Leveraging GPUs for Effective Clause Sharing in Parallel SAT Solving

Nicolas Prevot, Mate Soos, Kuldeep S. Meel

June 27, 2021

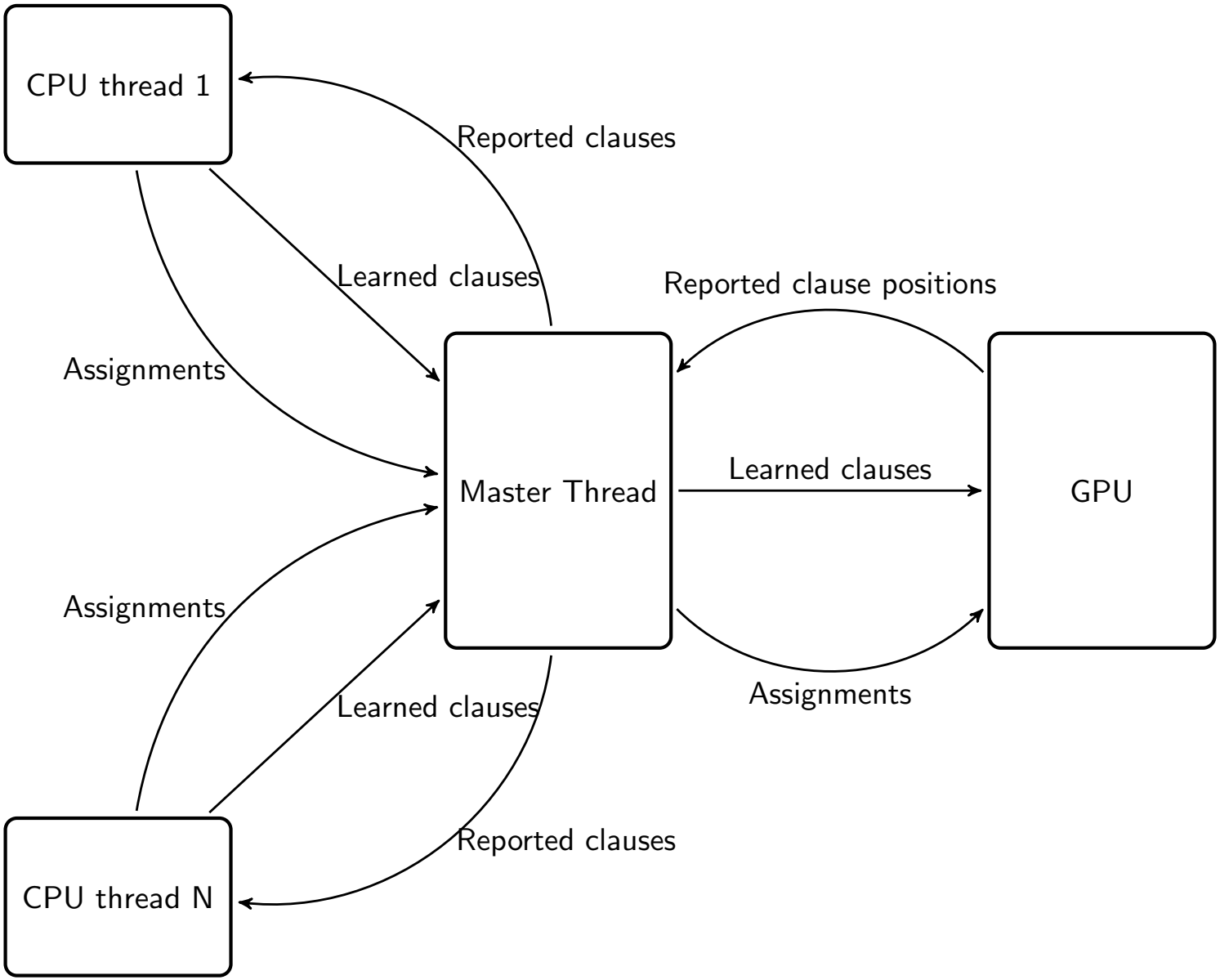
Overview

- GPUs are powerful, but it is not easy to use them for SAT solving
- CPU for the CDCL procedure
- GPU for clause exchange between threads

Clause exchange

- Portfolio approach: multiple CPU threads running concurrently
- They share clauses
- Sharing too many clauses leads to a slowdown
- We need to identify good clause and only share those
- Some state-of-the-art solvers share only clauses with small size or lbd

Architecture



Our approach

- Good clauses are used, in propagation or in conflict analysis
- Clauses that have been used are more likely to be used again
- Identify clauses which would have been used recently, import those
- Consider assignments where unit propagation completed without conflict

Example of used clauses

- Assume thread completes propagation with: $v_1 \leftarrow F, v_2 \leftarrow F$
  - Clause $v_1 \vee v_2 \vee v_3$ would have propagated $v_3 \leftarrow T$
  - Clause $v_1 \vee v_2$ would have been in conflict
  - Clause $\neg v_1 \vee v_2$ would not have told us anything new
- We say a clause triggers an assignment if it would have propagated a literal or been in conflict

Bitwise representation of assignments

	v_1	v_2
assignment A	T	U
assignment B	U	F
isSet	10	01
isTrue	10	00

Testing a clause with bitwise operations

Clause: $v_1 \vee v_2$, 4 assignments

variable	values	allFalse	oneUndef
		1111	0000
v_1	TFUU	0100	0011
v_2	FFFU	0100	0010

Answer: 0110, the clause triggers on the second and third assignments
All clauses can be tested independantly from each other, so can be done on the GPU

Group testing

- Testing a clause on 32 assignments at once still isn't fast enough
- Group assignments by the solver thread they come from
- Test a clause on the groups first, only test it on individual assignments if it's positive

Pooled assignment

	v_1	v_2	v_3
assignment A	T	F	U
assignment B	U	F	F
pooled assignment	{T, U }	{F}	{U, F}

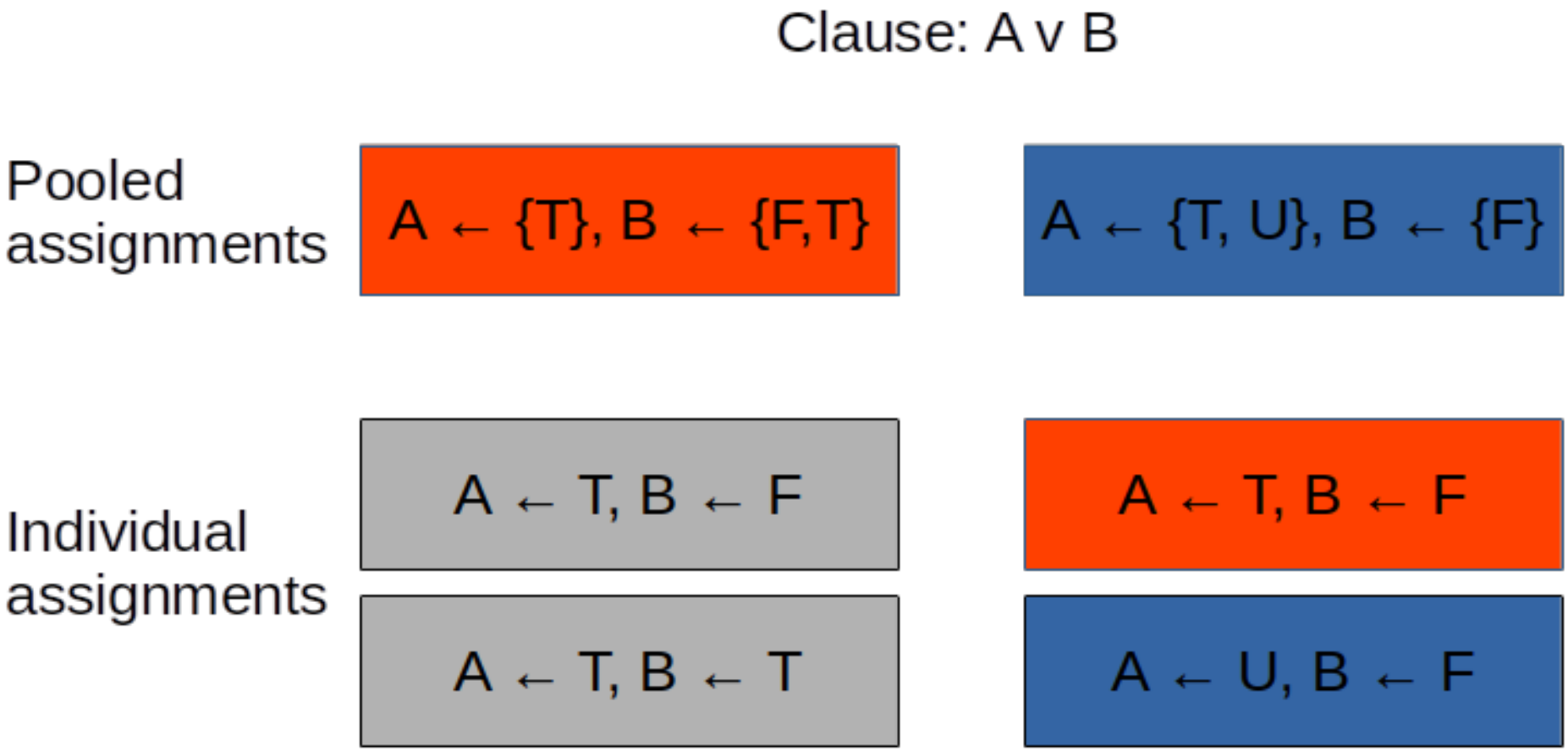
If a clause does not trigger on a pooled assignment, it does not trigger on any of the associated assignments

Bitwise representation of pooled assignments

	v_1	v_2
pooled assignment A	{T, U}	{U}
pooled assignment B	{U}	{T, F, U}
canBeTrue	10	01
canBeFalse	00	01
canBeUndef	11	11

With this representation, we can test a clause on up to 32 pooled assignments at once. All clauses can be tested independantly from each other, so can be done on the GPU

Overalll clause testing



- Blue: positive tests
- Red: negative tests
- Grey: tests that did not need to be done

Comparison with glucose-syrup

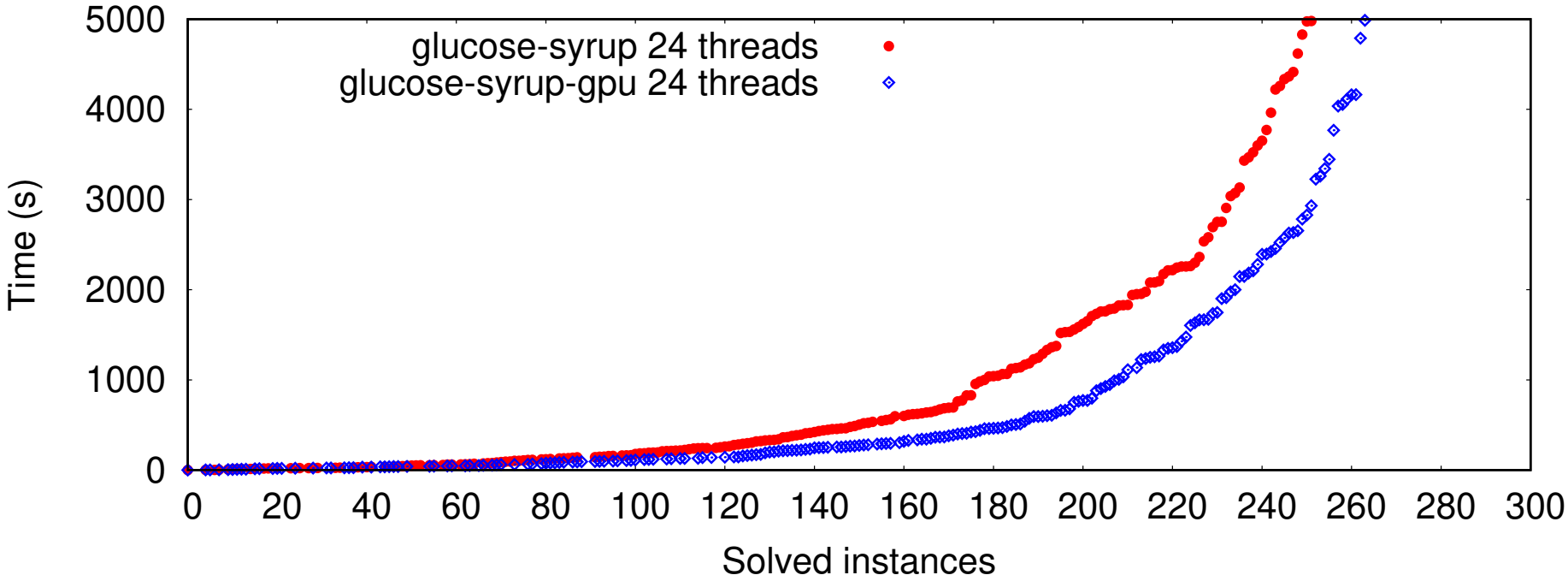


Figure: glucose-syrup vs glucose-syrup + GpuShareSat.

- glucose-syrup solved 251 instances
- glucose-syrup + GpuShareSat solved 263 instances

Comparison with P-MCOMSPS-STR

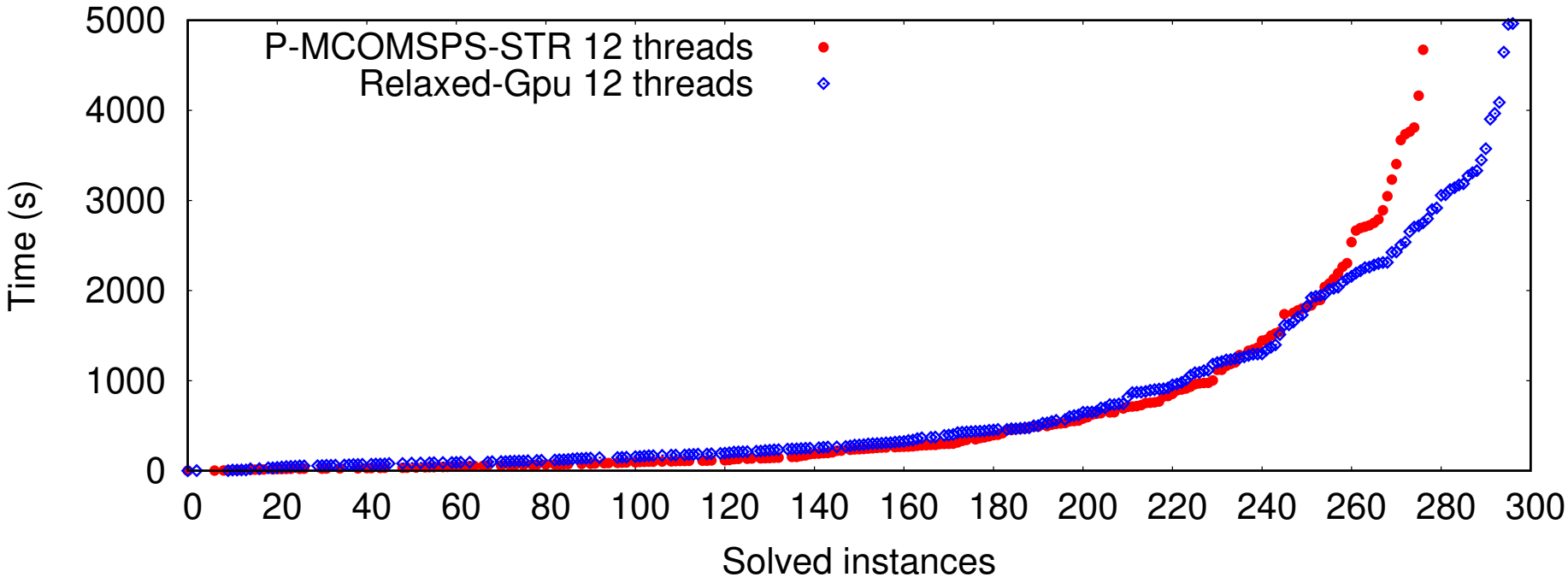


Figure: P-MCOMSPS-STR vs Relaxed_LCMDCBDL_newTech + GpuShareSat

- P-MCOMSPS-STR solved 276, Relaxed_LCMDCBDL_newTech + GpuShareSat solved 296
- P-MCOMSPS-STR came first in SAT 2020 parallel competition
- Relaxed_LCMDCBDL_newTech came third in SAT 2020 single threaded competition

Conclusion

- Implemented as a library, at `https://github.com/nicolasprevot/GpuShareSat`
- Designed to make it easy to use from any solver
- You are encouraged to add it to your solver