Methodology for Verification Regression Throughput Optimization using Machine Learning

Arun.K.R, Preetham Lakshmikanthan, Ashwani Aggarwal
(Samsung Semiconductor India R&D)
Sundararajan Ananthakrishnan (Cadence Designs Systems, Bangalore)

Problem Statement/Introduction

- Verification engineers taking more time in coverage closure.
- Random nature of test sequences never guarantee 100% coverage in every regression.
- Significant increase in resource usage(servers, simulation tool licenses, manual effort and time)
- Verification Engineer's skillset has a dependency on the regression efficiency.
- Currently no automation to increase the regression efficiency and achieve coverage closure faster.

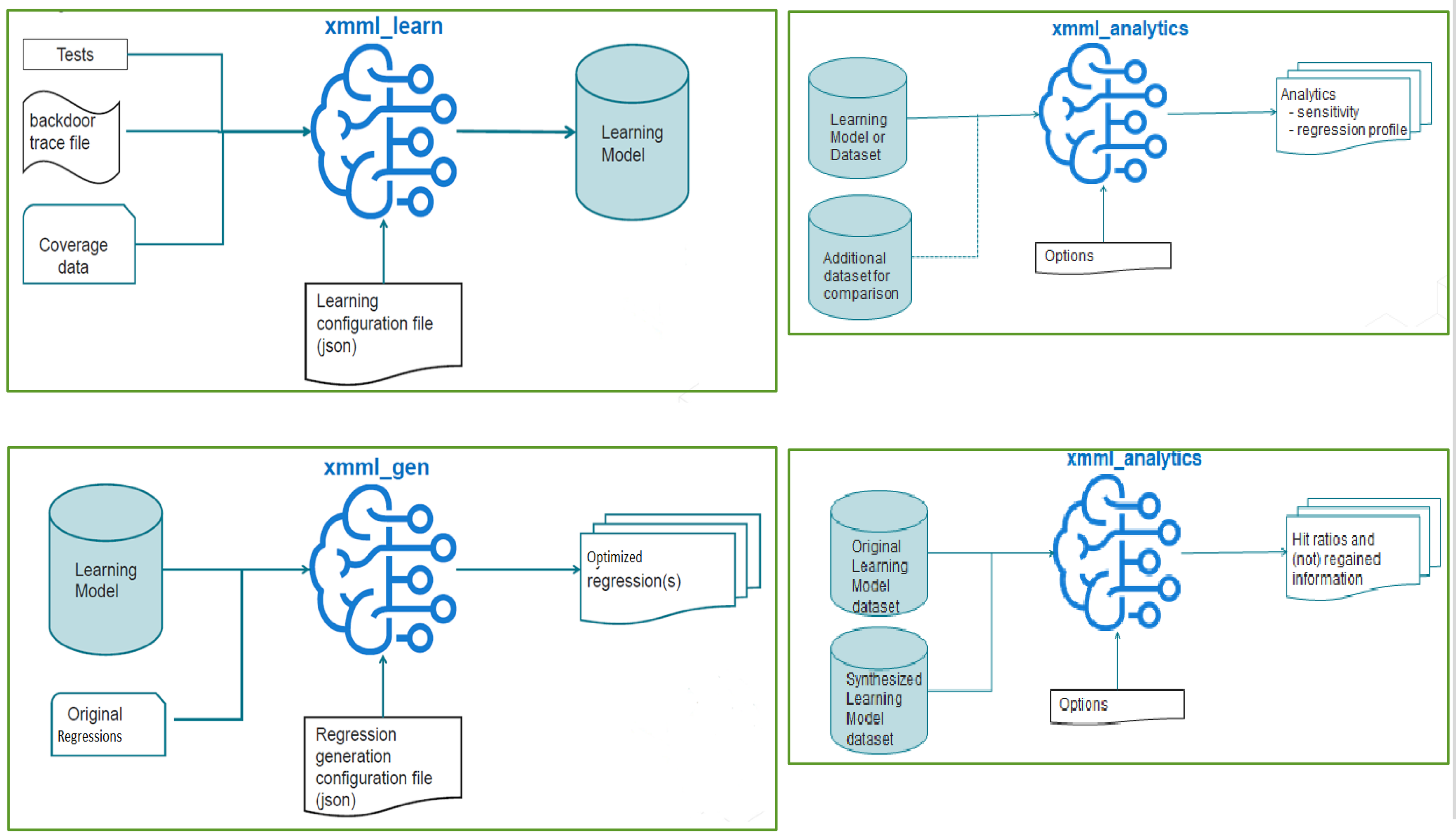
Proposed Methodology/Advantages

This paper introduces a methodology using machine learning with metric driven verification using logic simulation (*ML-MDVLS),* which achieves regression efficiency in a structured way. It uses machine learning to analyse randomly generated test data and also how that data correlates to the coverage goals. Then, ML-MDVLS creates a new highly optimized regression set with special simulator instructions to achieve the desired coverage more efficiently.

Machine Learning (ML) technology and core computational software when integrated into simulation engines, enable orders of magnitude (up to 3X) faster verification closure schedule of the randomized regressions run. Experiments on a live project at Samsung show 3-4X overall regression size optimization, which translates to saving at least 3-5 person-days of effort per regression run for that IP block. Machine learning provides a great deal of potential for finding patterns in verification environments which correlate to coverage goals, and thereby can effectively be used to vastly improve verification efficiency and coverage closure.

Implementation Diagrams

Learning Model building, Generating ML based Optimized regressions, Analytics cmds



Implementation Flow

The ML-MDVLS methodology for optimizing regression efficiency is explained below:

- Run regressions, with the ML interface of the simulator enabled to gather randomization data (variables, test sequences, configuration items etc.) and also the coverage data.
- Build machine learning models
  - Collect data from the regressions and create machine learning models that correlate the random test data to the coverage goals.
- Generate ML based optimized regressions
  - The user invokes an optimized regression generation process which specifies how many random seeds should be assigned to the ML identified tests and simulator instructions that should be applied when executing those tests
- Once the simulator instructions and optimized regression test sets are generated, run new regressions with this optimized test suite and collect coverage statistics.
- Check the optimized run coverage statistics to see whether the complete coverage from the original regression has been re-gained.
- Run analytics (if necessary) on the optimized regression results to get details on the ML regression data.
- Re-generate a focussed (smaller) ML regression test set for the non-regained bins.
- Run this smaller focussed regression set and combine its coverage with that from all the optimized regressions. At this point, it should have regained the coverage when compared to the original regression.
- These new ML generated set of runs are resilient to design and test bench changes and allow the new regression to be used over a period of time before learning needs to be redone.
- As an effect of randomness, there is a very high probability that some new bins (not covered in the original regression) could be covered here, which is an added bonus of this methodology.

Results Table

Optimized results of modem IP after using the ML-MDVLS methodology on a randomized set of regressions.

Regression					Coverage (%)		
Name	# of Runs	Regr. Optim. (%)	Cumulative Sim. Time (Hrs)	Regr. Time Reduction (%)	Functional	Code	
					(Bins hit / Total bins) * 100	Block	Expr.
Original randomized regression	5710	--	1210	--	4447 / 4447 =100%	47928 / 47928 =100%	39958 / 39958 =100%
ML optimized regression	457	5710 /457 =12.49X	130	1210 /130 ~9.3X	4392 / 4447 =98.76%	47910 / 47928 =99.9%	39719 / 39958 =99.4%
ML optimized regressions+ML incremental regressions	1494	5710 /1494 =3.82X	280	1210 /280 ~4.3X	4435 / 4447 =99.73%	47926 / 47928 =99.9%	39876 / 39958 =99.79%
ML Run with additional focus on un-hit bins	1684	5710 /1684 =3.39X	400	1210 /400 ~3.02X	4446 / 4447 =99.97%	47928 / 47928 =100%	39889 / 39958 =99.82%
ML Rerun with focus on remaining corner un-hit bins	1708	5710 /1708 =3.34X	416	1210 /416 ~3X	4447 / 4447 =100%	47928 / 47928 =100%	39889/ 39958 =99.82%

Optimized results of multimedia IP after using the ML-MDVLS methodology. Here the input regressions were ranked regressions

Regression					Coverage (%)		
Name	# of Runs	Regr. Optim. (%)	Cumulative Sim. Time (Hrs)	Regr. Time Reduction (%)	Functional	Code	
					(Bins hit / Total bins) * 100	Block	Expr.
Original ranked regression	1390	--	30	--	2616 / 2616 =100%	8497 / 8497 =100%	15625 / 15625 =100%
ML optimized regression	800	1390/800 =1.73X	11.25	30 /11.25 =2.6X	2616 / 2616 =100%	8384 / 8497 =98.67%	15161 / 15625 =97.03%

Conclusion

The experiments on live IP projects at Samsung showed 3-4X overall regression optimization which translates to saving at least 3-5 person-days of effort per regression run for the Modem IP which originally had a fully random regression. In case of the Multimedia IP which originally had a ranked regression, 1.73X overall regression optimization was observed. Earlier verification signoff of these IPs using the ML-MDVLS methodology enabled faster hand-off to downstream teams thereby shortening the whole IP design cycle – the productivity improvement was tremendous.

REFERENCES