Smarter Verification Management with vManager Platform

David Zhang, AE Director, Cadence
Agenda

1. Fundamentals of Smarter Verification Management
2. Next-Generation Architecture
3. Centralized Multi-Geography Collaboration
4. Multi-User, Multi-Engine Concurrent Verification Planning
5. Automation and Customization
Fundamentals of Smart Verification Management

It’s all about predictability, productivity and quality
Better PPQ Verification Management
Across the Flow

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Predictability</td>
<td>• Productivity</td>
</tr>
<tr>
<td>Tracking</td>
<td>Regression Management</td>
</tr>
<tr>
<td>Traceability</td>
<td>Analysis</td>
</tr>
<tr>
<td>Planning</td>
<td>Multi-Engine</td>
</tr>
</tbody>
</table>

Plan → Run → Collect → Triage → Merge → Analyze → React
Productivity: Regression Round Trip

- Prioritize highest impact failure modes
- Identify shortest test for each failure mode
- Automated rerun to generate debug data
Predictability: Traceability

- Requirements traceability
  - Requirement import to seed verification plan creation
  - Link requirements and verification plan
  - Visibility of changes to requirements

- Coverage analysis
  - Multi-engine coverage merge/combine
  - Refinement (unreachability (UNR), UNR crosses)
  - Analysis of coverage vs. plan
Quality: Defined and Agreed Goals and Metrics for Entire Team

- Identify Tests
- Identify Coverage
- Verification Plan
- Link Specs
- Import vPlans

Common Understanding
Verification Management
System Architecture

Scalability, reliability, and ease of admin
Current vManager Architecture

- Minimum requirement for a verification management system
  - Multi-user
  - Centralized database
  - Multiple project support
Next-Generation Architecture

- Load balancing and resilient routing
- Distributed processing for redundancy, scalability, and fault tolerance
- Provide scalable, resilient, reliable replacement for previous single, monolithic system
Distributed Architecture = Consolidation Opportunity

- Reduced maintenance overhead
- Lower IT infrastructure requirements
- Improved reliability and scale
Multi-Geography Verification Management

Utilizing the advanced architecture to enable data centralization
Review – Basic Summary Data Sharing
True Multi-Region Capability

- Distributed, enterprise ready architecture
- Scale across multiple sites and into cloud
- Centralized regression database for collaboration
Multi-Project, Multi-Region Topology

- Single vManager Server
- Central data management across projects
- Aggregated analysis and reporting across geographies
Case Study: vManager实际部署及系统性能

- 项目部署数量: 15+
- vManager server 部署规模数量: 10+
- vPlan更新性能: 实时
- vAPI运行稳定: 全年零故障
- 看板提取数据性能
  - 和实际项目规模强相关
  - 完成一个项目的数据提取需约0.5~1小时
Multi-User, Multi-Engine, Multi-Region Verification Planning

Providing collaboration along with advanced planning features
File-Based Verification Plans

I’m going to check out this vPlan. I need to make some edits.

Darn, the vPlan is locked, I guess I’ll wait.

I’m not waiting, I’m going to work on a copy.

Tim

Mary

Chris
File-Based Verification Plans

Ahh, the vPlan is available, I’m going to take it!

I have more changes to make. I’ll have to wait.

I’m ready to merge my changes. Oh no, my copy doesn’t look anything like the vPlan. I’m going to spend a lot of time merging changes.
File-Based Verification Plans

Tim

Mary

Chris

vPlan locked

Uggh, I’m furious. I’ve been waiting to enter my changes!

vPlan copy

Clip art on this page is licensed under CC BY-SA
Co-Authoring Verification Plans

- Move verification plan storage to the centralized database
- Enable co-authoring for multi-user, multi-region verification plan creation
Verification Plan in Database

- Multi-user co-authoring ability to work on the same vPlan
- History tracking of changes
- Import/export from DB to vPlan file
- Section-level access permissions
Multi-Region vPlan Collaboration

• Single vPlan acts as Verification Contract for diverse teams
• Automatically in sync – no need for problematic cross-site document merge
• Central persistence for connection to external systems (e.g., req. mgmt.)
Case Study: Microsoft TFS 与 vManager 的数据交互

• TFS 负责市场需求和产品需求的管理和跟踪
  - 提供源代码管理、数据收集、报告和项目跟踪
• TFS 里的产品需求点能够自动转换成 vPlan 的验证需求条目
• TFS 里的验证需求点产生变更，则 vPlan 的内容需要进行迭代更新，特别地，需要支持产品需求点被删除、取消、挂起、新建四种状态
• 验证需求完成情况自动翻转 TFS 中产品需求的完成状态
Multi-Engine Case Study: Mobile SoC

Interconnect Workbench
Xcelium™ with UVM
JasperGold® UNR
JasperGold Formal
Palladium®
Perspec™
Protium™

- Manual Tests, Checklists, or Milestone Views
- IP vPlan Results
- System-Level Verification Results
- Specialized Verification Tasks, i.e., Low Power
- Use-Case Testing
- Gate-Level Results
- Embedded SW Coverage
Sim + Formal Combined vPlan

Multi-Engine vPlan Usage Example - http://events.dvcon.org/events/proceedings.aspx?id=234--1
“A Mutually Exclusive Deployment of Formal and Simulation Techniques” - Devarajegowda, Vliegen et al – DVCon Europe 2017
Automation and Customization
Summarizing Regression Data – Tracking

- Automate to easily capture into snapshots the important data from each regression
- Once captured the regression results are no longer needed for tracking, freeing up disk space and capacity
- Summarized snapshot data can easily be charted
What Data Is Interesting to Track?

- Code coverage results for a particular module
- vPlan results for a particular perspective
- Unique failures produced by each run for the entire project
- Who owns the most failures historically
- What are the most common tests in the regression
- Total number of runs, passed and failed
Charting Tracked Data
Quickly spot trends and anomalies

More Tests = More Coverage??

On Schedule??

Converging??

Snapshot’s values
Trend line
Goal line
Customization – vManager API: vAPI

- vAPI implements programmatic access to Verification Management
  - vManager™ database provides centralized data repository
  - Verification-specific abstraction of database content
- Query and mine verification specific data, with automation around tasks like coverage merge
  - “what is the toggle coverage on top.sig1 from all failing tests from last Tuesday”
- Execution of verification management actions on specific datasets
  - “rerun all tests that failed with parity error in the past 5 days”
Failure clustering using Machine Learning

In today’s daily regression failure analysis, the first analysis effort is to go over the failures and identify:

1. Are they new?
2. If they look alike, are they really the same as one found before.
3. First Failure Analysis is not always sufficient:
   1. Error over-generic (Many bugs maps into same group)
   2. Error over-specific (Single bug maps into many groups)

We are applying Machine Learning (ML) techniques by “learning” from the user manual assignments and “predict” future assignments automatically:

- user assigns a failed run to a FC (Failure Cluster)
- The system extract the properties of that run and consider them as characteristics of the FC
- Whenever a run with “similar” characteristics would show up it could be candidate for assignment to the same FC.
Test Weight Optimization

- Identify and adjust the weighting of stimulus targeting a specific verification goal
- Each iteration results are evaluated against goals, effective tests are given more weight in the next iteration

Test 1 with 25 random seeds: 75 runs

Test 2 with 25 random seeds: 75 runs

Test 3 with 25 random seeds: 75 runs

Goals:
- Hit more coverage design
- Hit specific features of the vPlan
- Find more bugs

Iteration 1:
- 10.3% coverage

Iteration 2:
- 22.5% coverage

Iteration 3:
- 57.8% coverage

Higher coverage and effectiveness
Xcelium ML User Flow

Xcelium ML analyzes patterns hidden in verification regression results.
Xcelium ML – User Feedback

I had yet to find *any* user comments about ML being applied to logic simulators for regressions or anything simulation related. That is, until now.

In this year’s report, two early ML users share that **Xcelium-ML got them a 2.5X to 3X speed-up in regression runtimes** -- with comparable coverage compared to their constrained random approach.

"Xcelium-ML helped us generate a 3X smaller regression set while retaining 99+% coverage."

"Xcelium-ML improved our regression runtimes by 2.5X vs. Xcelium."

Wrap Up
Smarter Verification Management

• Better Productivity, Predictability, and Quality need more than ever.

• Planning, Collaboration and Centralization needs continue to expand.

• Multi-User, Multi-Engine, Multi-Region, Multi-Project are not optional.

• Thank you!
Q&A