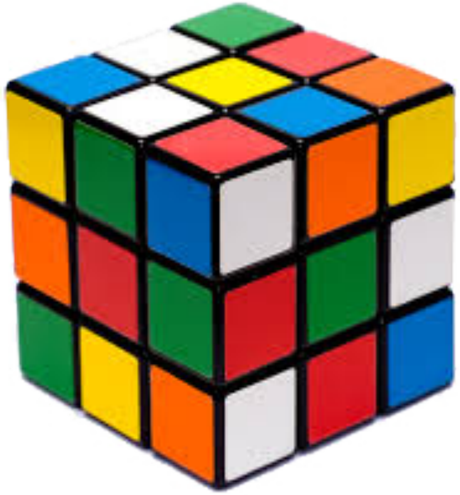
InTime: A Machine Learning Approach for Efficient Selection of FPGA CAD Tool Parameters

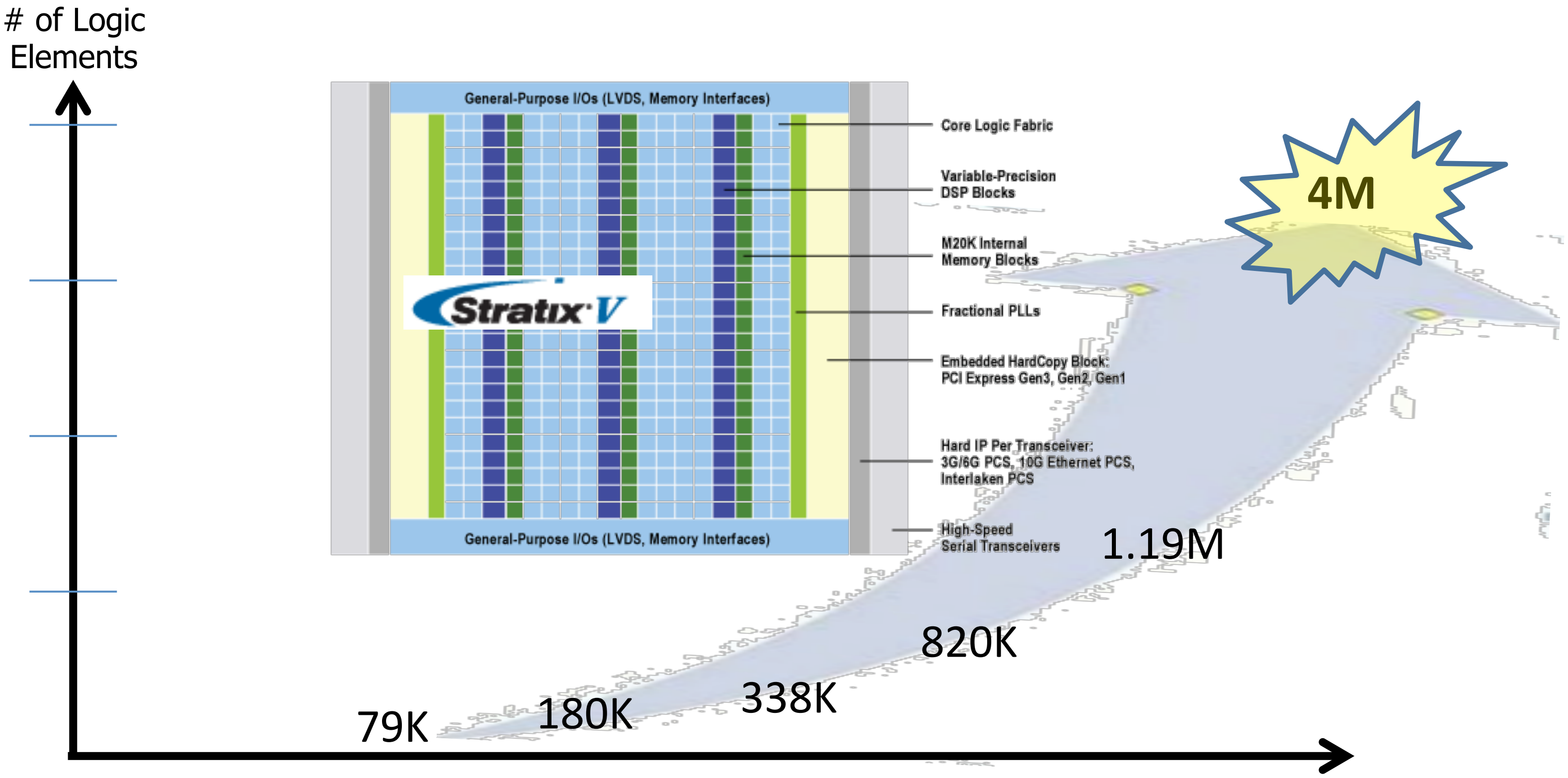
Nachiket Kapre[1], Harnhua Ng[2], Kirvy Teo[2], Jaco Naude[2]

[1]NTU and [2]Plunify, Singapore

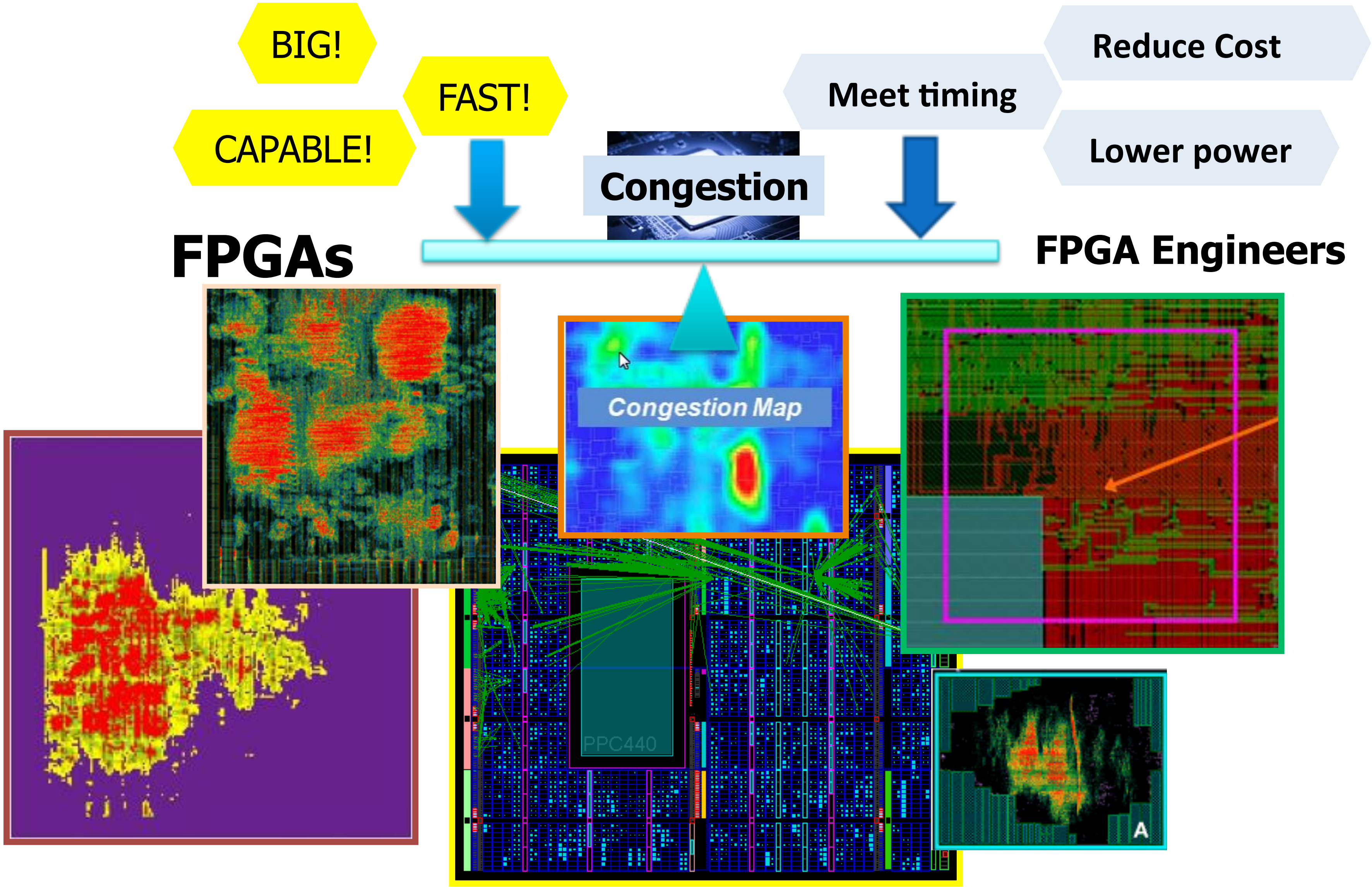
Solve chip design problems using big data and machine learning



BIG! FAST! CAPABLE!



Device Family	Stratix	Stratix II		Stratix III		Stratix IV		Stratix V	Stratix 10
Year of introduction	2002	2004	2006	2008	2010	2013			
Process technology	130 nm		90 nm	65 nm	40 nm	28 nm	14 nm		



BIG!

FAST!

CAPABLE!

FPGAs

Congestion

Meet timing

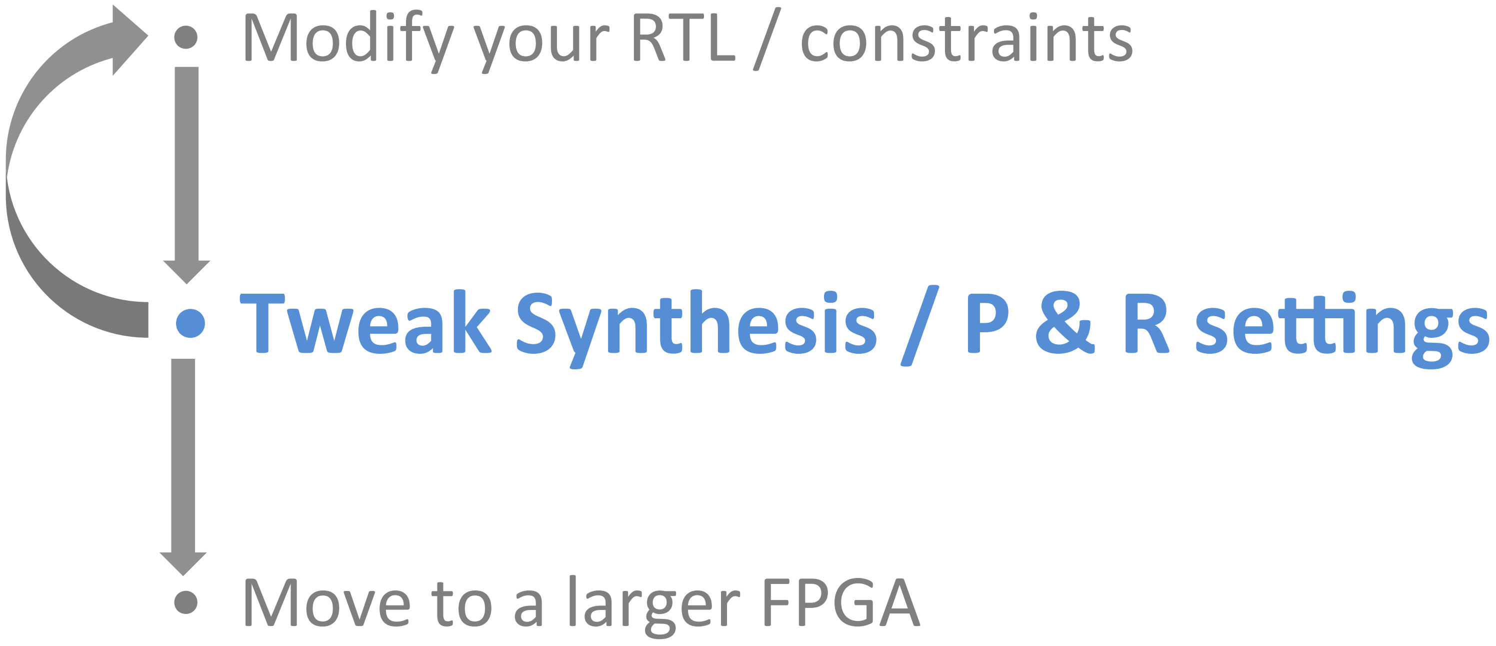
Reduce Cost

Lower power

FPGA Engineers

How to meet timing?

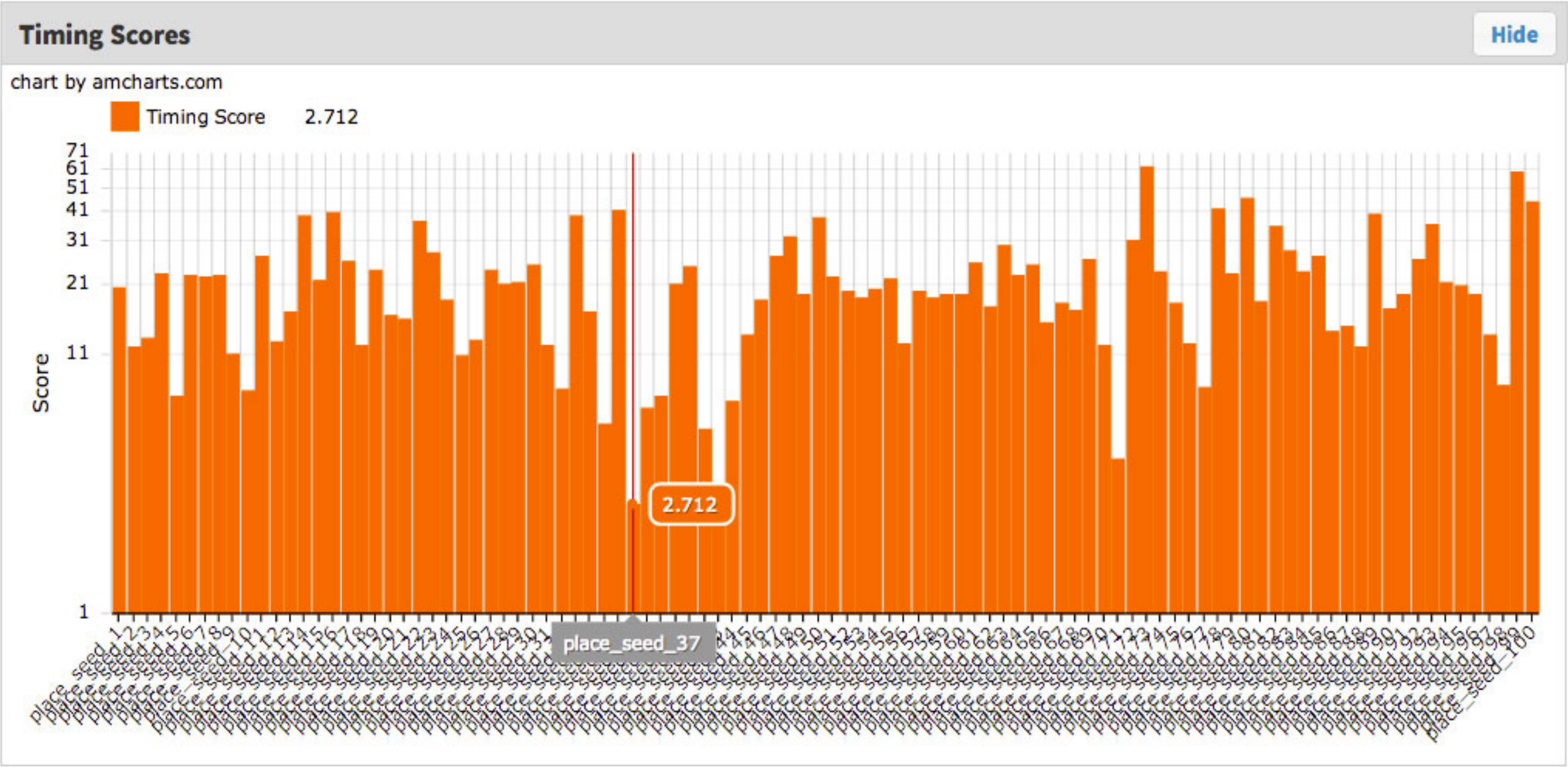
1. **Modify your RTL**
   - Another verification cycle is required
     - IP management risk is increased
   - Is the source code even available?
     - Is the RTL allowed to be modified?
2. **Tweak Synthesis+P&R settings**
   - Which parameters to change?
   - What is a good value to choose for the parameter?
   - Can we even cover all possible cases?



- Modify your RTL / constraints
- **Tweak Synthesis / P & R settings**
- Move to a larger FPGA

PLUNIFY

Placement Seed Exploration



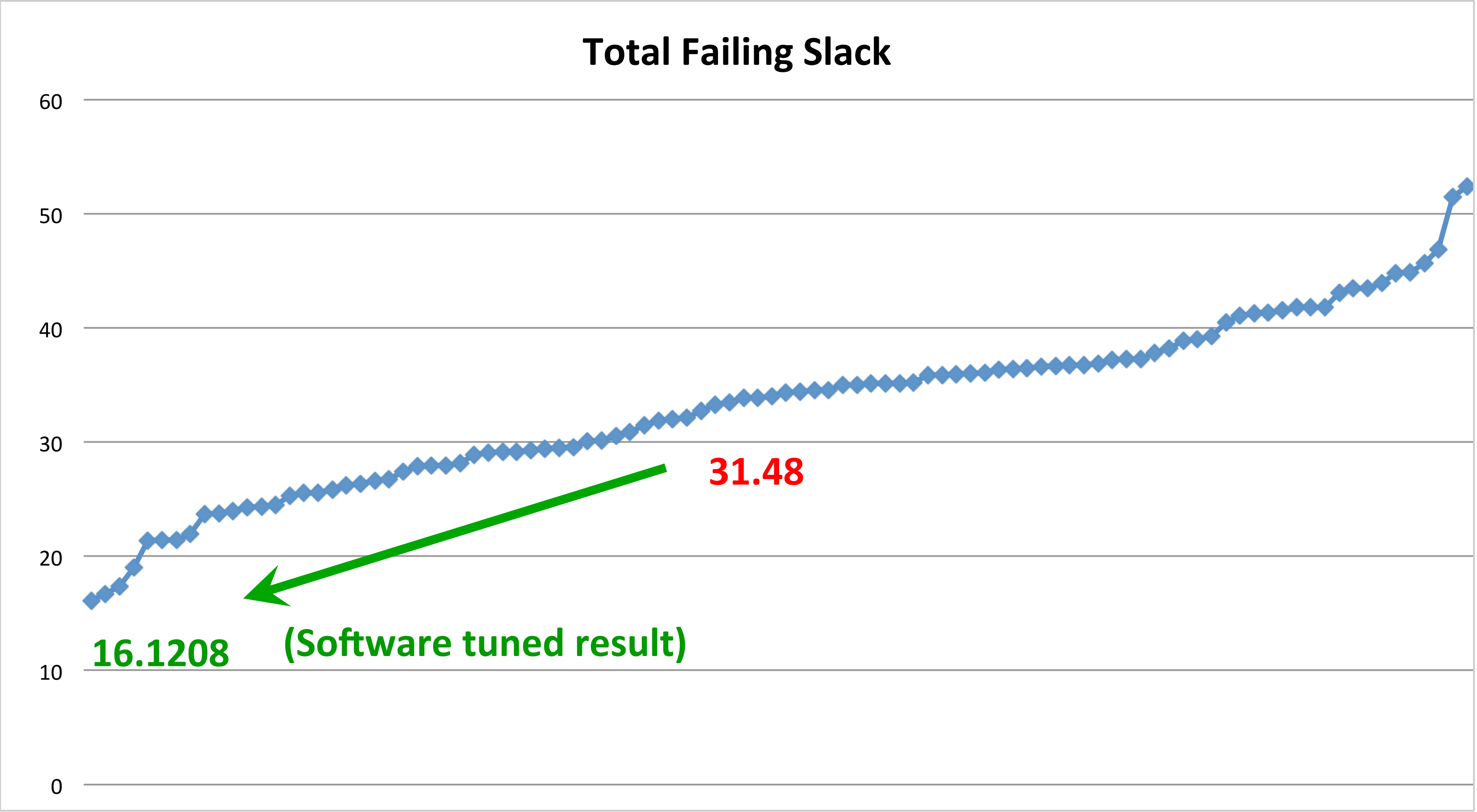
“Seeds” - Blind brute force attempt.

Results must be zero to pass

FAIL!

Intelligent Exploration

Using optimal groups of Synthesis / P & R settings, you can achieve better results.



How do we do it?

1. Cloud Computing
   - Computing costs have fallen dramatically!
     - Google Compute Node → 5—10c/hr
   - Multiple CAD tool runs are completely parallel
     - High machine utilization
2. Machine Learning
   - We automate the “learning” process
   - What tool options work for

     (1) this design,

     (2) this constraint, and

     (3) this FPGA device

Naïve Bayes Classification

Probability of getting a good result,
given that setting CONFIG is set to “X”?

Probability of CONFIG being “X” given a good result

Probability of a good result

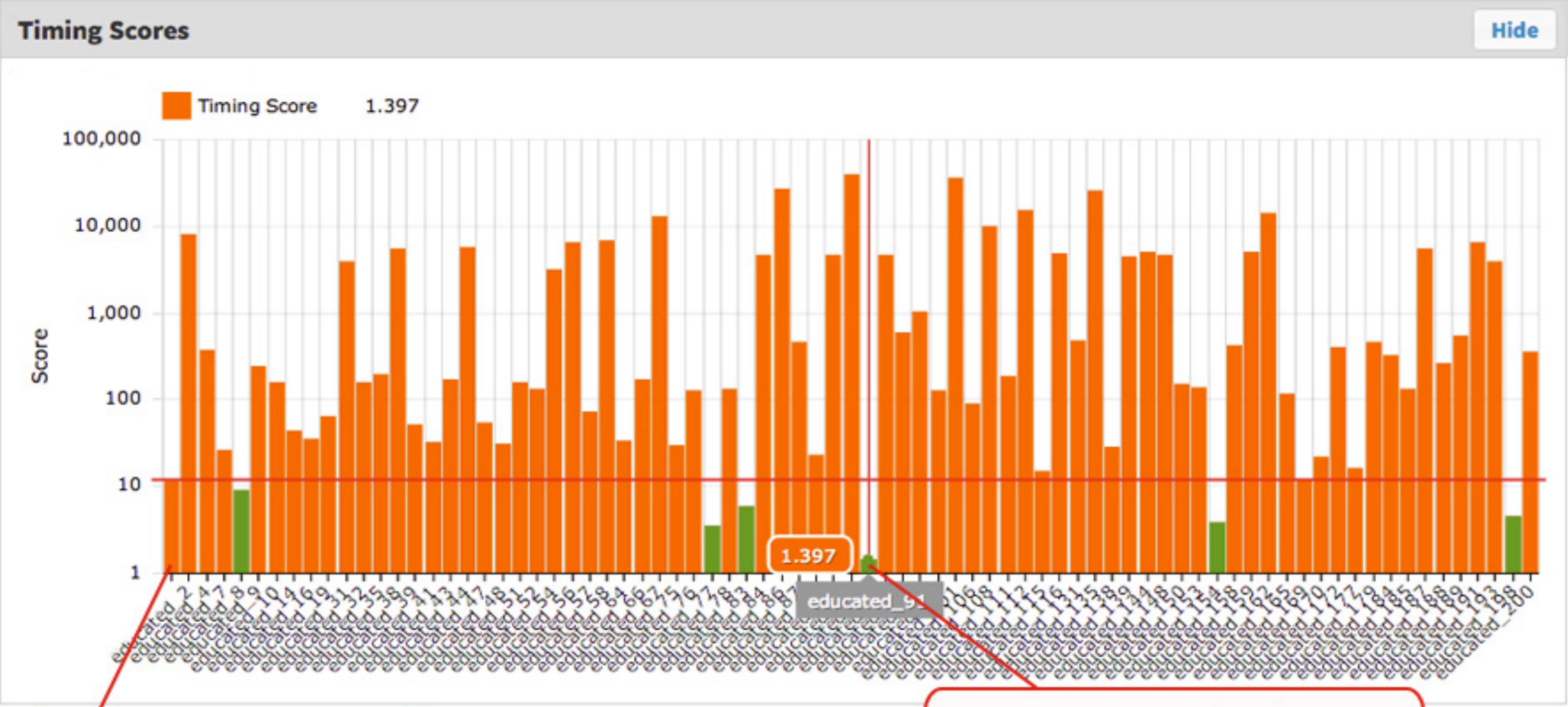
$$p(\text{ good } \mid \text{CONFIG=“X” }) \;=\; \frac{p(\text{CONFIG=“X” } \mid \text{ good})\; p(\text{good})}{p(\text{CONFIG=“X”})}$$

Probability of CONFIG being “X”

Tradeoffs

- **Advantages**
  - Fast to classify
  - Not sensitive to irrelevant features
  - Handles real and discrete data
- **Disadvantages**
  - Assumes feature independence
  - Can perform pre-filtering using techniques such as PCA, etc
- And, there are other ML-based approaches…

Effect of using InTime (**Round 1**)



This is the original result

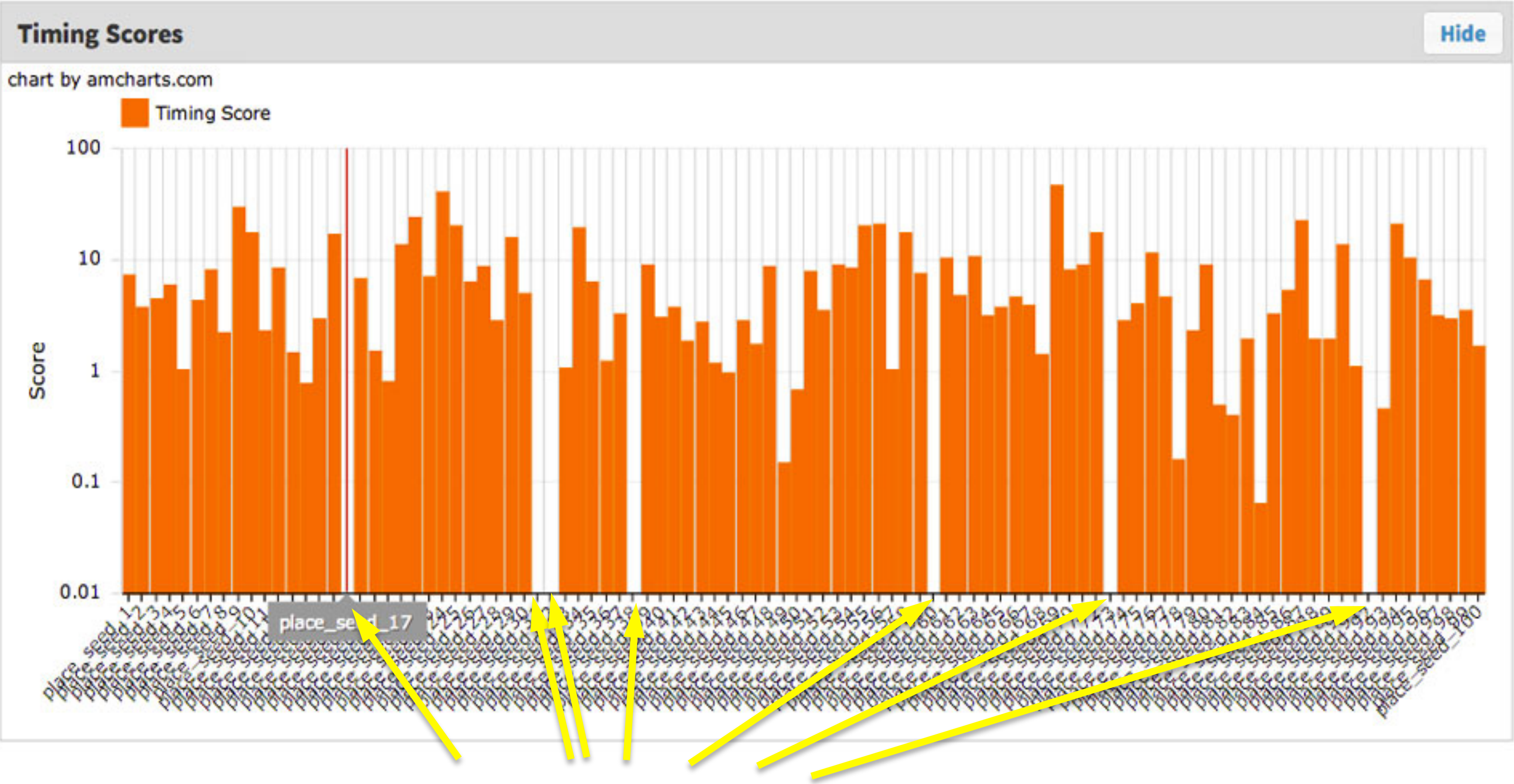
Utilization: **97%**
Timing Score: **11.694**

"Best" result after applying learning

Utilization: **86%**
Timing Score: **1.397**

13% less area
8.3x better timing

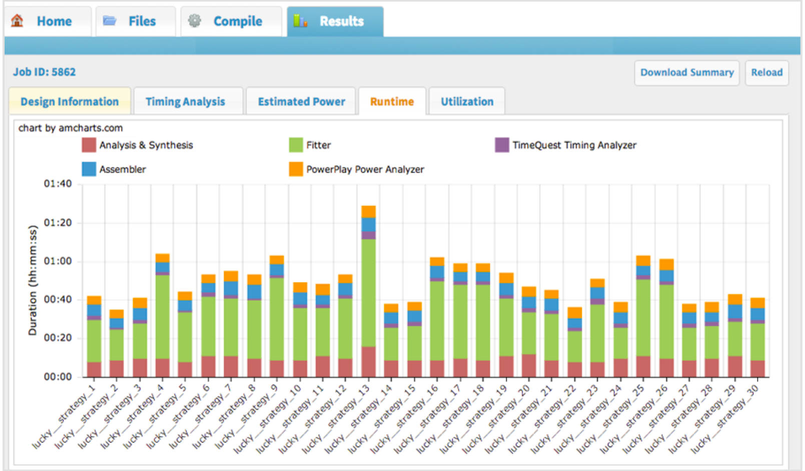
Effect of using InTime (**Round 2**)

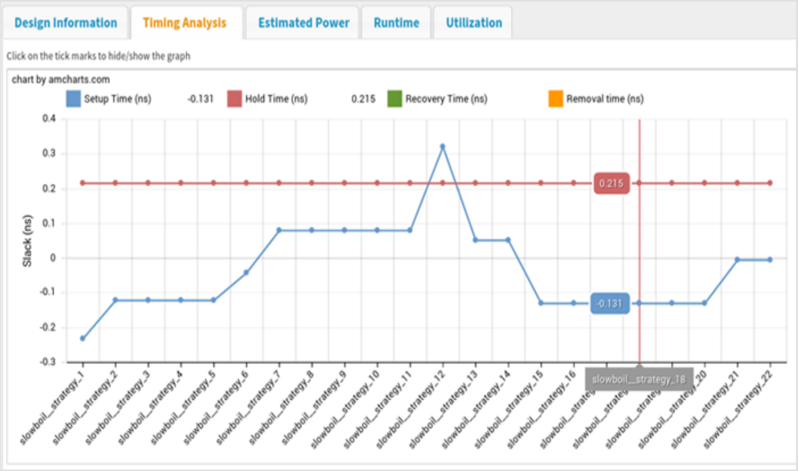


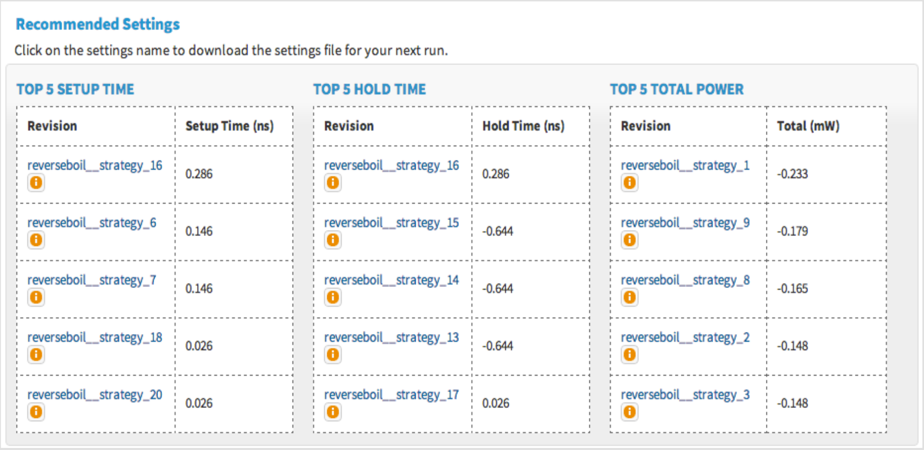
PASS!

7 results!

Other features of InTime







Recommended Settings

Click on the settings name to download the settings file for your next run.

TOP 5 SETUP TIME

Revision	Setup Time (ns)
reverseboil__strategy_16	0.286
reverseboil__strategy_6	0.146
reverseboil__strategy_7	0.146
reverseboil__strategy_18	0.026
reverseboil__strategy_20	0.026

TOP 5 HOLD TIME

Revision	Hold Time (ns)
reverseboil__strategy_16	0.286
reverseboil__strategy_15	-0.644
reverseboil__strategy_14	-0.644
reverseboil__strategy_13	-0.644
reverseboil__strategy_17	0.026

TOP 5 TOTAL POWER

Revision	Total (mW)
reverseboil__strategy_1	-0.233
reverseboil__strategy_9	-0.179
reverseboil__strategy_8	-0.165
reverseboil__strategy_2	-0.148
reverseboil__strategy_3	-0.148

Conclusions

- Meet design goals without modifying the design
  - Changes affect other parts of the system. Minimize changes with data analytics
- Make your FPGA tools work harder
  - Under-utilized features, yearly releases, hard to fully understand all new improvements.
  - Untapped performance gains up to 20%
- People are valuable, machines are cheap
  - Focus on other important issues (or go home early!)