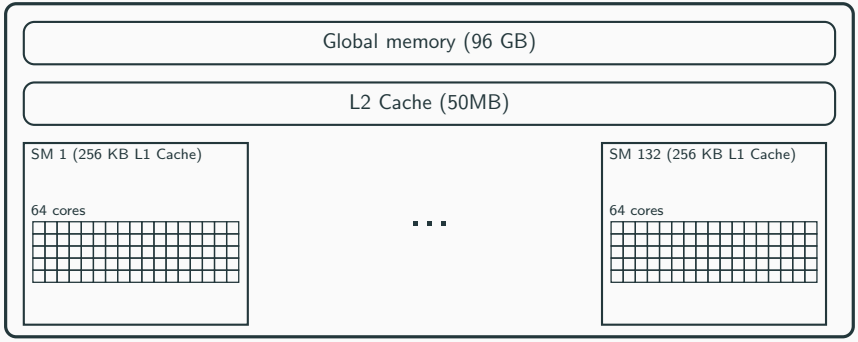
Overview

(Simplified) Architecture of the GPU Nvidia H100



8448 cores grouped in 132 streaming multiprocessors (SM) of 64 cores each.

(Simplified) Architecture of the GPU Nvidia H100



8448 cores grouped in 132 streaming multiprocessors (SM) of 64 cores each.

$\Rightarrow$ **Oversubscribe** (to hide memory latency): 1024 threads per SM

135168 threads running in parallel!