Regressions in the 21st Century – Tools for Global Surveillance

David Crutchfield, Cypress Semiconductor
Tushar Grupta, Cypress Semiconductor
Frank Roberts, Cypress Semiconductor
Venkataramanan Srinivasan, Cypress Semiconductor
Agenda

• Why we should track metrics and make them available
• Starting point for regression management
• What the solution is:
  – Sources of data
  – Gathering information
  – Data Organization
  – Ruby on Rails
• Example web interface
• Usage
Global Surveillance

- Design complexity is increasing / time-to-market is shrinking
- Verification is often in critical path / status is heavily scrutinized
- Verification spans global sites
  - Products come from IP developed globally
  - Verification teams spread globally
  - Global management of projects
Global Surveillance

What is needed?

• Automation of centralized metrics gathering
• Standardization of status reporting
• Access to status from anywhere
Existing Verification System

• VMS – Verification Management System
• Established a standard approach to:
  – Design and test bench organization
  – Specification of tools arguments
  – Test list creation
  – Regression status / coverage

INPUT

dut.files, tb.files, vms.cfg, test-list, command-line, pre/post scripts

VMS_RUN

OUTPUT

UCDB, HTML, Reports, logs, email, waves, debug, test-lists
Existing Verification System

- VMS manages compilation and simulation jobs through Mentor’s Questa VRM (Verification Run Management)
- VMS does not provide
  - Centralized metrics gathering
  - Global access to regression data and status
  - Standardized trends across metrics

**INPUT**
- dut.files, tb.files, vms.cfg, test-list, command-line, pre/post scripts

**VMS_RUN**

**OUTPUT**
- UCDB, HTML, Reports, logs, email, waves, debug, test-lists
VMS 2.0 Provides

- Centralized database for regression metrics from VMS
- Access to regression data via a web interface
- Project info from other Cypress systems
VMS 2.0 Provides

- Default trend plots with raw data
- Default landing page / dashboard
  - Project level
  - Regression level
  - Test level
VMS 2.0 Provides

• Customizable landing page / dashboard
• Customizable plots
  – Trends
  – Choice of metrics
  – Multiple metrics
Gathering Information

- Sources of information
  - IC Manage (Code churn, config info)
  - Project Management System
  - Defect Tracking System
  - Coverage Database (UCDB)
  - Compute Cluster
Data Organization

• Verification data is stored in three types of objects
  – Projects
  – Regressions
  – Tests
• The database tracks relationships between the different types of objects
• Each project has associated regressions, each regression has associated tests, and each test has associated metrics
Ruby on Rails

• Ruby-based web framework
• Provides all the infrastructure for the web application
  – Database abstraction layer
    ➢ Database entries = Ruby objects
    ➢ Queries = Ruby methods
  – HTTP connection handling
    ➢ Code provides the HTTP response body
    ➢ Rails wraps the response and sends to the client
  – Template engine for HTML/JavaScript
• Allows us to focus on content/business logic
Ruby on Rails

- Implements Model-View-Controller system
  - Model: software representation of a database entry (Ruby object fields map to columns in database table)
  - View: HTML/JavaScript template for a given page
  - Controller: Implements business logic
    Loads models, prepares data, and renders it in a view
User Dashboard

• Displayed when user logs in
• Lists subscribed projects
• Displays project status with links to regression dashboard
• Provides link to defect tracking report
IP Project Dashboard

- Lists all IP projects accessible for subscription
- Provides a subscription link for each project
- Displays project status with links to regression dashboard
- Provides link to defect tracking report

### IP Projects (21)

<table>
<thead>
<tr>
<th>No.</th>
<th>Project</th>
<th>IPS1</th>
<th>IPS2</th>
<th>IPS3</th>
<th>IPS4</th>
<th>CDT</th>
<th>Subscription</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>project 1</td>
<td>1535</td>
<td>1540</td>
<td>1650</td>
<td>1603</td>
<td>Defect Tracking Report</td>
<td>Subscribe</td>
</tr>
<tr>
<td>2</td>
<td>project 2</td>
<td>1840</td>
<td>1850</td>
<td>1802</td>
<td>1610</td>
<td>Defect Tracking Report</td>
<td>Subscribe</td>
</tr>
<tr>
<td>3</td>
<td>project 3</td>
<td>1547</td>
<td>1604</td>
<td>1614</td>
<td>1620</td>
<td>Defect Tracking Report</td>
<td>Subscribe</td>
</tr>
<tr>
<td>4</td>
<td>project 4</td>
<td>NONE</td>
<td>NONE</td>
<td>1550</td>
<td>1601</td>
<td>Defect Tracking Report</td>
<td>Subscribe</td>
</tr>
<tr>
<td>5</td>
<td>project 5</td>
<td>1550</td>
<td>1610</td>
<td>1622</td>
<td>1641</td>
<td>Defect Tracking Report</td>
<td>Subscribe</td>
</tr>
</tbody>
</table>
Chip Project Dashboard

- Lists all chips accessible for subscription
- Provides a subscription link for each chip
- Displays project status with links to IP project dashboard
- Provides link to defect tracking report

### Chip Projects (53)

<table>
<thead>
<tr>
<th>No.</th>
<th>Project</th>
<th>Launch</th>
<th>PR1</th>
<th>PR2</th>
<th>PR3</th>
<th>PR4</th>
<th>PR5</th>
<th>CDT</th>
<th>Subscription</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Chip 1</td>
<td>1551</td>
<td>1511</td>
<td>1512</td>
<td>1519</td>
<td>1553</td>
<td>1553</td>
<td>Defect Tracking Report</td>
<td>Subscribe</td>
</tr>
<tr>
<td>2</td>
<td>Chip 2</td>
<td>1618</td>
<td>1525</td>
<td>1532</td>
<td>1538</td>
<td>1610</td>
<td>1613</td>
<td>Defect Tracking Report</td>
<td>Subscribe</td>
</tr>
<tr>
<td>3</td>
<td>Chip 3</td>
<td>1444</td>
<td>1411</td>
<td>1441</td>
<td>1508</td>
<td>1508</td>
<td>1549</td>
<td>Defect Tracking Report</td>
<td>Subscribe</td>
</tr>
<tr>
<td>4</td>
<td>Chip 4</td>
<td>1422</td>
<td>NONE</td>
<td>NONE</td>
<td>1422</td>
<td>1549</td>
<td>1722</td>
<td>Defect Tracking Report</td>
<td>Subscribe</td>
</tr>
<tr>
<td>5</td>
<td>Chip 5</td>
<td>1433</td>
<td>1445</td>
<td>1502</td>
<td>1534</td>
<td>1550</td>
<td>1609</td>
<td>Defect Tracking Report</td>
<td>Subscribe</td>
</tr>
<tr>
<td>6</td>
<td>Chip 6</td>
<td>1439</td>
<td>NONE</td>
<td>NONE</td>
<td>1449</td>
<td>1618</td>
<td>1618</td>
<td>Defect Tracking Report</td>
<td>Subscribe</td>
</tr>
</tbody>
</table>
Regression Dashboard

- Provides table of default metrics
  - Total coverage
  - Number of tests
  - Design / Test bench churn
  - Run times
  - Host / License utilization

50 regressions associated with this project: project_1

<table>
<thead>
<tr>
<th>Regression Name</th>
<th>Total Coverage</th>
<th>Total Tests</th>
<th>Max Concurrent Jobs</th>
<th>Max Concurrent Licenses</th>
<th>Design Churn</th>
<th>TB Churn</th>
<th>Config Used</th>
<th>Total Regression Time (minutes)</th>
<th>Work Week of Upload</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTL_reg_46</td>
<td>98.65</td>
<td>131</td>
<td>17</td>
<td>17</td>
<td>5.7</td>
<td>20.3</td>
<td>dev_all</td>
<td>106.5</td>
<td>1547</td>
</tr>
<tr>
<td>RTL_reg_47</td>
<td>99.36</td>
<td>133</td>
<td>23</td>
<td>22</td>
<td>11.2</td>
<td>30.5</td>
<td>dev_all</td>
<td>102.7</td>
<td>1548</td>
</tr>
<tr>
<td>RTL_reg_48</td>
<td>99.57</td>
<td>135</td>
<td>19</td>
<td>19</td>
<td>7.8</td>
<td>15.7</td>
<td>dev_all</td>
<td>105.8</td>
<td>1549</td>
</tr>
<tr>
<td>RTL_reg_49</td>
<td>99.80</td>
<td>137</td>
<td>20</td>
<td>20</td>
<td>6.5</td>
<td>10.4</td>
<td>dev_all</td>
<td>107.3</td>
<td>1549</td>
</tr>
<tr>
<td>RTL_reg_50</td>
<td><strong>100.00</strong></td>
<td><strong>140</strong></td>
<td><strong>21</strong></td>
<td><strong>20</strong></td>
<td><strong>3.1</strong></td>
<td>5.2</td>
<td>dev_all</td>
<td><strong>108.2</strong></td>
<td><strong>1550</strong></td>
</tr>
</tbody>
</table>
Trend plots

- Track coverage, passing tests and regression time over multiple regressions
- Displayed above regression table
Test Dashboard

- From regression link on regression dashboard
- Displays default test metrics
  - Mode
  - Status
  - Run times (Elab, Sim, Wall-clock, CPU)
  - LSF info

5 tests associated with this regression: RTL_reg_1

<table>
<thead>
<tr>
<th>Test Name</th>
<th>Test Mode</th>
<th>Status</th>
<th>Seed</th>
<th>Elab Time</th>
<th>Sim Time</th>
<th>Sim Memory</th>
<th>Sim V.Memory</th>
<th>LSF ID</th>
<th>Host</th>
<th>CPU Time</th>
<th>Real Time</th>
<th>Wait Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>basic_io_test</td>
<td>RTL</td>
<td>Pass</td>
<td>52577</td>
<td>35.2</td>
<td>127.5</td>
<td>235M</td>
<td>980M</td>
<td>43824</td>
<td>compute1</td>
<td>135</td>
<td>160.2</td>
<td>0.7</td>
</tr>
<tr>
<td>smoke_test_1</td>
<td>RTL</td>
<td>Pass</td>
<td>47322</td>
<td>47.5</td>
<td>98.4</td>
<td>130M</td>
<td>514M</td>
<td>43825</td>
<td>compute2</td>
<td>100</td>
<td>147</td>
<td>0.5</td>
</tr>
<tr>
<td>smoke_test_2</td>
<td>RTL</td>
<td>Pass</td>
<td>121815</td>
<td>42.5</td>
<td>90.6</td>
<td>154M</td>
<td>622M</td>
<td>43826</td>
<td>compute1</td>
<td>97.8</td>
<td>139.1</td>
<td>165.1</td>
</tr>
<tr>
<td>branch_rw_1</td>
<td>RTL</td>
<td>Pass</td>
<td>921959</td>
<td>87.4</td>
<td>195.4</td>
<td>347M</td>
<td>1138M</td>
<td>43827</td>
<td>compute2</td>
<td>257.6</td>
<td>288.4</td>
<td>150.2</td>
</tr>
<tr>
<td>rand_mem_test</td>
<td>RTL</td>
<td>Pass</td>
<td>20732</td>
<td>62.1</td>
<td>135.7</td>
<td>154M</td>
<td>622M</td>
<td>43828</td>
<td>compute5</td>
<td>178.5</td>
<td>201.3</td>
<td>307.2</td>
</tr>
</tbody>
</table>
Usage of Data

• Project planning
  – Use past project performance to predict future
  – Properly set customer expectations
  – Plan resource utilization (People, licenses, hardware)

• Current project resource utilization
  – Need more licenses? Hardware? Engineers?
  – Efficiently utilizing hardware?

• Status Meetings
  – Accurately track project closure trend
  – Automate status reporting
  – Minimize meeting time
Questions?