Università della Svizzera italiana
Facoltà di scienze informatiche

Clause Sharing and Partitioning for Cloud-Based SMT Solving

Matteo Marescotti, Antti E. J. Hyvärinen, and Natasha Sharygina

Università della Svizzera italiana
Switzerland

SMT@IJCAR2016

USI

Why parallel SMT

- Software verification tasks are a source of computationally challenging problems.
- SMT solvers are often used as backend for software verification techniques.
- Given a way to parallelise SMT solver, it would be possible to speedup software verification.
- Parallel computing in e.g. economics and physics-related domains has been a huge success: run time reduced from years to days.
- Parallel constraint solving has only recently gained success
  - Heule et al.: Solving and Verifying the boolean Pythagorean Triples problem via Cube-and-Conquer, to appear at SAT16

USI

Related work on parallel SMT solving

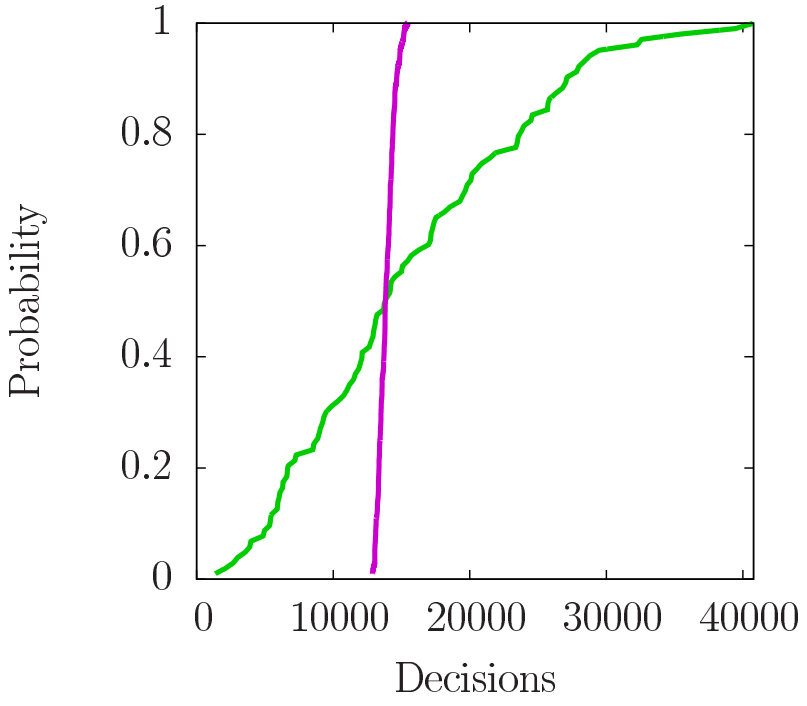
- **Portfolio [BCD11, WHM09]:**
  - The concurrent execution of a set of solvers.
  - Each solver has a different initial seed.
- **Divide and Conquer [REI14]:**
  - Creation of a set of partitions $F_1, \ldots, F_n$
  - $\text{sat}(F) \iff \text{sat}(F_1 \vee \ldots \vee F_n) \iff \text{sat}(F_1) \vee \ldots \vee \text{sat}(F_n)$
  - First proved sat proves sat, all proved unsat proves unsat.
- **Clause Sharing [WHM09]:**
  - Feature of parallel SAT/SMT techniques.
  - Used to prevent the search on already explored space.

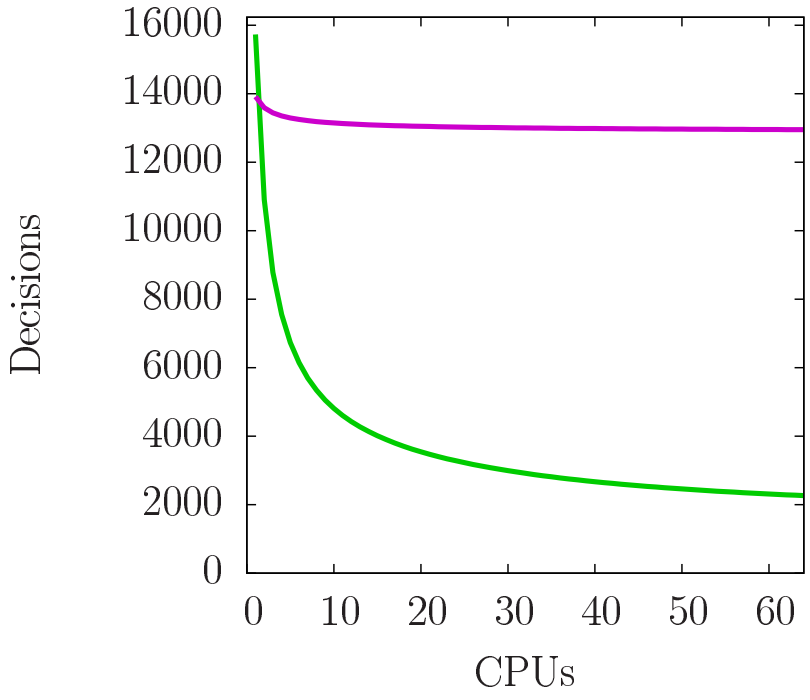
[BCD11] Barrett et al.: CVC4. 2011.
[WHM09] Wintersteiger et al.: A concurrent portfolio approach to SMT solving. 2009.
[REI14] Reisenberger et al.: PBoolector: a parallel SMT solver for QF_BV […]. 2014.

USI

Portfolio

- Parallel approaches try to take advantage from randomness.
- Formulas with a more widespread run-time distribution are better suited for Portfolio.





✶ data from two SAT instances solved with MiniSAT 100 times

USI

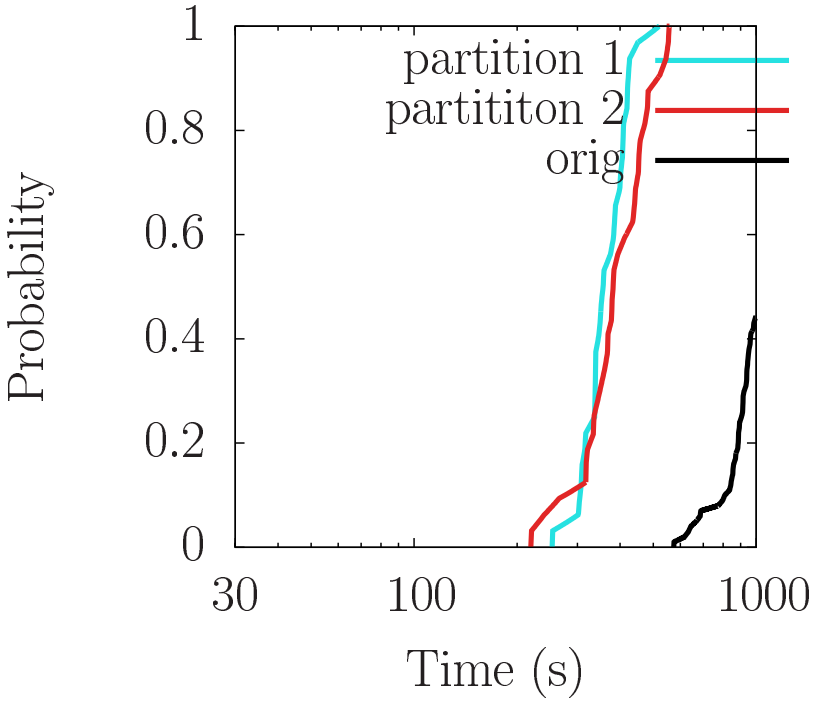
Partitioning

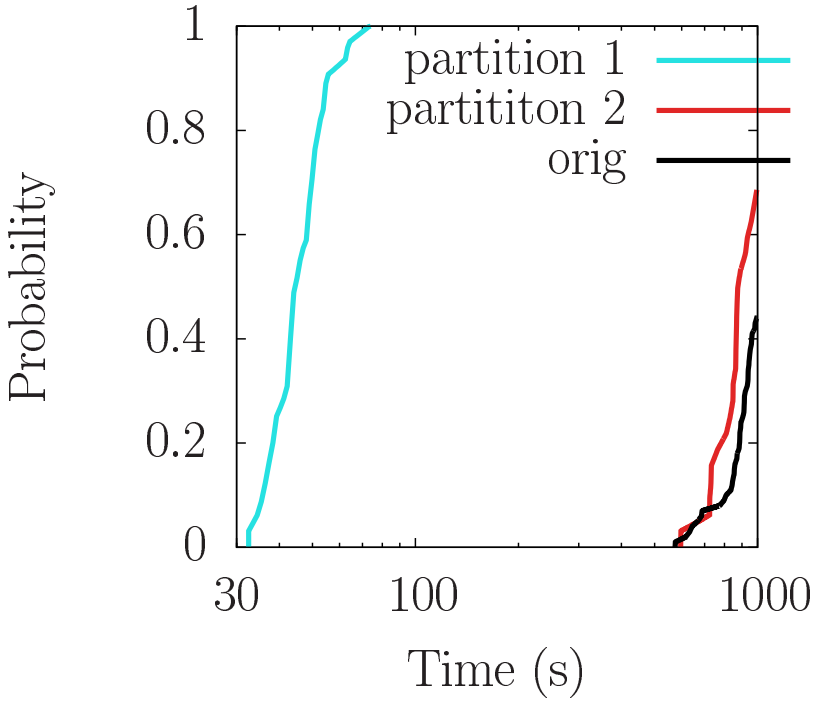
- Given a number of partitions n, partitioning tries to formulate n partitions such that each partition's search space is equal.
- Each partition is created by constraining the input formula with new clauses.
- Scattering based on VSIDS (Variable State Independent Decaying Sum)
  - Formula is solved for a fixed amount of time.
  - We choose the literals which appear most during conflict analysis (VSIDS) to construct the partitions.
- Lookahead

USI

Partitioning

- Depending on how the division is done, the performance might be dramatically different.





✶ data from OpenSMT2: splitting *NEQ016_size8* QF_UF benchmark with VSIDS and lookahead

Learned clauses in DPLL(T)

- A new clause $c \Leftarrow F$ is *learnt* by:
  - SAT solver on every conflict.
  - Theory Solver which periodically checks if the current propositional assignment is consistent with the theory.
- Learnt clauses are used to guide the search.

USI

Clause sharing in DPLL(T) [MHS16]

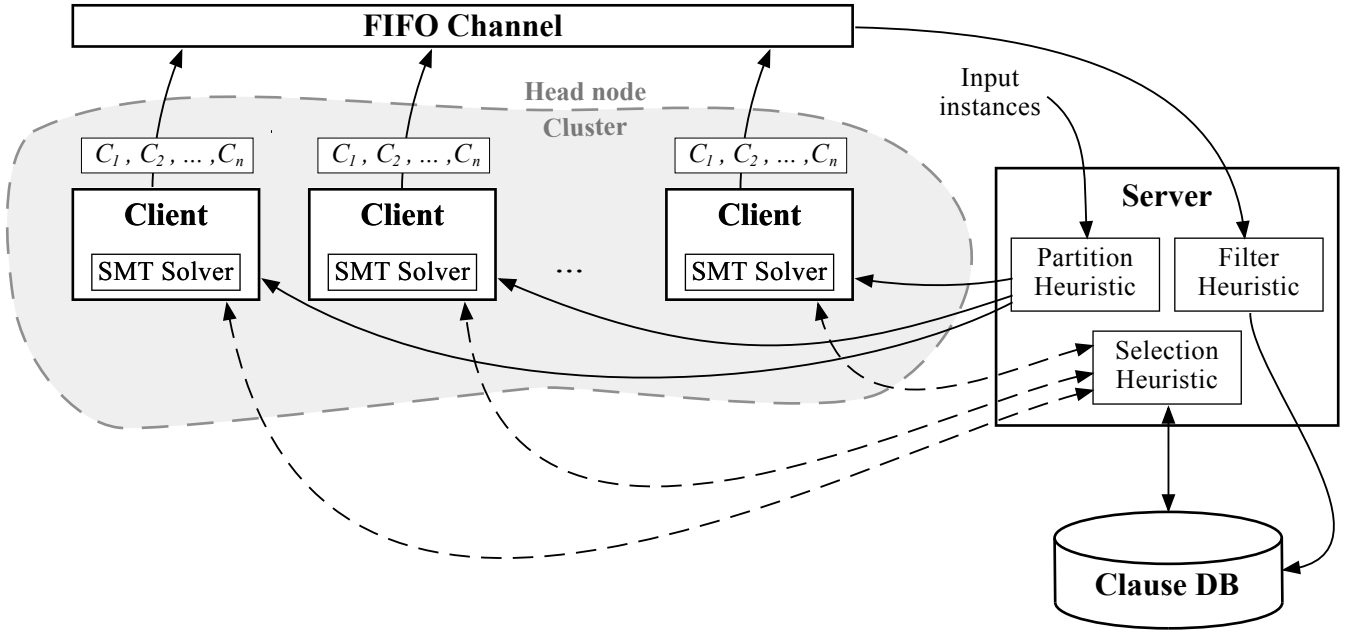
- Clause sharing helps the solving of:
  - unsat to produce an unsat set of clauses.
  - sat to reduce the number of assignment.
- Clause sharing could slow down all the solvers due to the high throughput of learnt clauses.
- More critical the more the number of solvers increases.
- Heuristics must be used to choose what is worth sharing.
- All solvers must have the same internal representation of the formula for clause sharing to be sound.

[MHS16] M. Marescotti, A. E. J. Hyvärinen, and N. Sharygina.
Clause Sharing and Partitioning for Cloud-Based SMT Solving. *to appear at ATVA 2016, proc*.

USI

Parallel Solver [HMS15, MHS16]

- Server / Workers **modular** design.
- Load balancing: as simpler partitions are solved, solvers are assigned to unsolved partitions in a fair way.



[HMS15] A. E. J. Hyvärinen, M. Marescotti, and N. Sharygina.
Search-space partitioning for parallelizing SMT solvers. *SAT 2015, Proceedings*.

[MHS16] M. Marescotti, A. E. J. Hyvärinen, and N. Sharygina.
Clause Sharing and Partitioning for Cloud-Based SMT Solving. *to appear at ATVA 2016, proc*.

USI

Parallel Solver [HMS15, MHS16]

- The modular system runs in a computing cloud.
- Partition heuristics:
  - The *VSIDS*-based heuristic used in *MiniSat*.
  - A *look-ahead* heuristic.
- Filter heuristic:
  - Statically based on clause size
- Selection heuristic:
  - Random fixed number of clauses
- SMT Solver:
  - **OpenSMT** supporting **QF_UF** and **QF_LRA**.

[HMS15] A. E. J. Hyvärinen, M. Marescotti, and N. Sharygina.
Search-space partitioning for parallelizing SMT solvers. *SAT 2015, Proceedings*.
[MHS16] M. Marescotti, A. E. J. Hyvärinen, and N. Sharygina.
Clause Sharing and Partitioning for Cloud-Based SMT Solving. *to appear at ATVA 2016, proc*.

USI

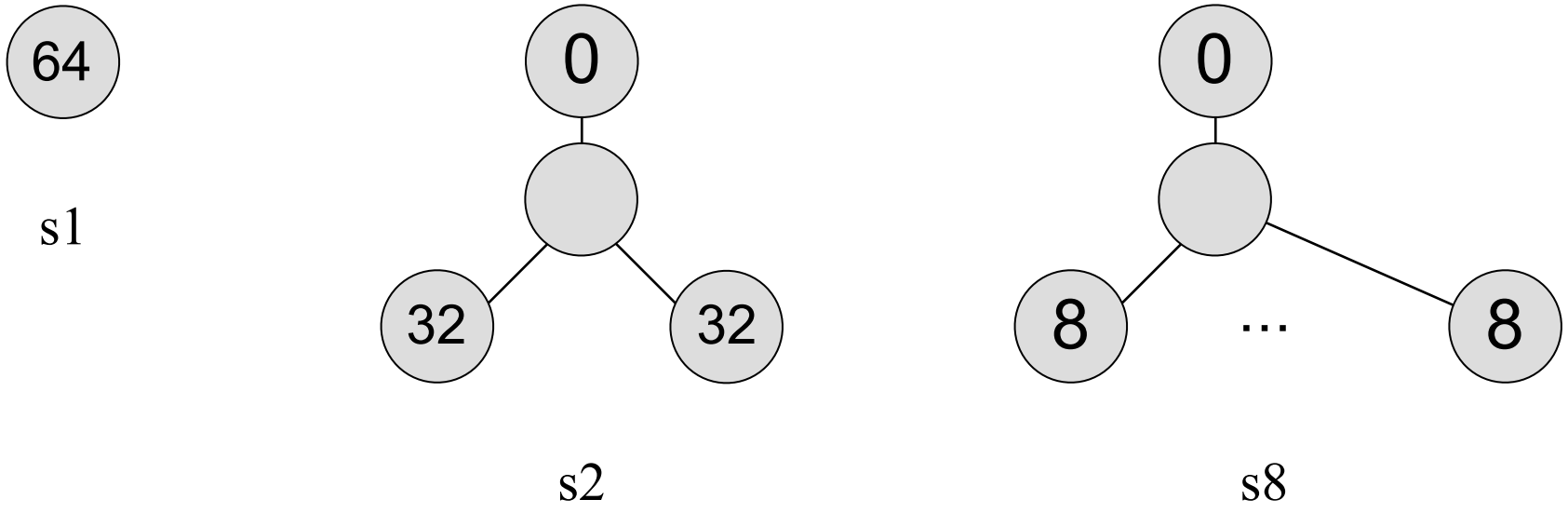
Binary format for SMT

- SMT solver represent the instance as DAG of terms.
- Several simplifications applied. The result is a set of clauses.
- Hard to connect the clauses and the DAG representation.
- We solve this by having a binary format that describes the state of the solver.
  - Binary format used both for learned clauses and the clauses used as constraints.
  - Our naïve implementation consists of a memory dump.
- As a downside it is difficult to plug in other solvers to the framework.

USI

Experiments

- Combination using the parallelisation tree framework [HMS15].

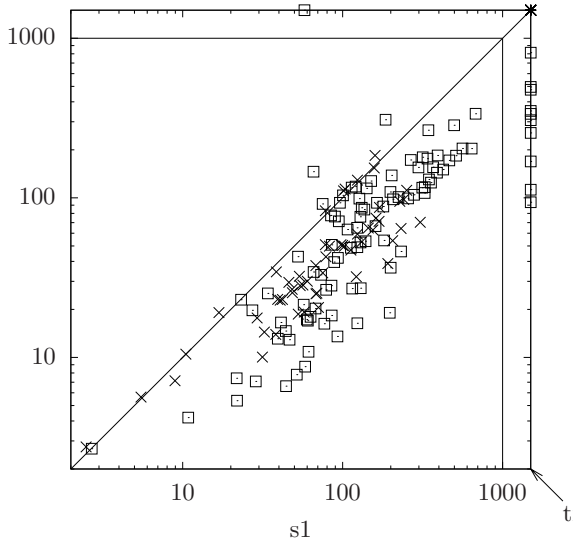


[HMS15] A. E. J. Hyvärinen, M. Marescotti, and N. Sharygina.
Search-space partitioning for parallelizing SMT solvers. *SAT 2015, Proceedings*.

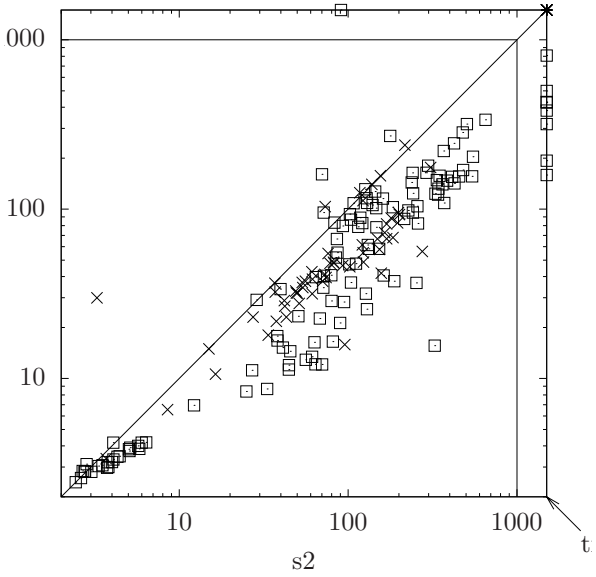
USI

Experiments: Clause Sharing

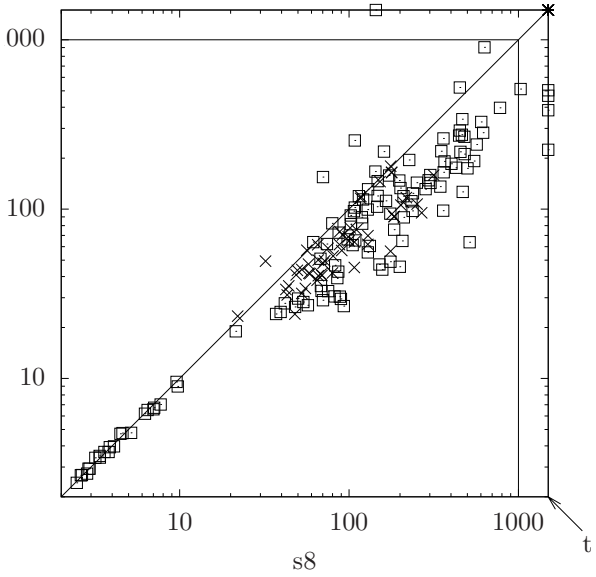
- Works better with pure portfolio.



2.05x
1 more QF_LRA
9 more QF_UF



1.97x
1 more QF_LRA
7 more QF_UL

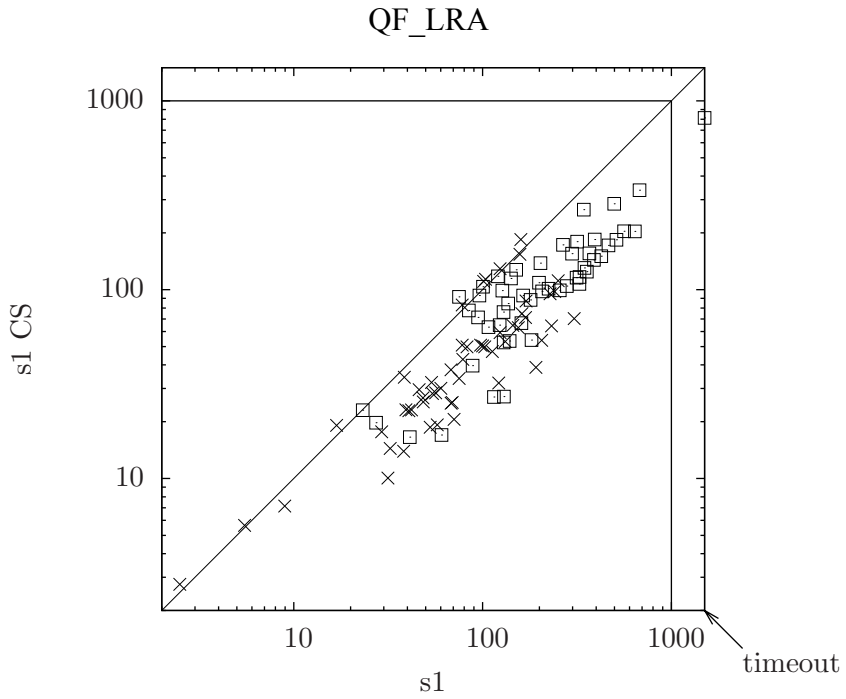


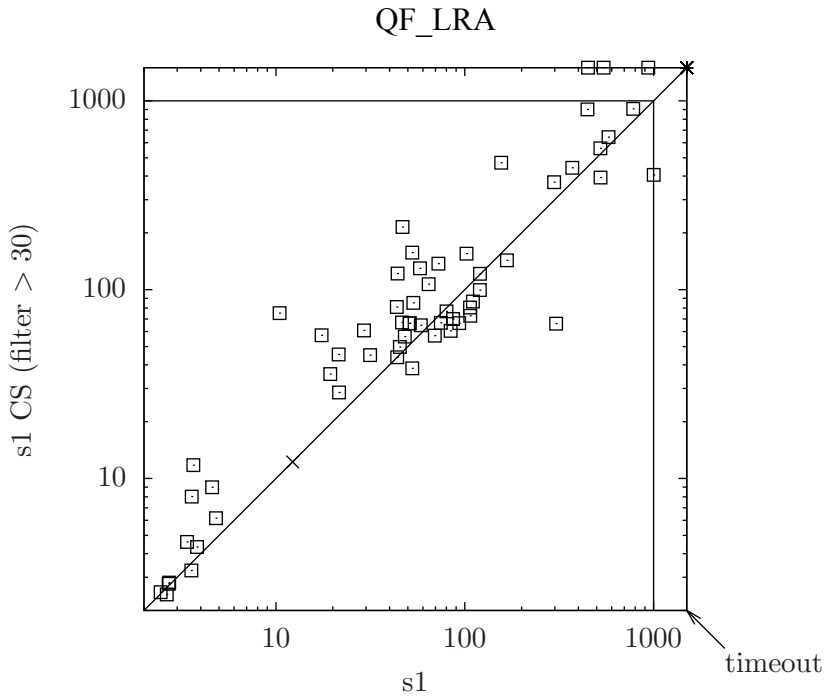
1.67x
0 more QF_LRA
4 more QF_UF

✳ data from OpenSMT2 Parallel Framework using 8 nodes on QF_UF and QF_LRA benchmarks.
s1: Portfolio of 64 solvers.
s2: Partition in 2 and portfolio of 32 on each. s8: Partition in 8 and portfolio of 8 on each.

USI

Experiments: Clause Sharing Heuristics





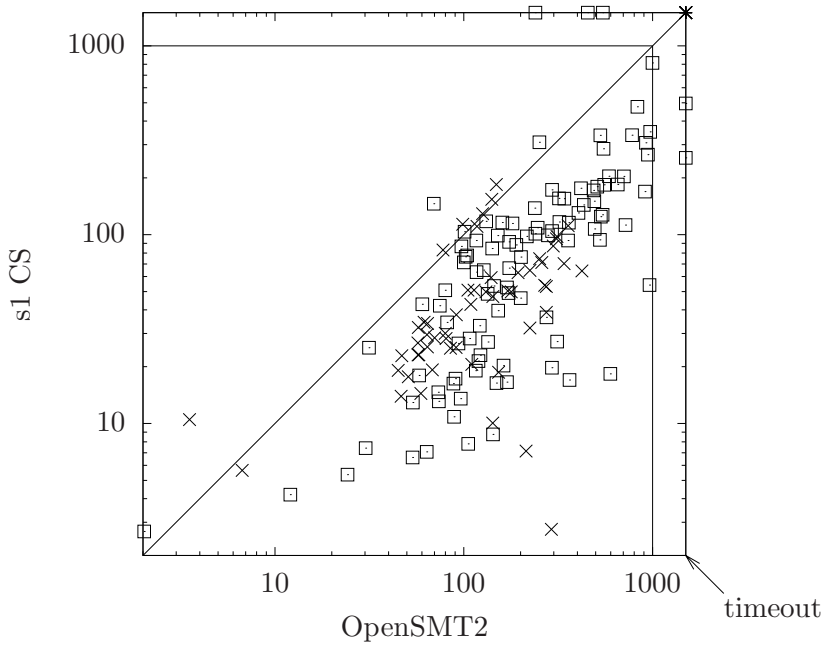
s1:	portfolio of 64 solvers
s1 CS:	portfolio of 64 solvers with clause sharing

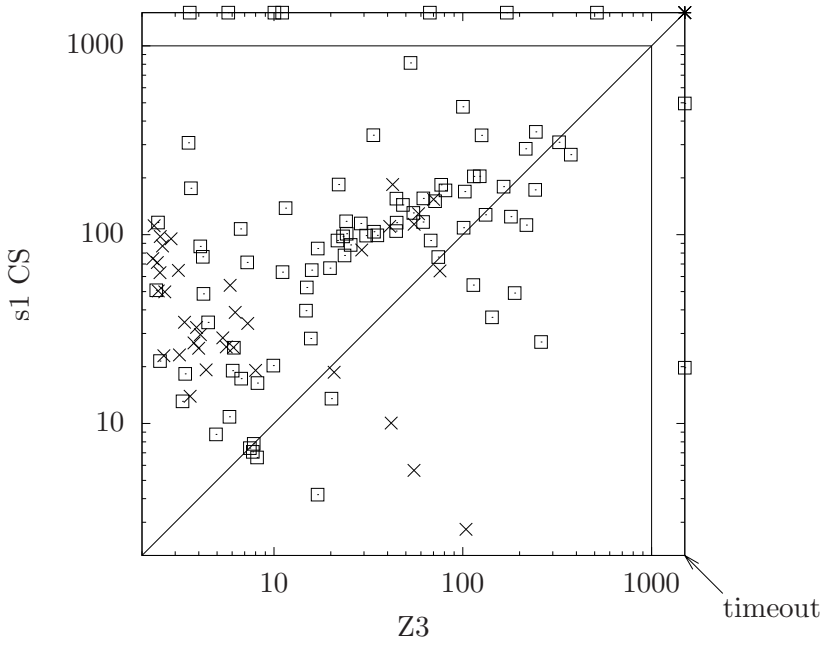
✳ data from OpenSMT2 Parallel Framework using 8 nodes on QF_LRA benchmarks.

[MHS16] Matteo Marescotti, Antti E. J. Hyvärinen, and Natasha Sharygina.
Clause Sharing and Partitioning for Cloud-Based SMT Solving. *to appear at ATVA 2016, proc*.

USI

Experiments: Against sequential





s1 CS: portfolio of 64 solvers with clause sharing

✴ data from OpenSMT2 Parallel Framework using 8 nodes on QF_UF and QF_LRA benchmarks.

USI

Future work

- Loose clause sharing heuristics result in slowdown:
  - how to improve these heuristics.
- Full iterative partitioning support in the implementation:
  - currently limited to the first level.
- Global clause sharing:
  - now limited to the tree rooted at the node who learnt them.
- Integration in model checker.
- OpenSMT2 GIT Repo at:

https://scm.ti-edu.ch/repogit/opensmt2.git

USI