A Propagation Rate based Splitting Heuristic for Divide-and-Conquer Solvers

Saeed Nejati, Zack Newsham, Joseph Scott, Jimmy Liang, Catherine Gebotys, Pascal Poupart and Vijay Ganesh

University of Waterloo

September 1st
SAT 2017

Introduction

- Parallel SAT solvers (Availability of computing nodes)
- Portfolio, Divide-and-Conquer
- Divide-and-Conquer: Split the formula into several sub-formulas and solve them using CDCL solvers in parallel, and share information while solving

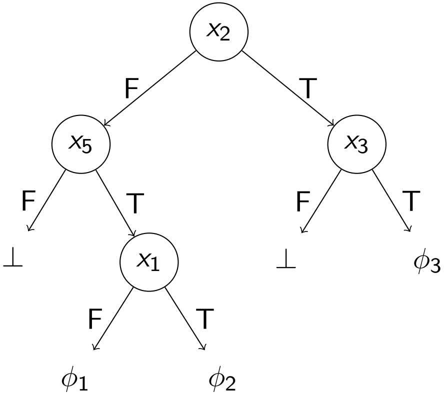
Introduction

- Parallel SAT solvers (Availability of computing nodes)
- Portfolio, Divide-and-Conquer
- Divide-and-Conquer: Split the formula into several sub-formulas and solve them using CDCL solvers in parallel, and share information while solving

- How to "Divide" so the "Conquer"s become easier?

Search Space Partitioning

- $\phi_1 = \phi \wedge \neg x_2 \wedge x_5 \wedge \neg x_1$
- $\phi_2 = \phi \wedge \neg x_2 \wedge x_5 \wedge x_1$
- $\phi_3 = \phi \wedge x_2 \wedge x_3$



AMPHAROS - Baseline

- AMPHAROS as a baseline for our implementation

AMPHAROS - Baseline

- AMPHAROS as a baseline for our implementation
- Divide-and-Conquer parallel solver

AMPHAROS - Baseline

- AMPHAROS as a baseline for our implementation
- Divide-and-Conquer parallel solver
- Dynamically partitions/splits the search space

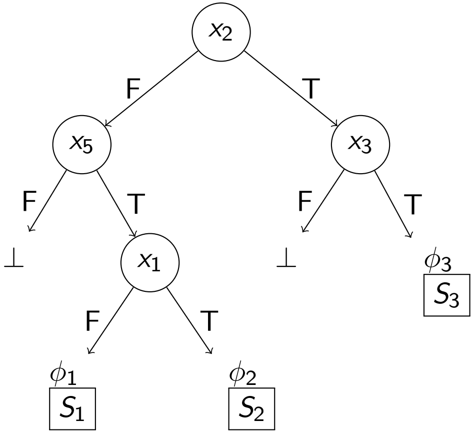
AMPHAROS - Baseline

- AMPHAROS as a baseline for our implementation
- Divide-and-Conquer parallel solver
- Dynamically partitions/splits the search space
- Uses VSIDS to pick the next variable for splitting

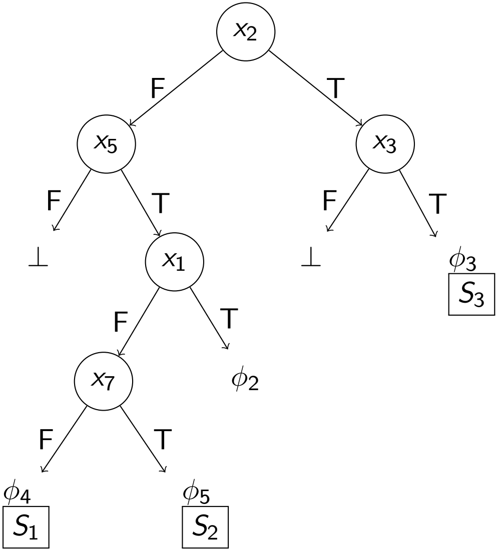
AMPHAROS - Baseline

- AMPHAROS as a baseline for our implementation
- Divide-and-Conquer parallel solver
- Dynamically partitions/splits the search space
- Uses VSIDS to pick the next variable for splitting
- Adaptive load balancing of solvers over cubes

AMPHAROS - Baseline



AMPHAROS - Baseline



What's added?

- Propagation rate-based splitting heuristic
- Worker Diversification

Backend Solver

- Fairly modular, easy to modify
- Included: Minisat, Glucose
- Added: **MapleSAT**
- Small improvement over existing workers

Splitting Heuristic

- Propagation rate

Splitting Heuristic

- Propagation rate
  - For each variable: (# of propagations / # of decisions)
  - Pick the variable with the highest rate (at the conflict limit)

Splitting Heuristic

- Propagation rate
  - For each variable: (# of propagations / # of decisions)
  - Pick the variable with the highest rate (at the conflict limit)
- A dynamic metric

Splitting Heuristic

- Propagation rate
  - For each variable: (# of propagations / # of decisions)
  - Pick the variable with the highest rate (at the conflict limit)
- A dynamic metric
- Computed during solving of each cube

Splitting Heuristic

- Propagation rate
  - For each variable: (# of propagations / # of decisions)
  - Pick the variable with the highest rate (at the conflict limit)
- A dynamic metric
- Computed during solving of each cube
- Minimal computation overhead

Splitting Heuristic

- Propagation rate
  - For each variable: (# of propagations / # of decisions)
  - Pick the variable with the highest rate (at the conflict limit)
- A dynamic metric
- Computed during solving of each cube
- Minimal computation overhead
- Smaller sub-formulas are expected after splitting

Worker Diversification

- Similar to the Portfolio solvers approach

Worker Diversification

- Similar to the Portfolio solvers approach
- Used different restart strategies for worker solver

Worker Diversification

- Similar to the Portfolio solvers approach
- Used different restart strategies for worker solver
- The best configuration in our experiments:
  - Luby + Geometric + MABR (Multi-Armed Bandit Restart)

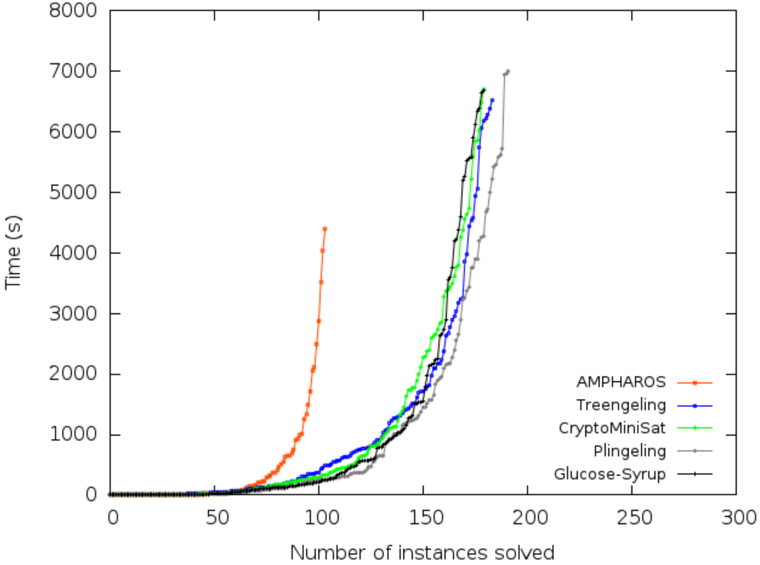
Results

Experimental Setup

- Machines:
  - 8 core Intel Xeon CPUs @ 2.53 GHz
  - 16GB RAM
- Benchmarks:
  - SAT 2016 Application
    - 300 instances
    - 2-hour timeout
  - Cryptographic Hash functions
    - Preimage of 21, 22, 23 rounds of SHA-1
    - 48-hour timeout

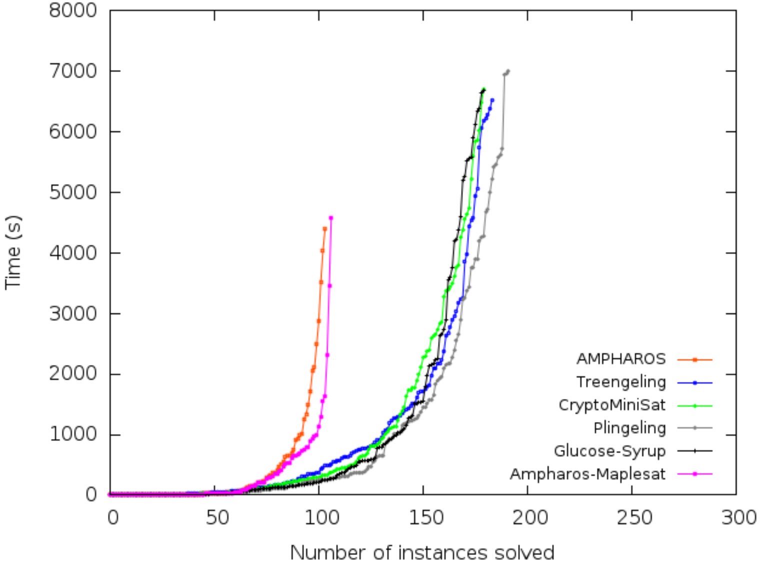
Resutls

SAT 2016 Application Benchmark



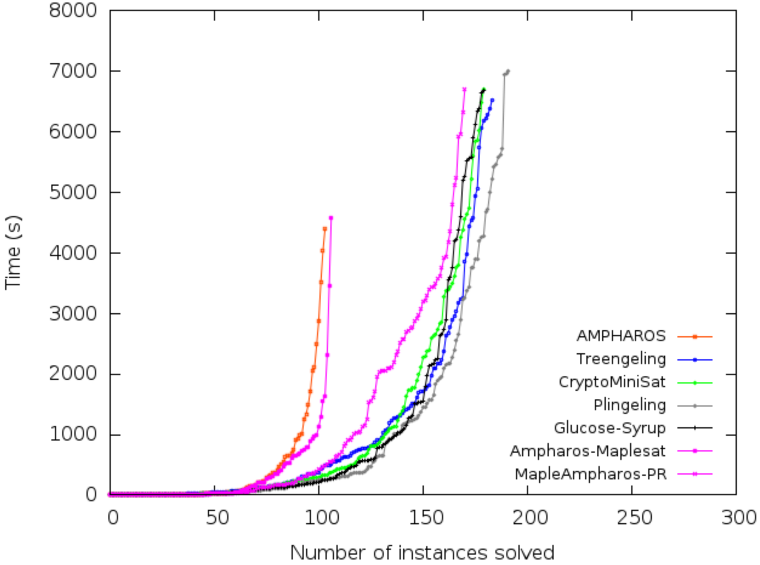
Resutls

SAT 2016 Application Benchmark



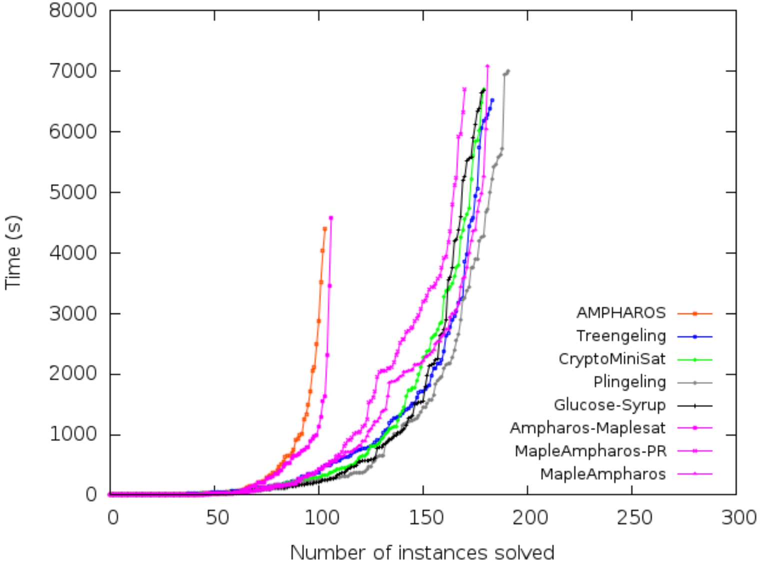
Resutls

SAT 2016 Application Benchmark



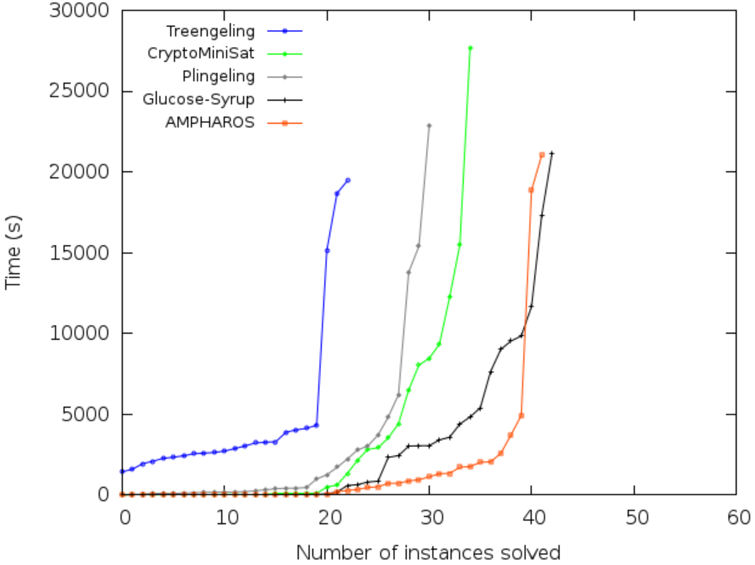
Resutls

SAT 2016 Application Benchmark



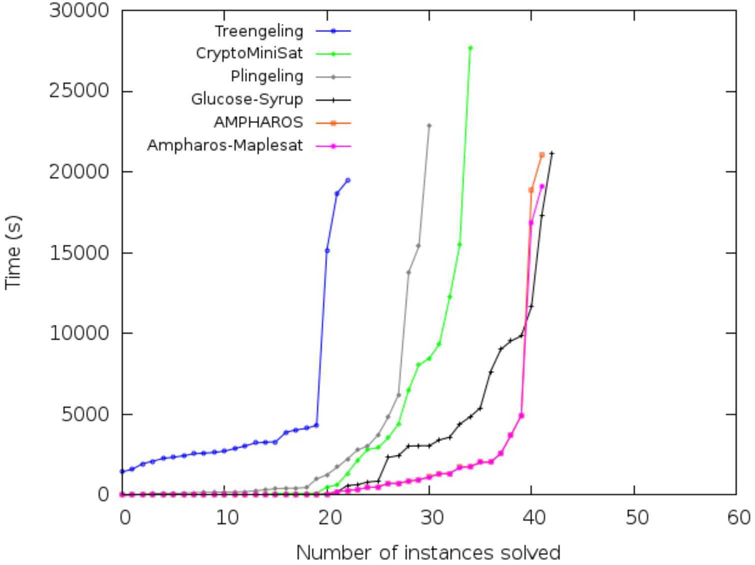
Resutls

SHA-1 preimage Benchmark



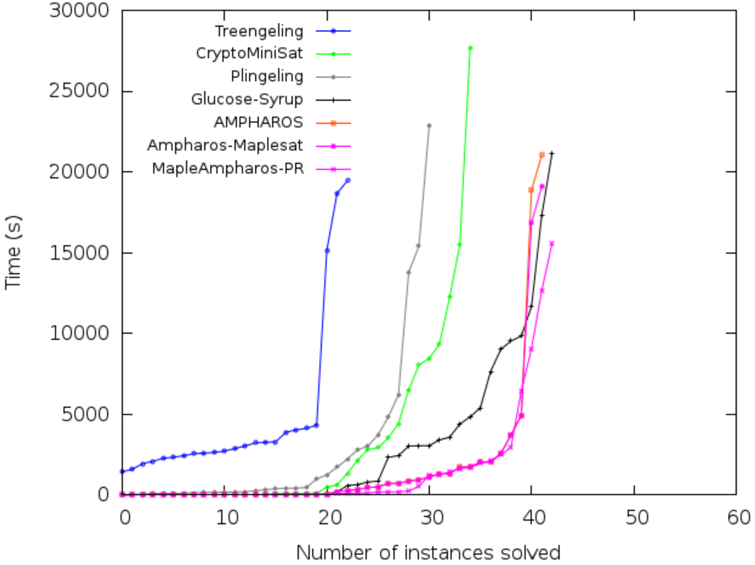
Resutls

SHA-1 preimage Benchmark



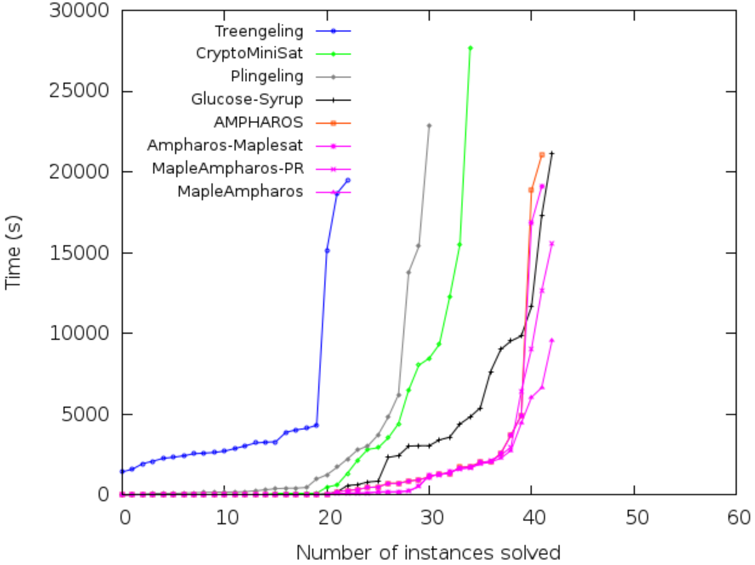
Resutls

SHA-1 preimage Benchmark



Resutls

SHA-1 preimage Benchmark



Conclusion

- An improved version of AMPHAROS competitive to top parallel solvers
- Important role of Splitting heuristic
- Dynamic vs Static metrics (Cheaper guess / Heavier, more accurate!)
- Still not as successful as portfolio solvers, but getting closer!

Conclusion

- An improved version of AMPHAROS competitive to top parallel solvers
- Important role of Splitting heuristic
- Dynamic vs Static metrics (Cheaper guess / Heavier, more accurate!)
- Still not as successful as portfolio solvers, but getting closer!

- More adaptive diversification and splitting

Thank you

Questions?