Case Story: Trust and Cloud

Marijn J.H. Heule

marijn@cmu.edu

Carnegie Mellon University

Formal Methods at Scale, September 25, 2019

“The Largest Math Proof Ever”

engadget

tom's HARDWARE
THE AUTHORITY ON TECH

THE NEW REDDIT

comments other discussions (5)

Mathematics

nature International weekly journal of science

Home | News & Comment | Research | Careers & Jobs | Current Issue | Archive | Audio & Video

Archive > Volume 534 > Issue 7605 > News > Article

Two-hundred-terabyte

19 days ago by CryptoBeer

265 comments share

NATURE | NEWS

Slashdot Stories

Two-hundred-terabyte maths proof is largest ever

Topics: Devices Build Entertainment Technology Open Source Science YRO

Become a fan of Slashdot on Facebook

Computer Generates Largest Math Proof Ever At 200TB of Data (phys.org)

143

Posted by BeauHD on Monday May 30, 2016 @08:10PM from the red-pill-and-blue-pill dept.

THE CONVERSATION

Academic rigour, journalistic flair

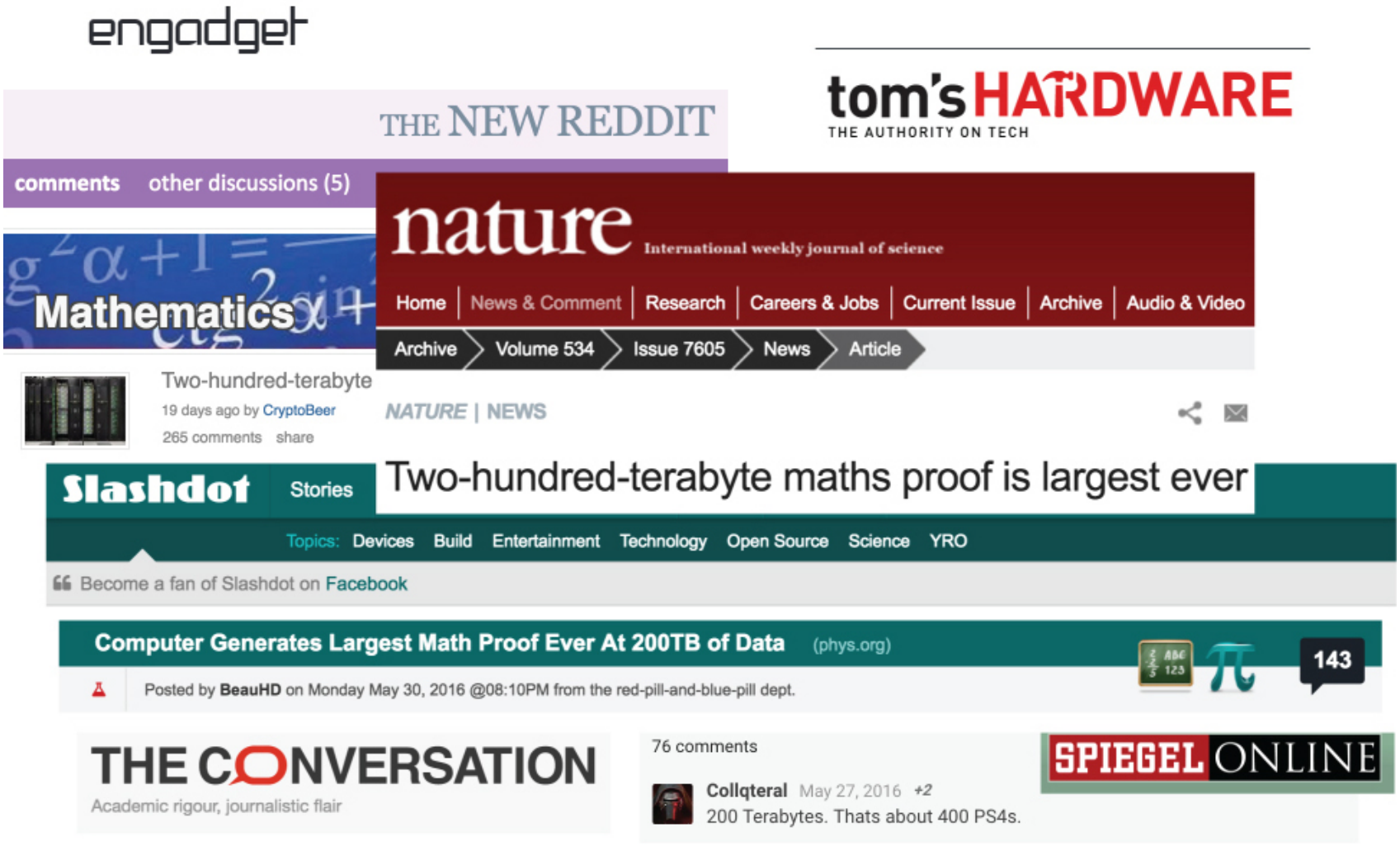
76 comments

SPIEGEL ONLINE

Collqteral May 27, 2016 +2

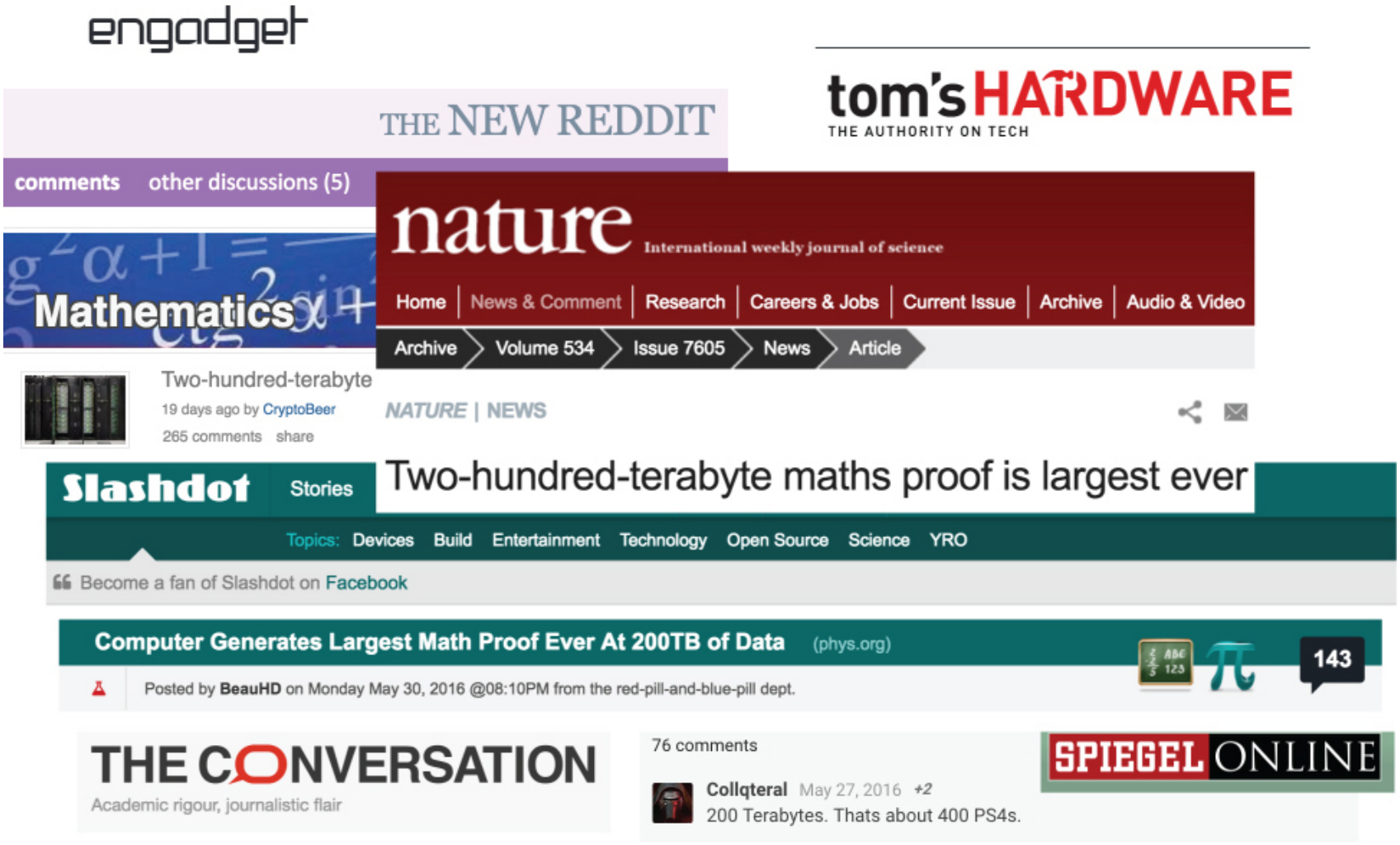
200 Terabytes. Thats about 400 PS4s.

“The Largest Math Proof Ever”



4 CPU years computation, but 2 days on cluster (800 cores)

“The Largest Math Proof Ever”



4 CPU years computation, but 2 days on cluster (800 cores)

200 terabytes proof, but validated with verified checker

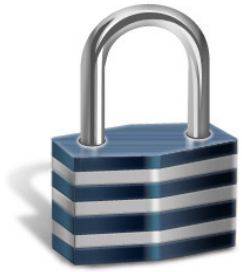
Computer-Aided Mathematics

Fields Medalist Timothy Gowers stated that mathematicians would like to use three kinds of technology [Big Proof 2017]:

- Proof Assistant Technology
  - Prove any lemma that a graduate student can work out
- Proof Search Technology
  - Automatically determine whether a conjecture holds
  - In this talk: Linear speedups on thousands of cores
- Proof Checking Technology
  - Mechanized validation of all details
  - In this talk: Formally verified checking of huge proofs

Automated Reasoning Has Many Applications

formal verification

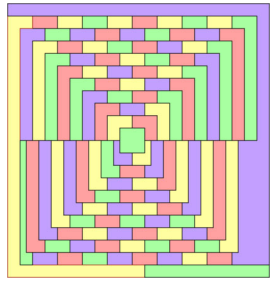


security

bioinformatics

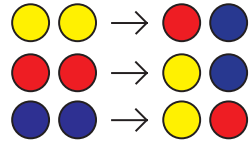
planning and scheduling

train safety



automated theorem proving

exploit generation



term rewriting termination

encode

automated reasoning

decode

Automated Reasoning Has Many Applications

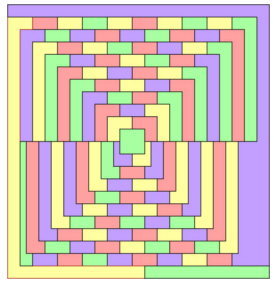
formal verification

security

bioinformatics

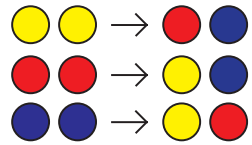
planning and scheduling

train safety



automated theorem proving

exploit generation

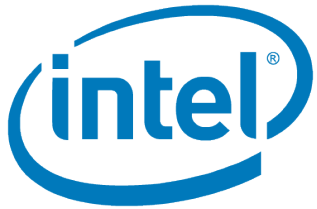


term rewriting termination

encode

automated reasoning

decode

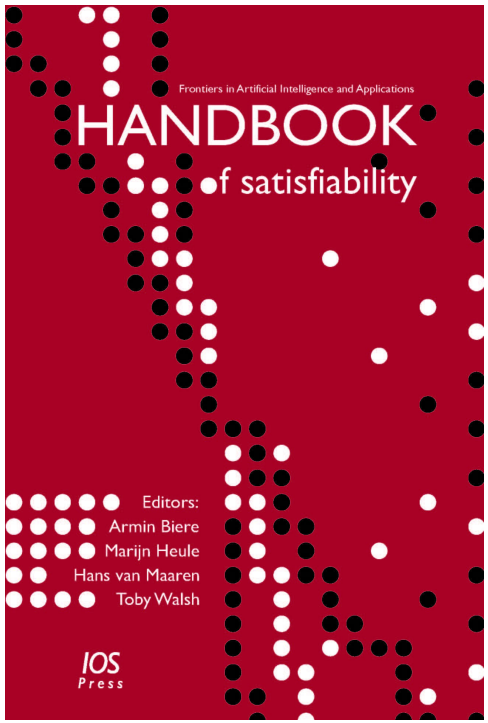


Breakthrough in SAT Solving in the Last 20 Years

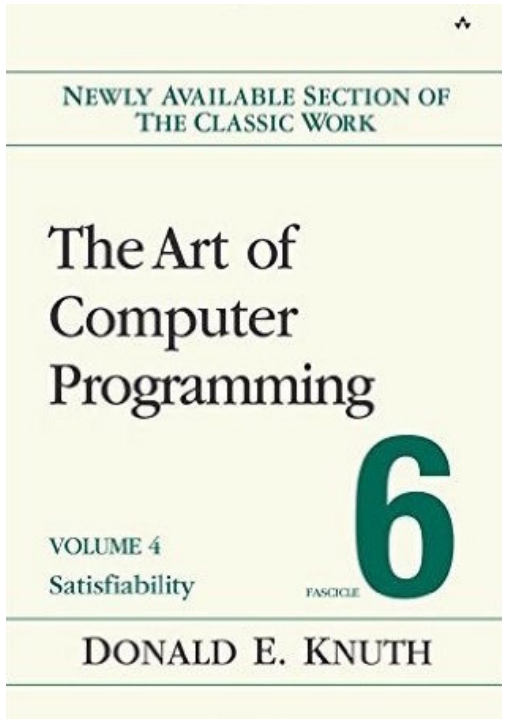
Satisfiability (SAT) problem: Can a Boolean formula be satisfied?

mid '90s: formulas solvable with thousands of variables and clauses

now: formulas solvable with millions of variables and clauses

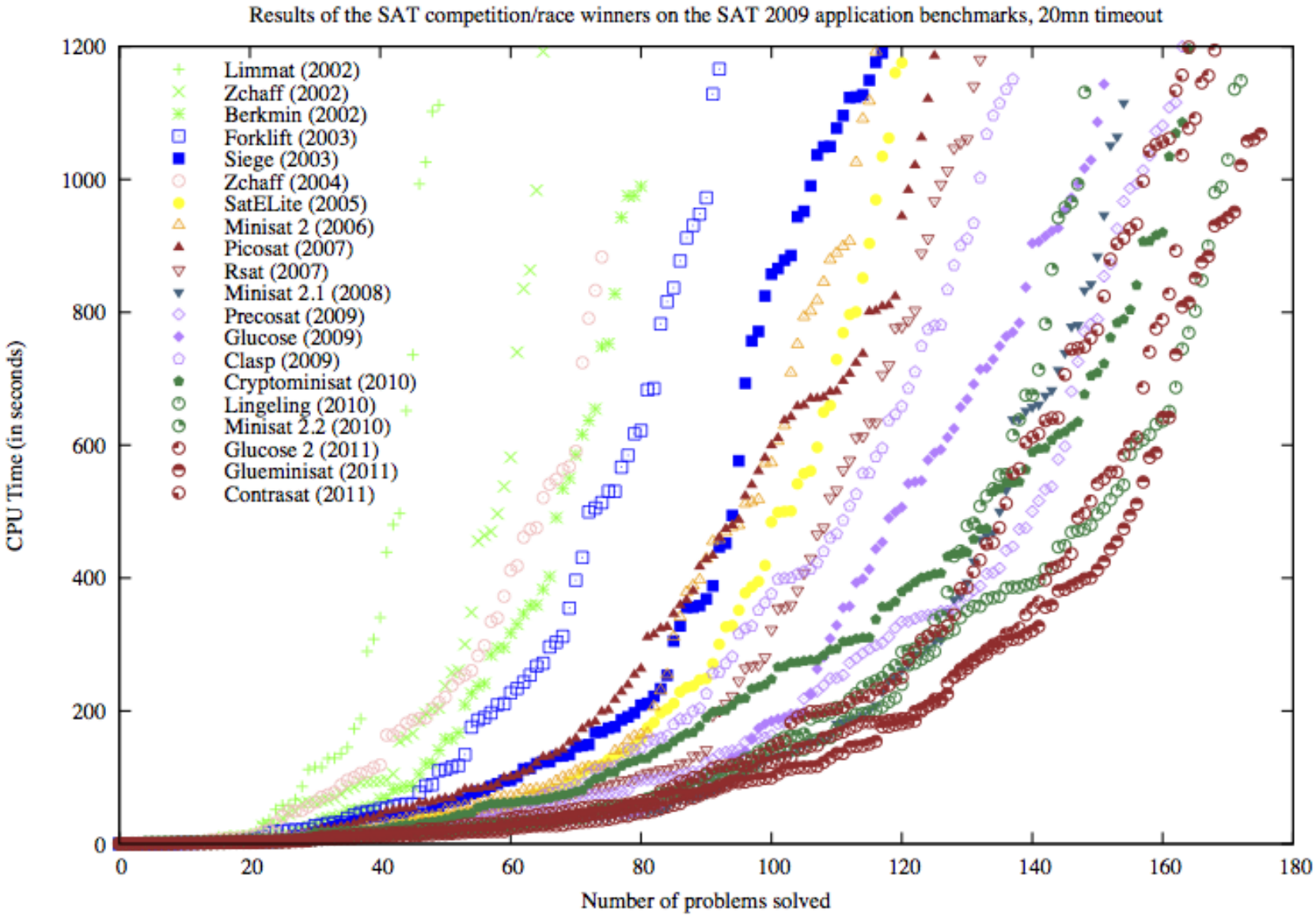


Edmund Clarke: *"a key technology of the 21st century"*
[Biere, Heule, vanMaaren, and Walsh '09]



Donald Knuth: *"evidently a killer app, because it is key to the solution of so many other problems"* [Knuth '15]

Progress of SAT Solvers



Recent Advances at SAT Competitions

Dozens of solvers participate in the annual SAT competition

A new idea contributes to winning the competition

Winner 2017: Clause minimization during search
[Luo, Li, Xiao, Manyá, and Lü 2017]

Winner 2018: Chronological backtracking
[Nadel and Ryvchin 2018]

Winner 2019: Multiple learnt clauses per conflict
[Kochemazov, Zaikin, Kondratiev, and Semenov 2019]

Motivation

Automated reasoning tools may give incorrect answers.

- Documented bugs in SAT, SMT, and QSAT solvers;
  [Brummayer and Biere, 2009; Brummayer et al., 2010]
- Claims of correctness could be due to bugs;
- Misconception that only weak tools are buggy;
- Implementation errors often imply conceptual errors;
- Proofs now mandatory in some competitive events;
- Mathematical results require a stronger justification than a simple yes/no by a tool. Answers must be verifiable.

Verified Solving versus Verified Proofs

Verifying efficient automated reasoning tools is a daunting task:

- Tools are constantly modified and improved; and
- Even top-tier and "experimentally correct" solvers turned out to be buggy. [Järvisalo, Heule, Biere '12]

Verified Solving versus Verified Proofs

Verifying efficient automated reasoning tools is a daunting task:

- Tools are constantly modified and improved; and
- Even top-tier and "experimentally correct" solvers turned out to be buggy. [Järvisalo, Heule, Biere '12]

Various simple solvers can verified, but they lack performance

- DPLL [Shankar and Vaucher '11]
- CDCL [Fleury, Blanchette, Lammich '18]

Verified Solving versus Verified Proofs

Verifying efficient automated reasoning tools is a daunting task:

- Tools are constantly modified and improved; and
- Even top-tier and "experimentally correct" solvers turned out to be buggy. [Järvisalo, Heule, Biere '12]

Various simple solvers can verified, but they lack performance

- DPLL [Shankar and Vaucher '11]
- CDCL [Fleury, Blanchette, Lammich '18]

Validating proof is the more effective approach

- Solving + proof logging + proof verification is much faster compared to running a verified solver
- One verified tool can validate the results of many solvers

Initial Challenges

Theoretical challenges:

- Some "simple" problems have exponentially large proofs in the resolution proof system [Urquhart '87, Buss and Pitassi '98];
- While some dedicated techniques can quickly solve them.

Solution: A proof system to compactly express all techniques.

Initial Challenges

Theoretical challenges:

- Some "simple" problems have exponentially large proofs in the resolution proof system [Urquhart '87, Buss and Pitassi '98];
- While some dedicated techniques can quickly solve them.

Solution: A proof system to compactly express all techniques.

Practical challenges:

- Earlier efforts failed due to complexity and overhead
- Convince developers to support proof logging

Solution:

- The computational burden and complexity is in the checker
- A reference implementation of proof logging

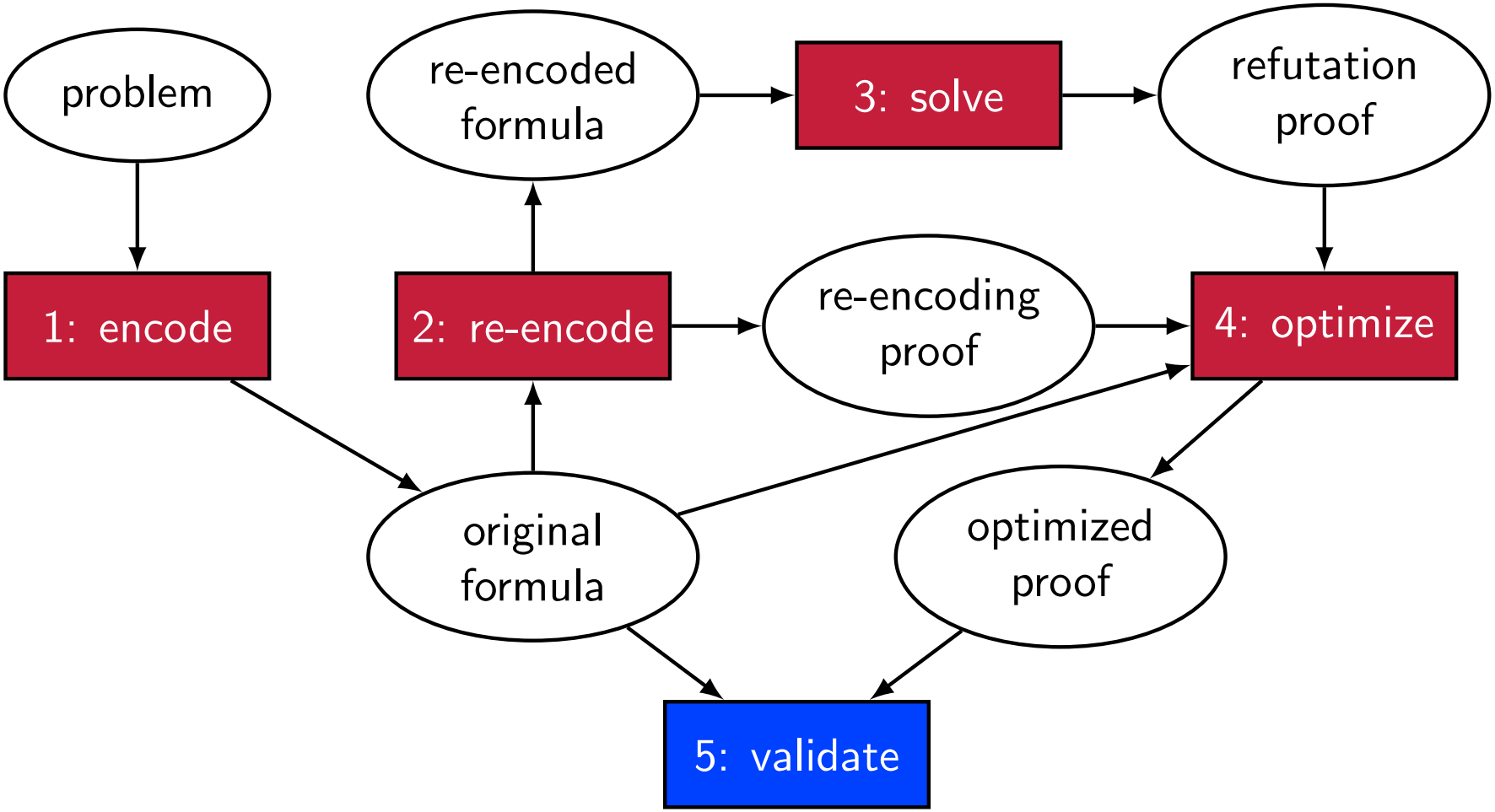
Arbitrarily Complex Solvers

Verified checkers of certificates in strong proof systems:

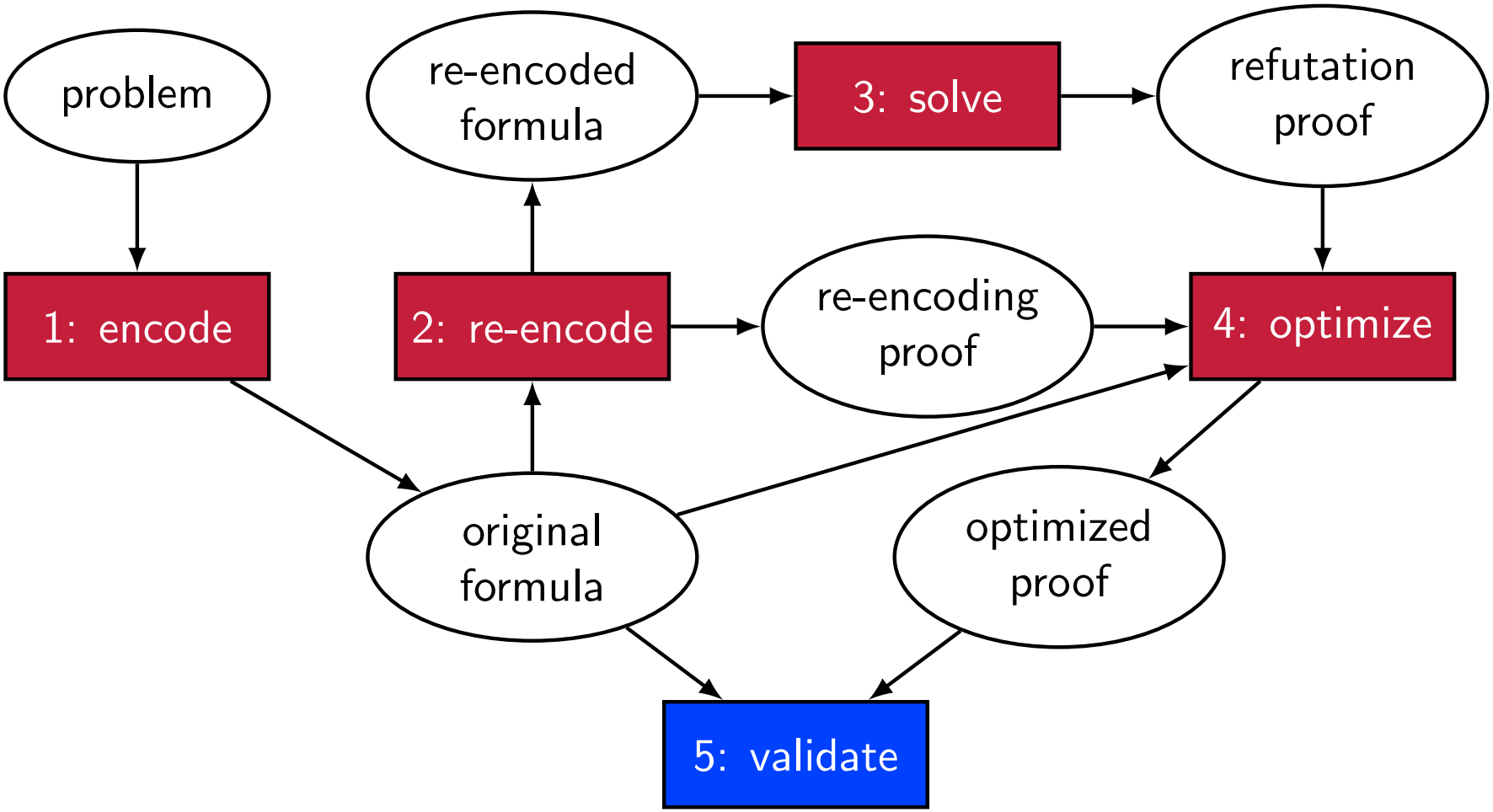
- Don't worry about correctness or completeness of tools;
- Facilitates making tools more complex and efficient; while
- Full confidence in results. [Heule, Hunt, Kaufmann, Wetzler '17]

Formally verified checkers now also used in industry

Formally-Verified SAT Solving Tool Chain



Formally-Verified SAT Solving Tool Chain



- The validate step uses a formally-verified checker;
- Ideally the encoding step is also formally-verified;
- The other steps can be heavily optimized and unverified.

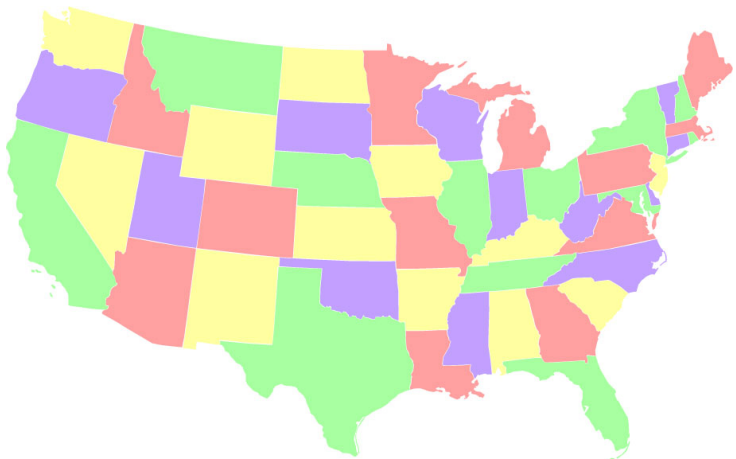
Dealing with Enormous Case Splits

What makes a problem hard?

The numbers angle: how many cases need to be explored?

The Four Color Theorem: Every map is colorable with 4 colors!

- Hard for humans as many cases need to be considered
- Computers can systematically check them all



www.cs.cmu.edu/∼bryant/boolean/US48_colored_balance.jpg

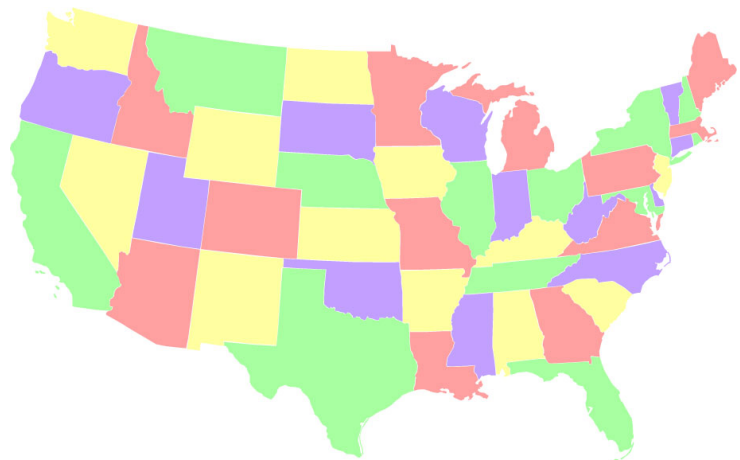
Dealing with Enormous Case Splits

What makes a problem hard?

The numbers angle: how many cases need to be explored?

The Four Color Theorem: Every map is colorable with 4 colors!

- Hard for humans as many cases need to be considered
- Computers can systematically check them all



www.cs.cmu.edu/∼bryant/boolean/US48_colored_balance.jpg

Some hard problems have a trillion or more cases

How to effectively parallelize computations?

SAT Solver Paradigms

Conflict-driven clause learning (CDCL): Makes fast decisions and converts conflicting assignments into learned clauses.

Strength: Effective on large, "easy" formulas.

Weakness: Hard to parallelize.

SAT Solver Paradigms

Conflict-driven clause learning (CDCL): Makes fast decisions and converts conflicting assignments into learned clauses.

Strength: Effective on large, "easy" formulas.

Weakness: Hard to parallelize.

Look-ahead: Aims at finding a small binary search-tree by selecting effective splitting variables via looking ahead.

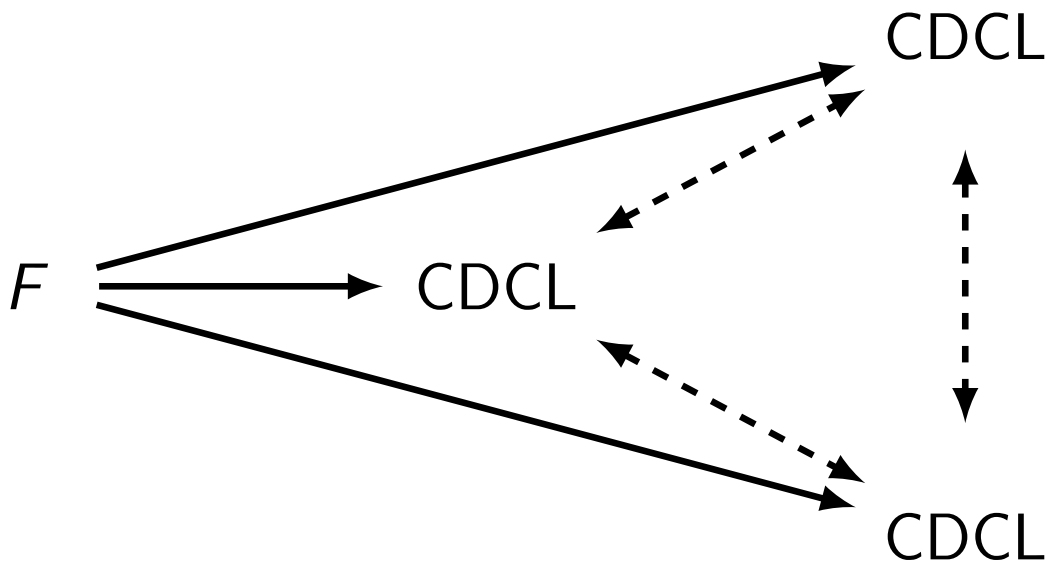
Strength: Effective on small, hard formulas.

Weakness: Expensive.

Portfolio Solvers

The most commonly used parallel solving paradigm is portfolio:

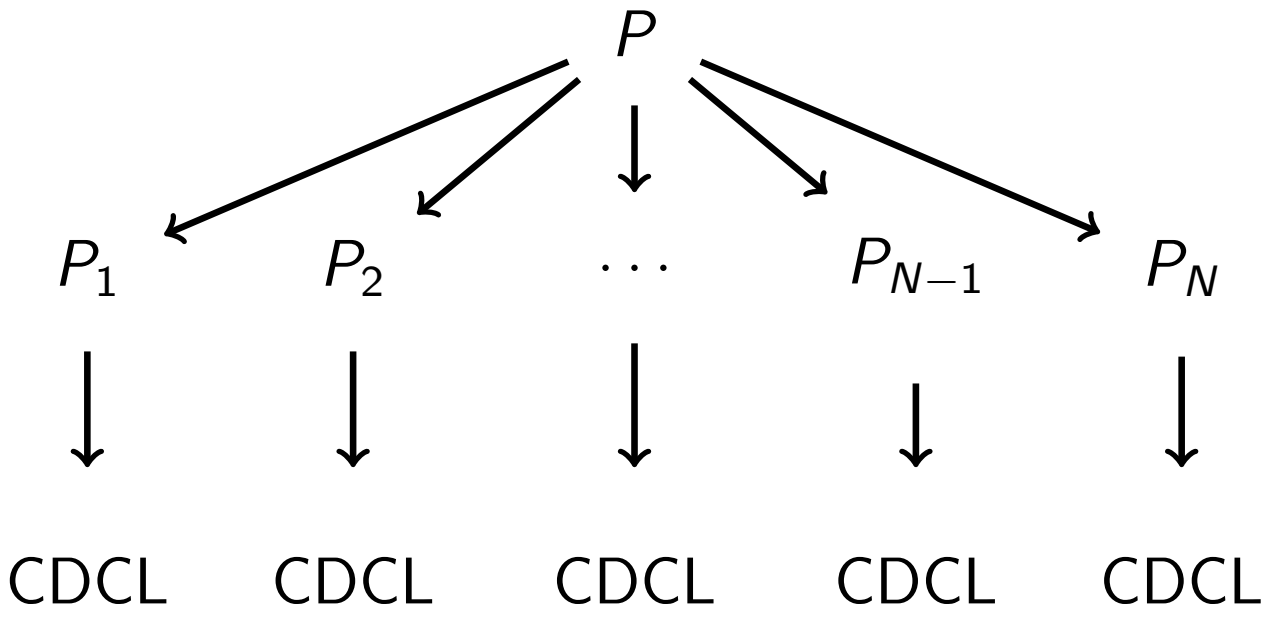
- Run multiple (typically identical) solvers with different configurations on the same formula; and
- Share clauses among the solvers.



The portfolio approach is effective on large "easy" problems, but has difficulties to solve hard problems (out of memory).

Cube-and-Conquer [Heule, Kullmann, Wieringa, and Biere '11]

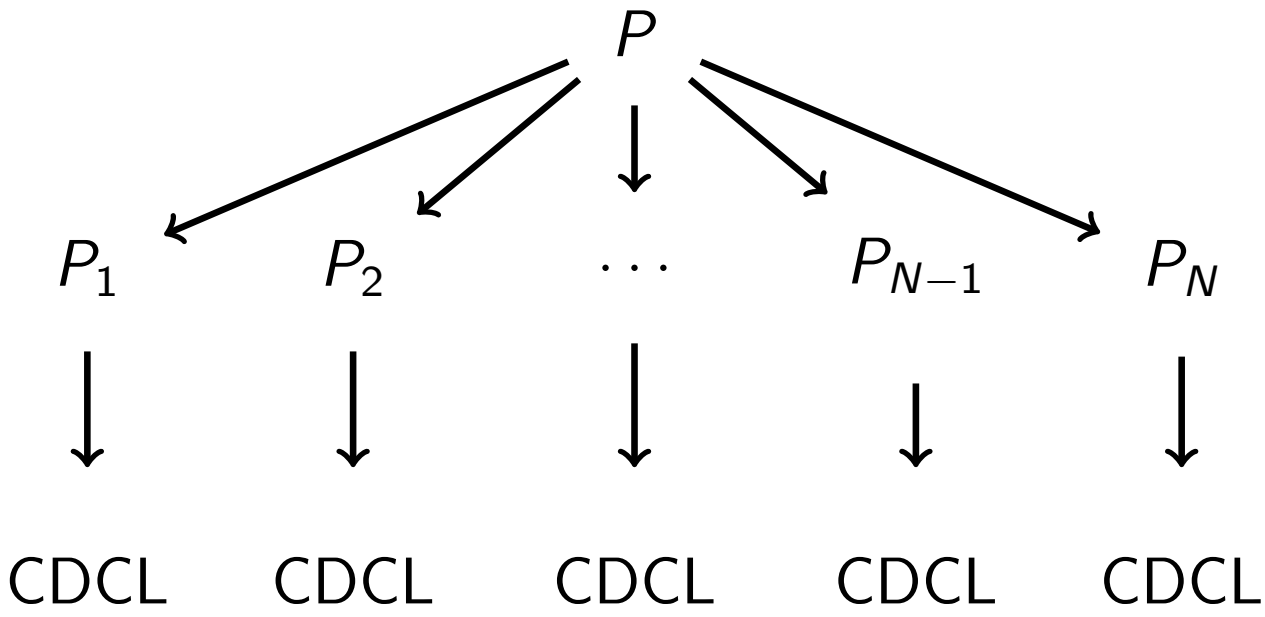
Cube-and-conquer splits a given problem into millions of subproblems that are solved independently by CDCL.



Efficient look-ahead splitting heuristics allow for linear speedups even when using 1000s of cores.

Cube-and-Conquer [Heule, Kullmann, Wieringa, and Biere '11]

Cube-and-conquer splits a given problem into millions of subproblems that are solved independently by CDCL.



Efficient look-ahead splitting heuristics allow for linear speedups even when using 1000s of cores.

Cube-and-conquer recently integrated in Z3

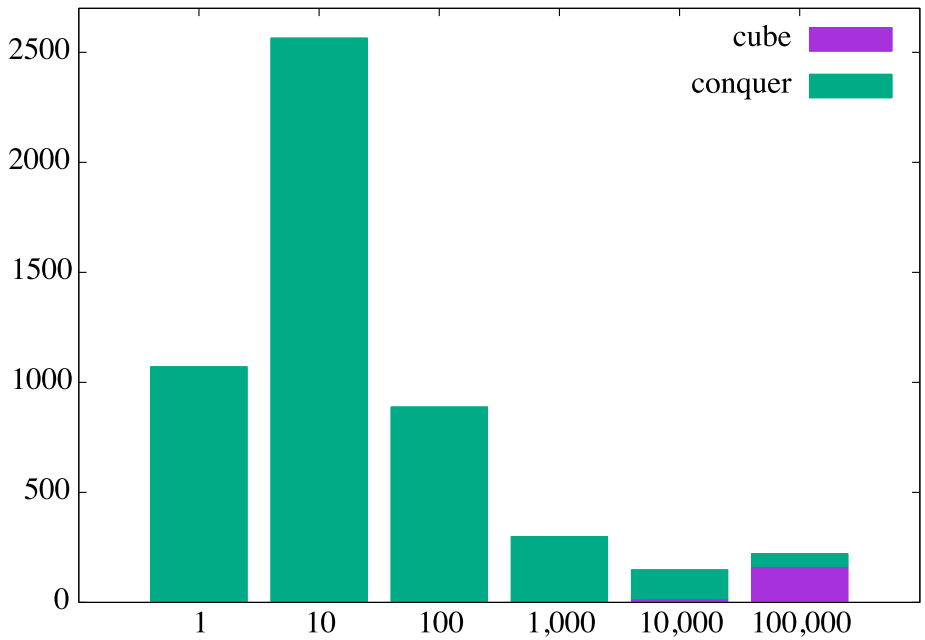
The Hidden Strength of Cube-and-Conquer

Let N denote the number of leaves in the cube-phase:

- the case $N = 1$ means pure CDCL,
- and very large N means pure look-ahead splitting.

Consider the total run-time (y-axis) in dependency on N (x-axis):

- typically, first it increases, then
- it decreases, but only for a large number of subproblems!



Example with Schur Triples and 5 colors: a formula with 708 vars and 22608 clauses.

The performance tends to be optimal when the cube and conquer times are comparable.

Reasoning in the Cloud

Automated reasoning as a service:

- Solves problems from easy to hard;
- Can provide correctness proofs;
- Explains the solution and/or method.

Joint work with Siemens to fully explore the design space of gearboxes.

SIEMENS

The NFL would like to have a cloud service to produce their schedules and they provided interesting test cases.

Introduction

Automated Reasoning and Satisfiability

Trusted Computing

Reasoning in the Cloud

Conclusions and Challenges

Conclusions and Challenges

We can have full confidence in the correctness of SAT solvers:

- All top-tier solvers emit proof logging (also for re-encoding)
- Formally-verified tools can efficiently certify the proofs

How to lift this success to richer logics (SMT/HWMC/FOL)?

Conclusions and Challenges

We can have full confidence in the correctness of SAT solvers:

- All top-tier solvers emit proof logging (also for re-encoding)
- Formally-verified tools can efficiently certify the proofs

How to lift this success to richer logics (SMT/HWMC/FOL)?

Linear speedups are possible on a range of problems

- Even when using 1000s of CPUs;
- And the enormous proofs can be validated in parallel.

Various challenges:

- Make the techniques effective on a broader range of problems
- Expand the potential users: automated reasoning in the cloud
- Explainable automated reasoning to increase understanding

Case Story: Trust and Cloud

Marijn J.H. Heule

marijn@cmu.edu

Carnegie Mellon University

Formal Methods at Scale, September 25, 2019