Computational performance of the GPU-accelerated LP-solver cuPDLP-C (topic presentation)

Pyry Ruotsalainen

17.06.2024

Advisor: Olli Herrala

Supervisor: Fabricio Oliveira

Aalto-yliopisto
Perustieteiden
korkeakoulu

Background – computational challenges in LP-solvers

- **Computational time** of large-scale linear and stochastic optimisation problems can be significant and impractical.
- Solution time **increases** as problem size **increases**
  - 90000 constraints, 70000 variables: 9 seconds
  - 7 million constraints, 39 million variables: 509 seconds
- Sample of Mittelmann benchmarks:

	Solvers									
Problem	**CLP**	**Gurob**	**COPT**	**MDOPT**	**OPTV**	**MOSEK**	**HiGHS**	**GLOP**	**SPLX**	**XOPT**
neos-3025225	663	20	11	20	17	9	94	96	430	65
neos5052403	188	4	4	2	4	4	24	370	298	15
neos5251015	249	4	3	1	1	6	151	99	3288	36
irish-e	297	3	5	17	5	5	21	t	440	15

Background – efficient solvers and parallelization

- Three well-studied **algorithms behind the LP-solvers**:
  - Simplex-based methods
  - Interior point method
  - First-order methods, for example PDLP

- Performance improvements by **CPU-parallelism**

Background – PDLP-method and cuPDLP-C solver

- Simplex and interior point methods:
  - **Solving sparse linear systems** computational bottleneck
  - GPUs not optimal for solving sparse linear systems

- PDLP-method:
  - **Matrix-vector multiplication** computational bottleneck
  - Modern GPU infrastructures designed for that bottleneck

- Only PDLP-method has potential for the GPU-parallelism
  - Open source design for GPUs: cuPDLP developed by HiGHS

Objective

- Evaluate computational performance of the GPU accelerated LP-solver cuPDLP-C with big reference problems of different sizes.
- Evaluate and compare the computational performance with reference solvers.
  - Replicating Mittelmann benchmarks for cuPDLP, and CPU parallelized version of simplex and interior point method based HiGHS solvers.
- Recognize possible strengths and weaknesses of cuPDLP-C solver analysing solver competitiveness and problem types.

Tools

- Triton (Aalto high-performance computing cluster) used for running calculations using CPUs and GPUs
- cuPDLP-C open source solver used for solving LPs on GPUs using the first-order algorithm
- HiGHS open source solver used for the CPU versions
- Mittelmann benchmarks used as a basis for the reference problems (https://plato.asu.edu/ftp/lpopt.html)

Schedule

- Introduction to the topic 04/2024
- Setting up and first tests of the cuPDLP-solver 05/2024
- Writing the thesis 06-07/2024
- Implementation of the computational performance evaluation 06-07/2024
- Results presentation 08/2024
- Thesis ready 08/2024

References

- Huangfu, Q. and Hall J. A. J, 2018, *Parallelizing the dual revised simplex method*, Mathematical Programming Computation, 10 (1), 119-142. DOI: 10.1007/s12532-017-0130-5
- Lu, H. and Yang, J. 2024, *cuPDLP.jl: A GPU Implementation of Restarted Primal-Dual Hybrid Gradient for Linear Programming in Julia,* https://arxiv.org/abs/2311.12180
- Koch, T., Berthold, T., Pedersen, J. and Vanaret, C., 2022, *Progress in mathematical programming solvers from 2001 to 2020,* EURO Journal on Computational Optimization, 10 (2022) 10003
- CuPDLP-C: https://github.com/COPT-Public/cuPDLP-C