

Enhancing verification throughput in random regression tests with a novel machine learning engine

Davide Sanalitro
Digital Verification Senior Team Leader
STMicroelectronics



SPONSORED BY



Everyday problem

Problem statement

Everyone has many tasks to complete



Challenges

Split the tasks hoping to finish earlier



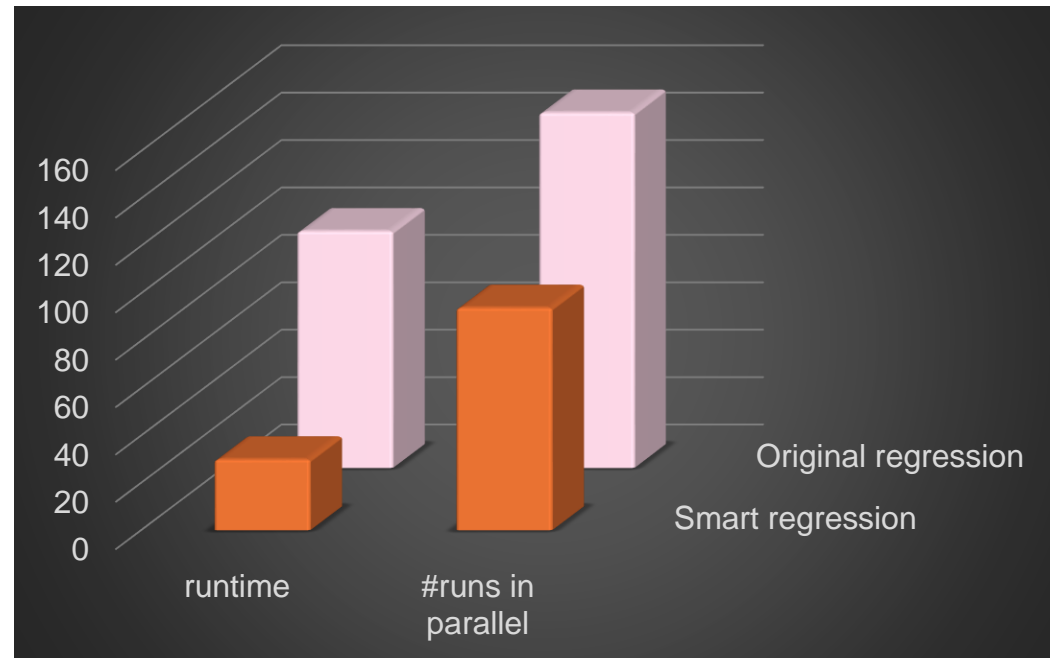
Ideal solution

Complete all tasks in less time and use fewer resources efficiently



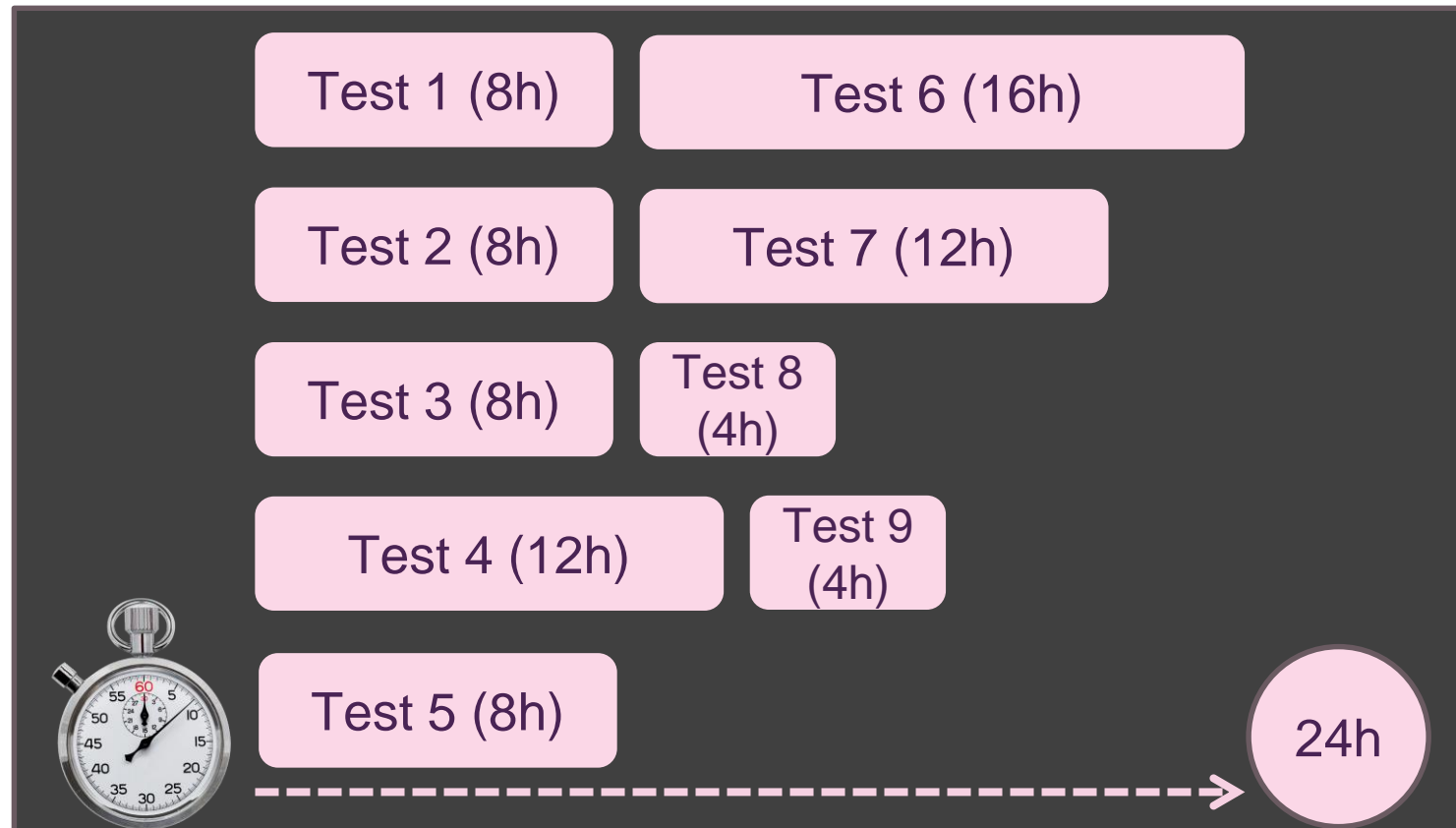
Let's move to digital verification

Every verification flow plans to launch many random tests and regressions to stimulate and verify as many operating scenarios as possible. This leads to very long regression runtimes.



Traditional use – original regression

The verification engineer runs the regression by setting an execution order of random tests and the maximum number of runs in parallel.



A novel machine learning engine

Let's use **Cadence SmartRun**: this machine learning engine analyzes the test duration of the original regression and generates reports containing new policies for the order of the test execution and new parallelism that accelerates the original regression.

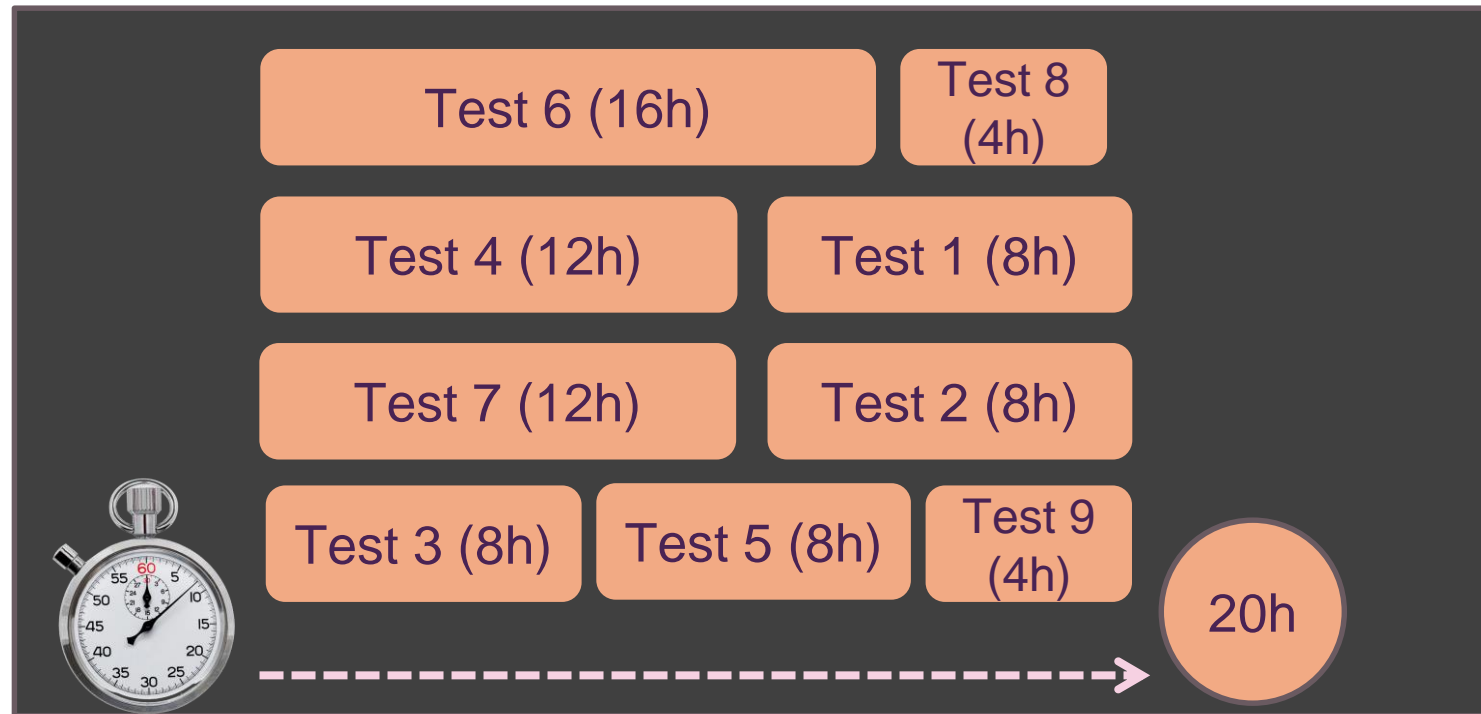
Estimated Time	Queuing policy	Parallel Runs
24h	original order	5
20h	policy 2	5
20h	policy 1	4

The best solution in terms of time and resources used



Traditional use – original regression

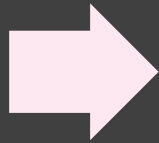
The verification engineer re-runs the regression by setting a new queuing policy and max runs in parallel based on the report generated on the previous step.



How it works

Launch the original regression

- The user runs the regression by setting an execution order of tests and the maximum number of runs in parallel.



Launch **Cadence Smart Run** and generate reports

- It analyzes the test duration of the original regression and generates a report containing new policies for the order of the test execution and new parallelism that accelerates the original regression.

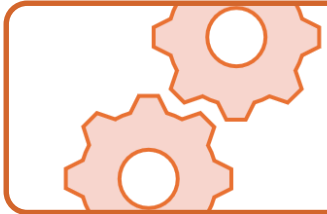


Launch the smart regression

- The user re-runs the regression by setting a new queuing policy and max runs in parallel based on the report generated on the previous step.



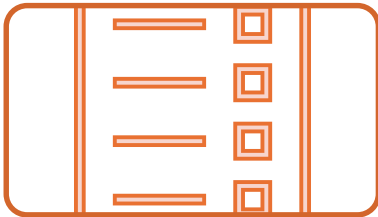
Benefits



Workflow automation: automates several steps required for setting up and running simulations, thereby reducing manual intervention and minimizing errors.

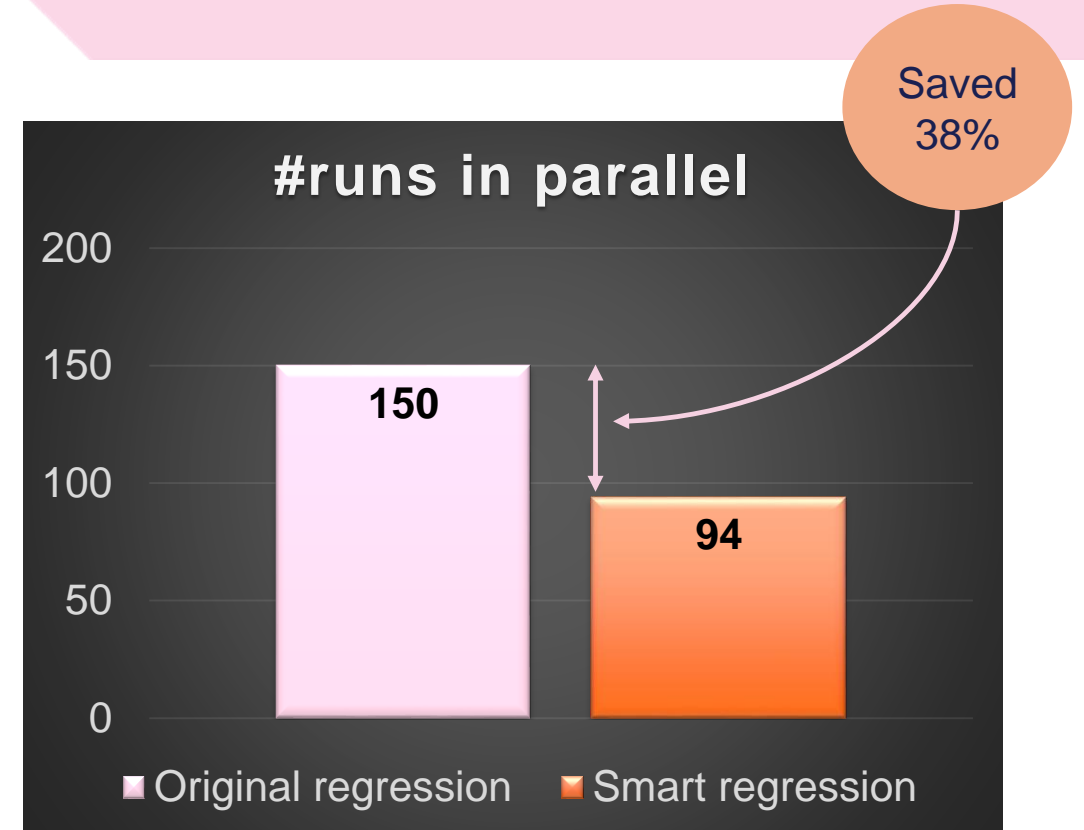
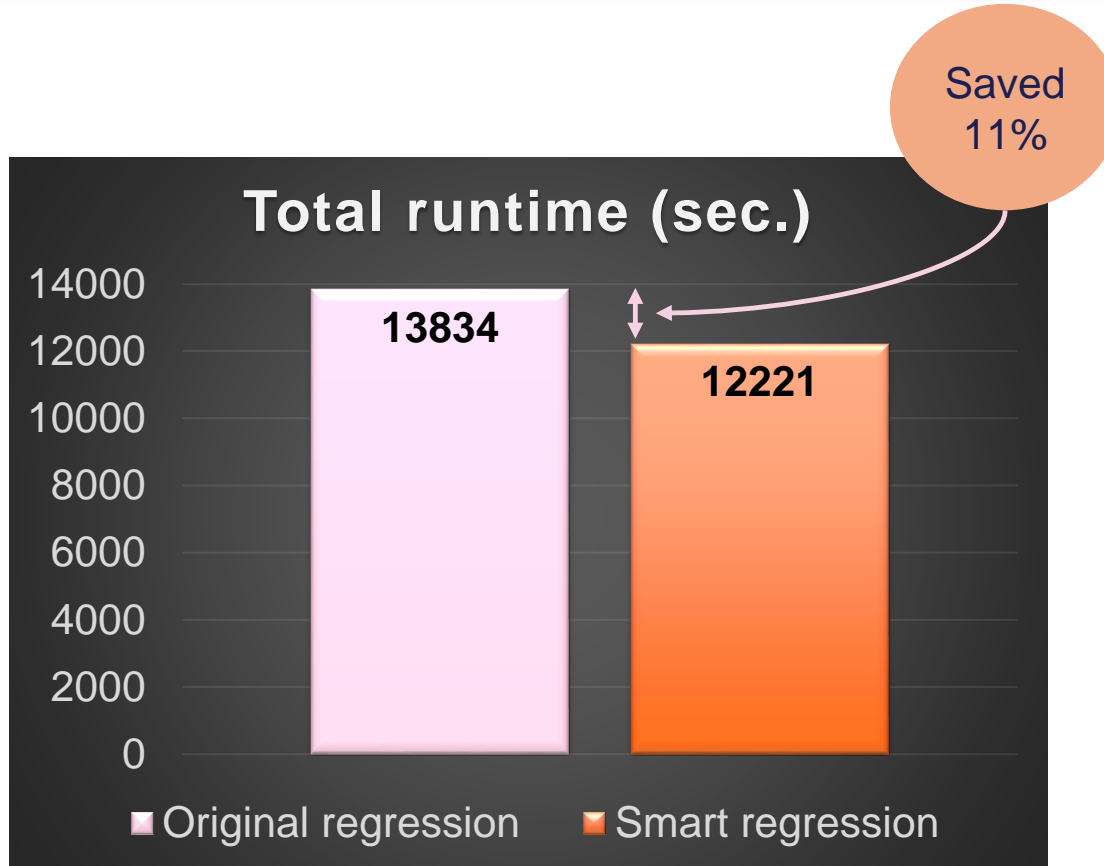


Resource optimization: efficiently manages to compute resources, distributing workloads optimally to speed up verification times.



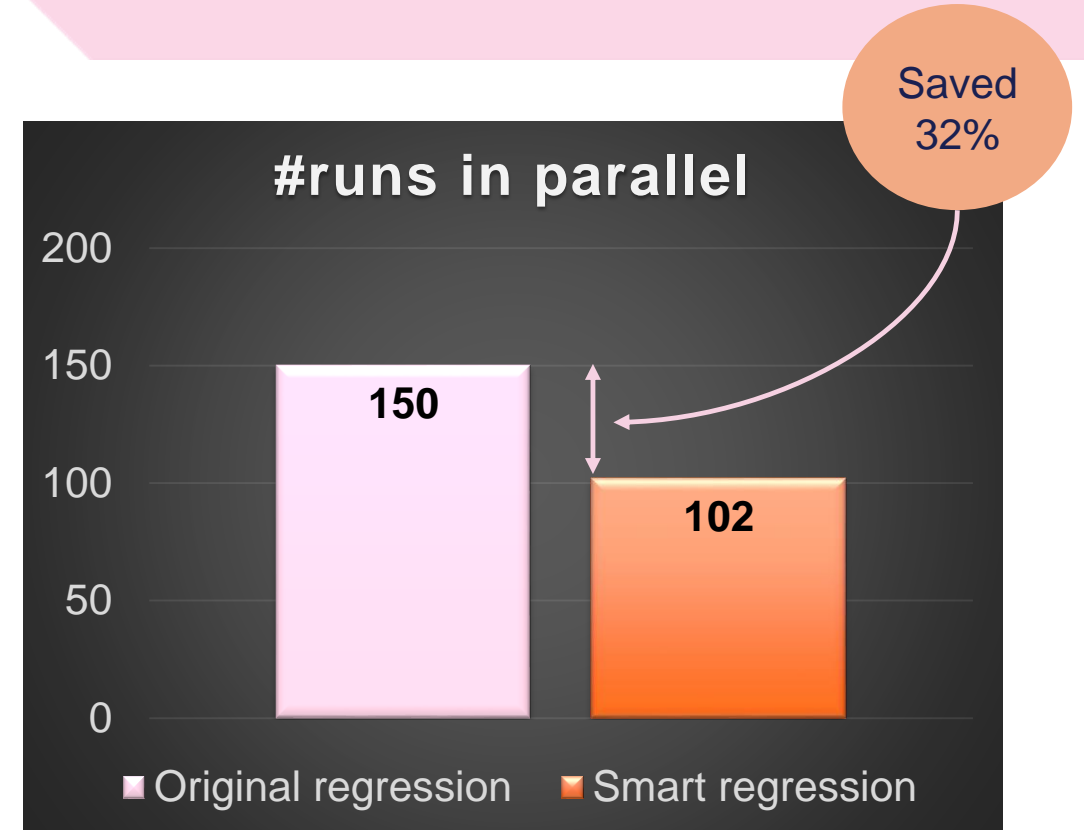
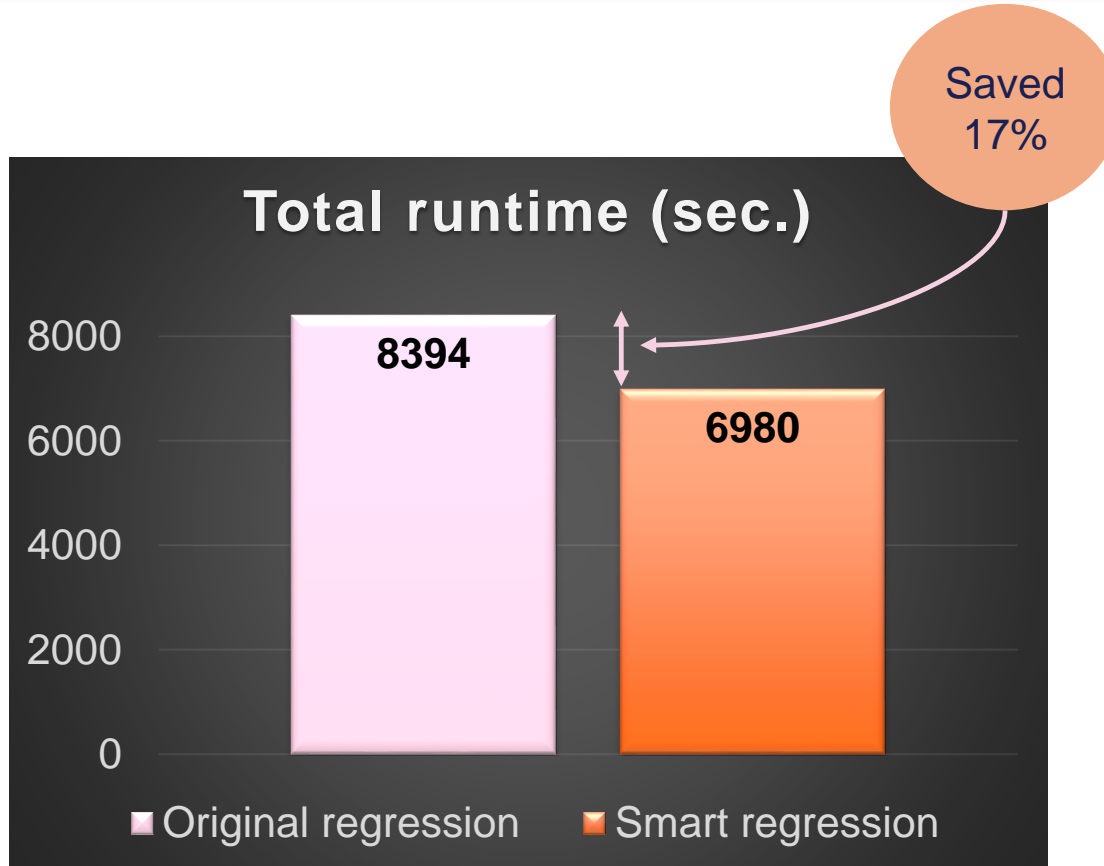
User-friendly interface: provides an intuitive interface that makes it easy to configure and run simulations

Evidence – test case 1



We reduced the runtime by 11% using 38% less resources

Evidence – test case 2



We reduced the runtime by 17% using 32% less resources

Summary

- The advantages:
 - Straightforward to set up and use
 - Save valuable regression time by **up to 17%**
 - Reduce the number of runs in parallel **by 38%**

Reducing the duration of a regression and the number of parallel runs saves time, decreases the number of hosts used and improves time to market.

- For these reasons, we'll use this methodology in all upcoming projects.

