Outline

- Objective
  - AKA “Problem we are trying to solve”
- Motivation
  - Aha, insight
- Background
- Our Methodology
- Case Studies
- Results and Summary
- References
OBJECTIVES
Verification Environment

- DUT, configurations, stimuli, checkers, coverage.
- A test checks a design for different configurations using different stimuli.
- A regression test-suite consists of tests, needed to be run for any changes in design.
  - Categorized like soft, hard, publish, nightly, golden etc.
  - Created using the designer and verifier’s insight.
  - Mostly a static entity e.g. whole publish suite is run to publish any changes in design.
Quality of Verification

- In the absence of tools for complete verification, the quality of a verification environment is measured by
  - Coverage of the design features
    - Code (line, toggle etc.), functional coverage.
  - Checkers
    - Assertions, protocol checkers, coherence checkers etc.
  - Fault simulations
    - Faults insertion, activation, detection.
An optimal regression-suite consists of only the needed tests and nothing less or more. 

– It has to be dynamic because changes in design occur randomly throughout the design.

Can we create a sub-list of the tests related to the changes being submitted?

– Hence save time and other resources like machines, licenses.

– Effectively, reduce the time to market?

©2016 Cavium Networks. All rights reserved.
When RTL is modified, we want to select a subset of the available test-suite that:

- Ensure the coverage of the changes made.
- Ensure the functionality related to the modified RTL is not broken.
- And take minimal resource in terms of run time, licenses, debug time etc.
MOTIVATION
Aha Moment!
Observations/Questions

- Some modules showing similar coverage behavior for the given tests suites.
  - Are these modules functionally related?
- But to qualify this observation we need to answer two questions:
  - Are these patterns a statistical coincidence or these modules are correlated?
    - Needs empirical case studies.
  - If these modules are indeed correlated, can we algorithmically identify them?
    - Machine learning
METHODOLOGY
Plan of Action

- Designs selection
  - Designs with matured test suites.

- Simulations and data collection
  - UCIS to collect coverage data.

- Data analysis
  - R (a data modeling/processing/mining language).

- Data comparison
  - Fault simulations, to compare the quality of dynamically generated suites with original suites.
Data Collection

RTL

Simulations

Coverage DB

UCIS Scripts

Data Frame

Tests
## Coverage DataFrame

<table>
<thead>
<tr>
<th>Module</th>
<th>Test</th>
<th>Time</th>
<th>Line</th>
<th>Toggle</th>
<th>Branch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bisr_ctl</td>
<td>Bar0_d</td>
<td>987</td>
<td>28</td>
<td>14</td>
<td>28</td>
</tr>
<tr>
<td>Bisr_dp</td>
<td>Bar0_d</td>
<td>987</td>
<td>13</td>
<td>914</td>
<td>4</td>
</tr>
<tr>
<td>Arb_rot</td>
<td>Bar0_d</td>
<td>987</td>
<td>5</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>
Data Standardization

Data Frame → Data Standardization & Reduction using R → Reduced Data Frame
Similarity Matrix

| Reduced Data Frame | Data Transformation using R | Similarity Matrix |
A Sample Similarity Matrix
Module Clustering

Similarity Matrix → k-means Clustering → Module Clusters
Regression List Generation

Module Clusters → Modified Files List

Reduced DataFrame → Chosen Clusters

Chosen Clusters → Resource Constraints

→ Regression List
# Tests List for a Cluster

<table>
<thead>
<tr>
<th>Tests</th>
<th>aScore</th>
<th>sd</th>
<th>time</th>
<th>CV</th>
</tr>
</thead>
<tbody>
<tr>
<td>basic_msix</td>
<td>2482</td>
<td>269</td>
<td>1005</td>
<td>75.99</td>
</tr>
<tr>
<td>basic_swip_rsp_err</td>
<td>2482</td>
<td>269</td>
<td>1476</td>
<td>75.99</td>
</tr>
<tr>
<td>basic_prp_sml_qsize</td>
<td>2479</td>
<td>268</td>
<td>1239</td>
<td>75.87</td>
</tr>
<tr>
<td>perf_bw_basic_1_1028</td>
<td>2478</td>
<td>268</td>
<td>2039</td>
<td>75.88</td>
</tr>
<tr>
<td>basic_multi</td>
<td>2475</td>
<td>267</td>
<td>2323</td>
<td>75.75</td>
</tr>
</tbody>
</table>
CASE STUDIES
Experimental Setup

- How to compare the quality of dynamically generated tests list with the statically generated?
  - Verification environment is kept the same.
  - Selection of tests is different.

- A fault simulator called Certitude.
  - Artificially induced faults and a statistical quality measure called ComputeMetric.
    - A sample of faults to get the statistical quality measure within predefined error limits.
Case Study A

- A queue management and submission block.
- 231 modules with 13811 instances.
- Verification environment consists of 45 tests.
- After data reduction phase, we were left with 34 modules that were related to Design A feature set.
Finding Optimal Clustering

![Graph showing Silhouette scores for different numbers of clusters, with a dip around 9 clusters indicating an optimal clustering solution.]
# Study Results

<table>
<thead>
<tr>
<th></th>
<th>Number of Modules</th>
<th>Number of Tests</th>
<th>Activation Score</th>
<th>Propagation Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full-Block</td>
<td>34</td>
<td>45</td>
<td>94.60%</td>
<td>76.02%+-4.54</td>
</tr>
<tr>
<td>Cluster-1</td>
<td>7</td>
<td>10</td>
<td>94.33%</td>
<td>74.32%+-4.44</td>
</tr>
<tr>
<td>Cluster-2</td>
<td>11</td>
<td>10</td>
<td>96.45%</td>
<td>73.55%+- 4.75</td>
</tr>
<tr>
<td>Cluster-3</td>
<td>7</td>
<td>10</td>
<td>94.33%</td>
<td>74.28%+-4.32</td>
</tr>
</tbody>
</table>
Case Study B

- A serial line interface.
- 296 modules.
- Verification environment consist of 112 tests.
- After the data reduction phase, the number of modules left were 92.
Finding Optimal Clustering
## Study Results

<table>
<thead>
<tr>
<th></th>
<th>Number of Modules</th>
<th>Number of Tests</th>
<th>Activation Score</th>
<th>Propagation Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full-Block</td>
<td>92</td>
<td>112</td>
<td>92.31%</td>
<td>86.43%+-4.72</td>
</tr>
<tr>
<td>Cluster-1</td>
<td>18</td>
<td>20</td>
<td>97.30%</td>
<td>85.43%+-4.44</td>
</tr>
<tr>
<td>Cluster-2</td>
<td>8</td>
<td>20</td>
<td>96.45%</td>
<td>83.55%+-4.75</td>
</tr>
<tr>
<td>Cluster-3</td>
<td>11</td>
<td>20</td>
<td>95.43</td>
<td>82.78%+-4.56</td>
</tr>
</tbody>
</table>
RESULTS AND SUMMARY
Results

- Got statistically equivalent results using dynamic clustering methods with
  - Fewer tests,
  - Less cumulative run time,
  - Fewer licenses.

- Most importantly,
  - Saving in terms of human resources as fewer tests to look at.
Summary

- Regression are key to success of a project.
- Regressions’ evolve with the project and may include redundant/overlapped tests that results in wastage of resources like time, money and humans.
- Faster and precise regression creation methods can be helpful.
- We have a shown one method to improve the process. Much more can be done.
Acknowledgements

- Monica Farkash’s pioneering work.
- R language developers and resources.
References


- Farkash, Monica etc., "Mining Coverage Data for Test Set Coverage Efficiency", DVCON 2015, Santa Clara, CA.

- Farkash, Monica etc., "Regression Optimization using Hierarchical Jaccard Similarity and Machine Learning", DAC 2013, Austin, TX.
