

Regressions in the 21st Century – Tools for Global Surveillance

David Crutchfield, Cypress Semiconductor

Tushar Gupta, Cypress Semiconductor

Frank Roberts, Cypress Semiconductor

Venkataramanan Srinivasan, Cypress Semiconductor

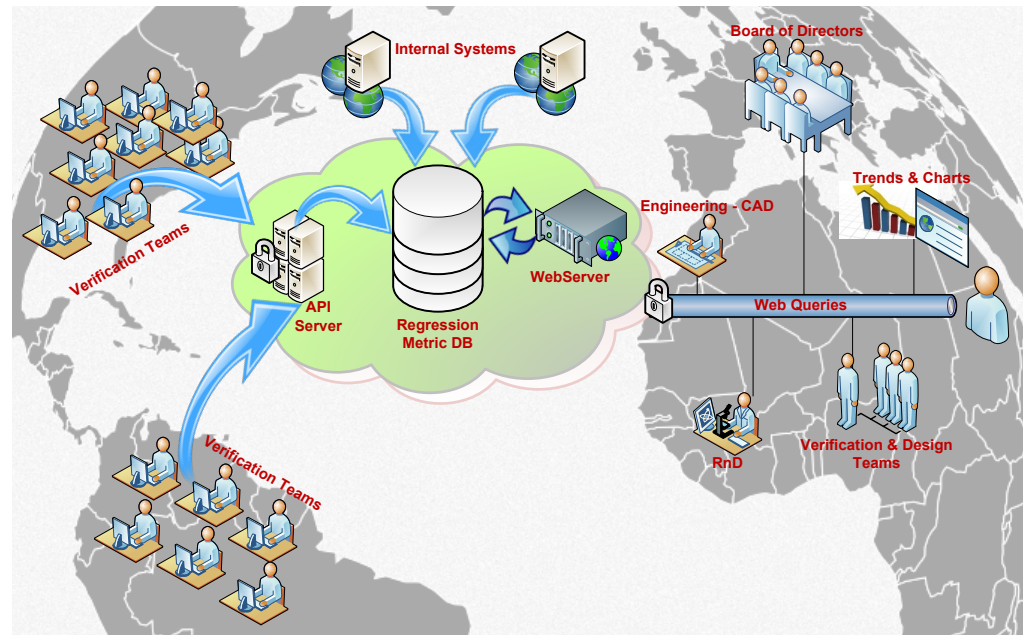


Agenda

- 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

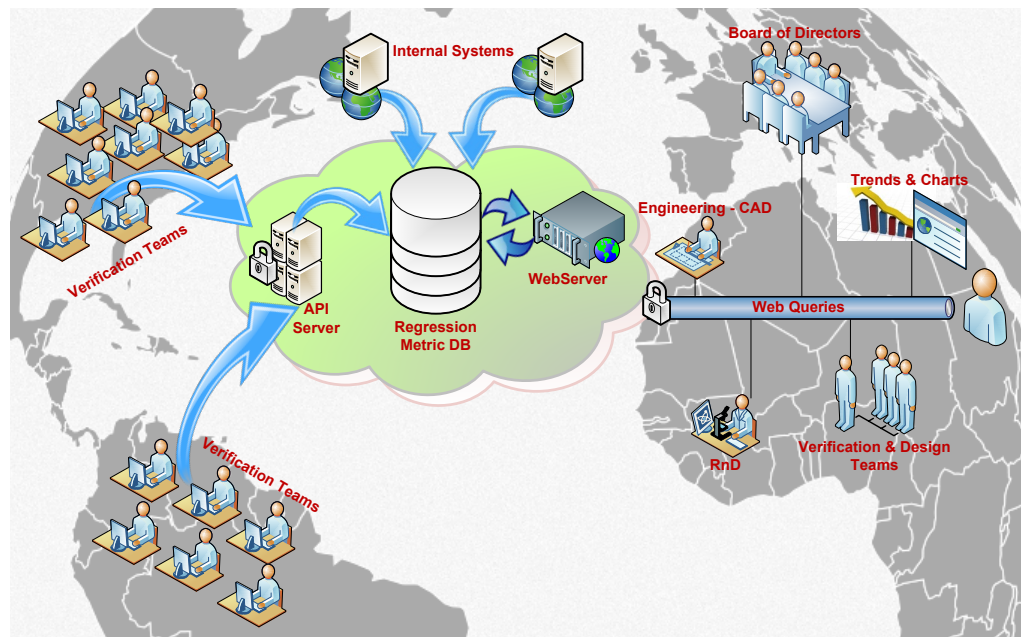
- Design complexity is increasing / time-to-market is shrinking
- Verification is often in critical path / status is heavily scrutinized
- Verification spans global sites
 - Products come from IP developed globally
 - Verification teams spread globally
 - Global management of projects



Global Surveillance

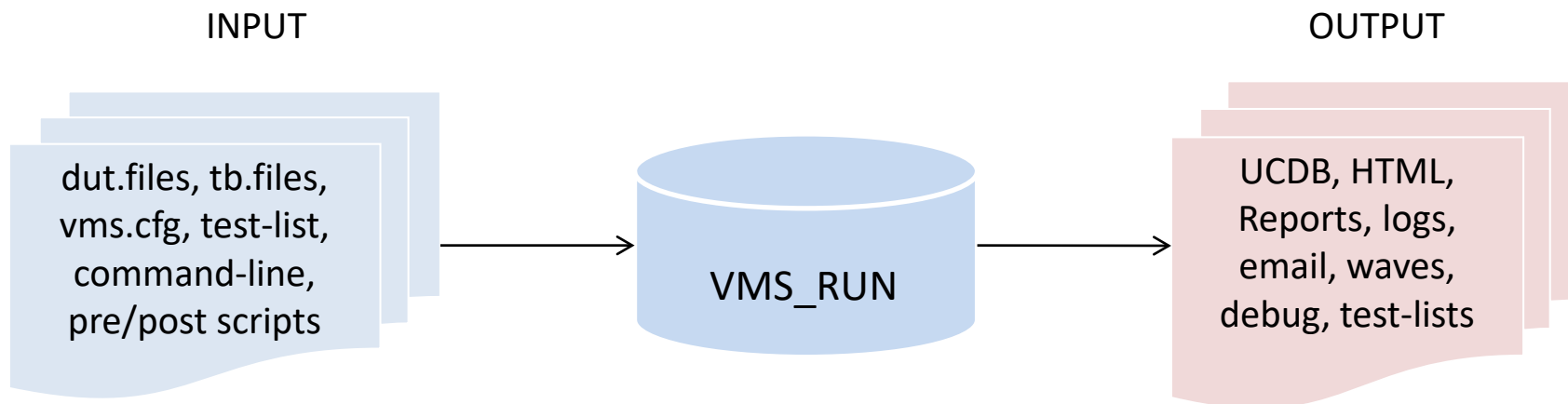
What is needed?

- Automation of centralized metrics gathering
- Standardization of status reporting
- Access to status from anywhere



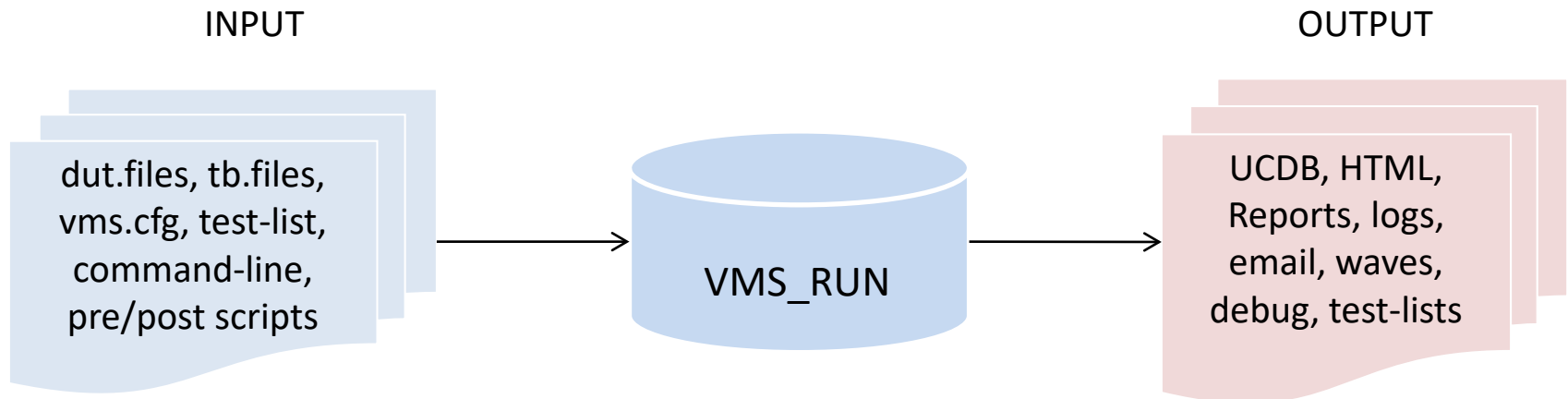
Existing Verification System

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



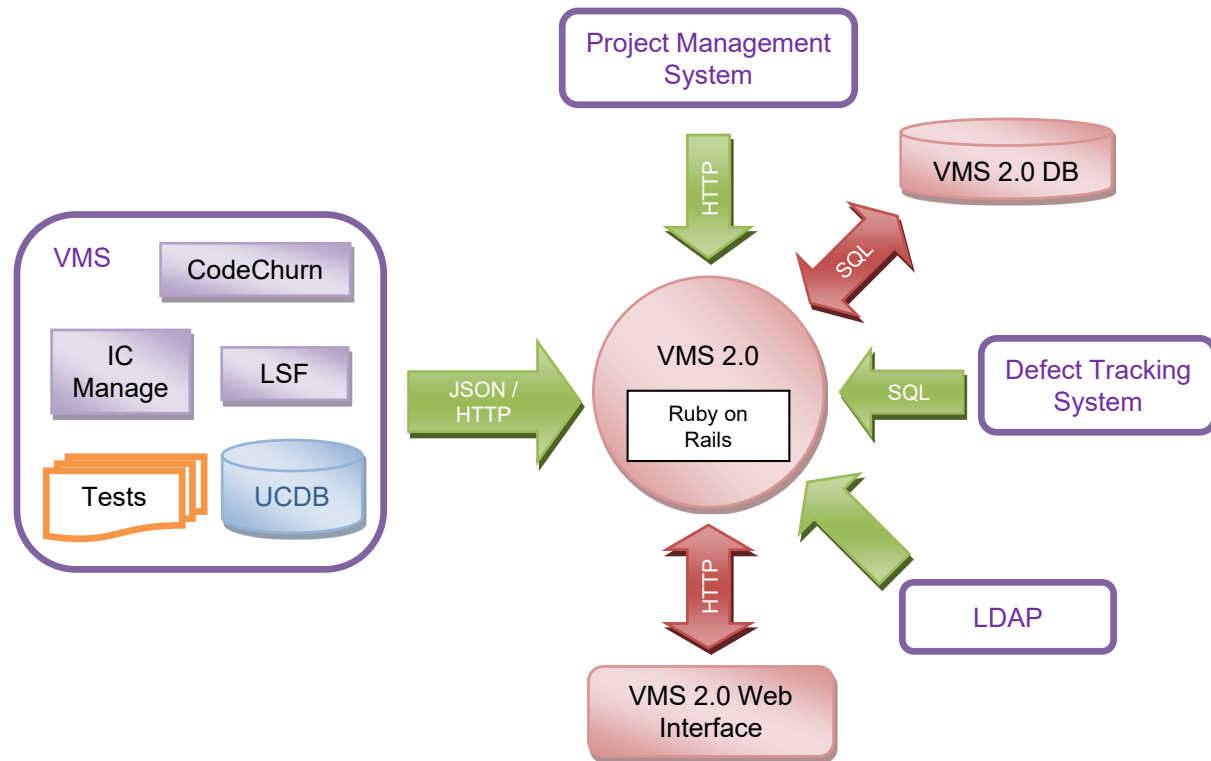
Existing Verification System

- 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



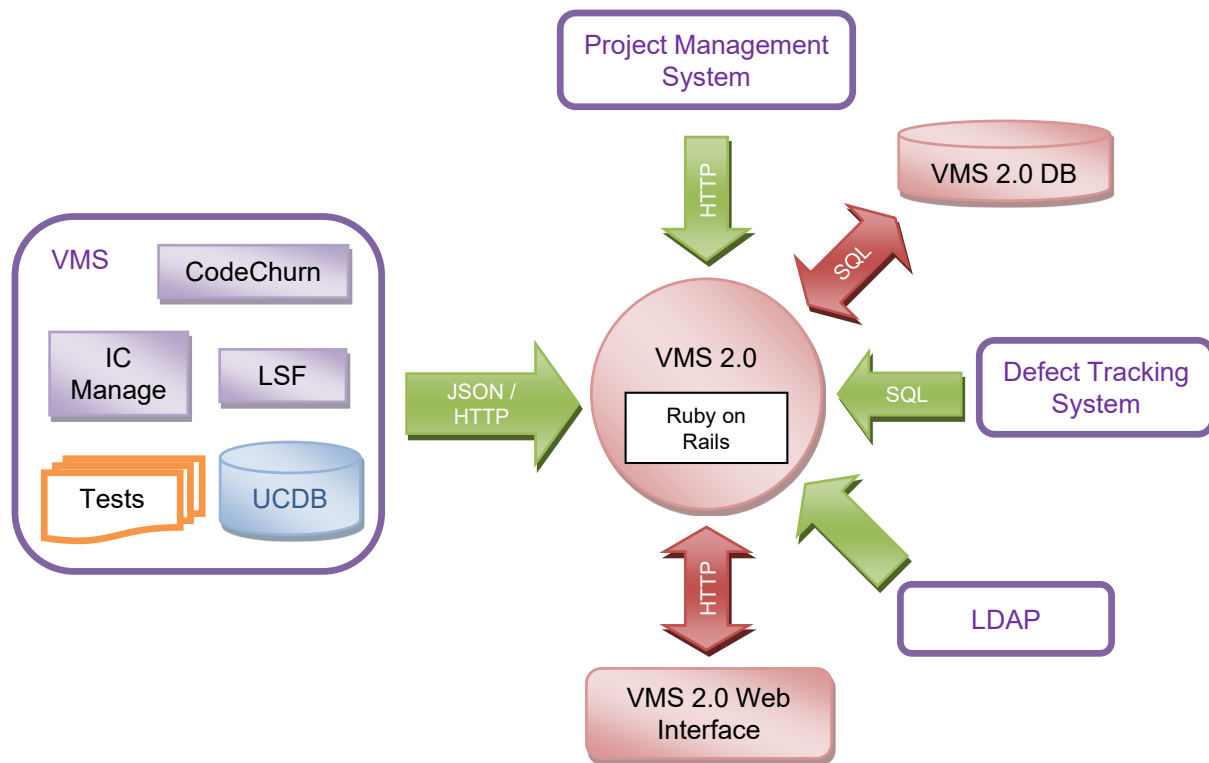
VMS 2.0 Provides

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



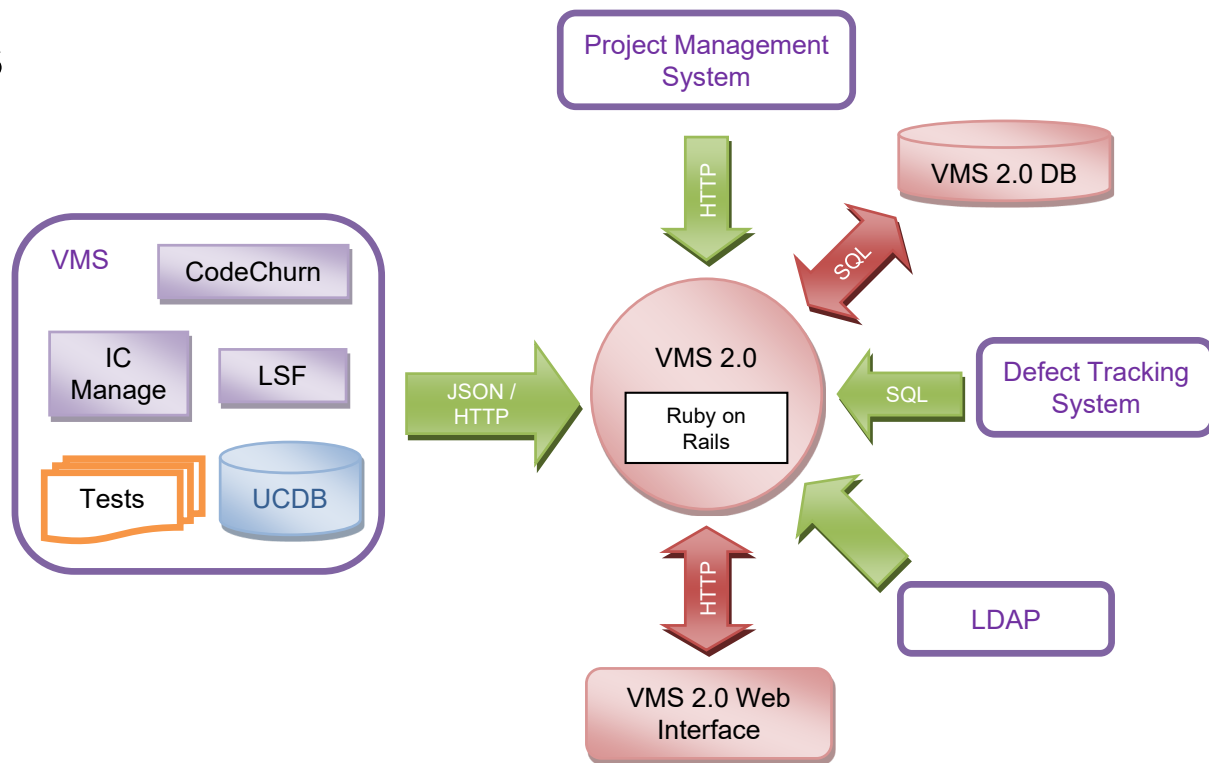
VMS 2.0 Provides

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



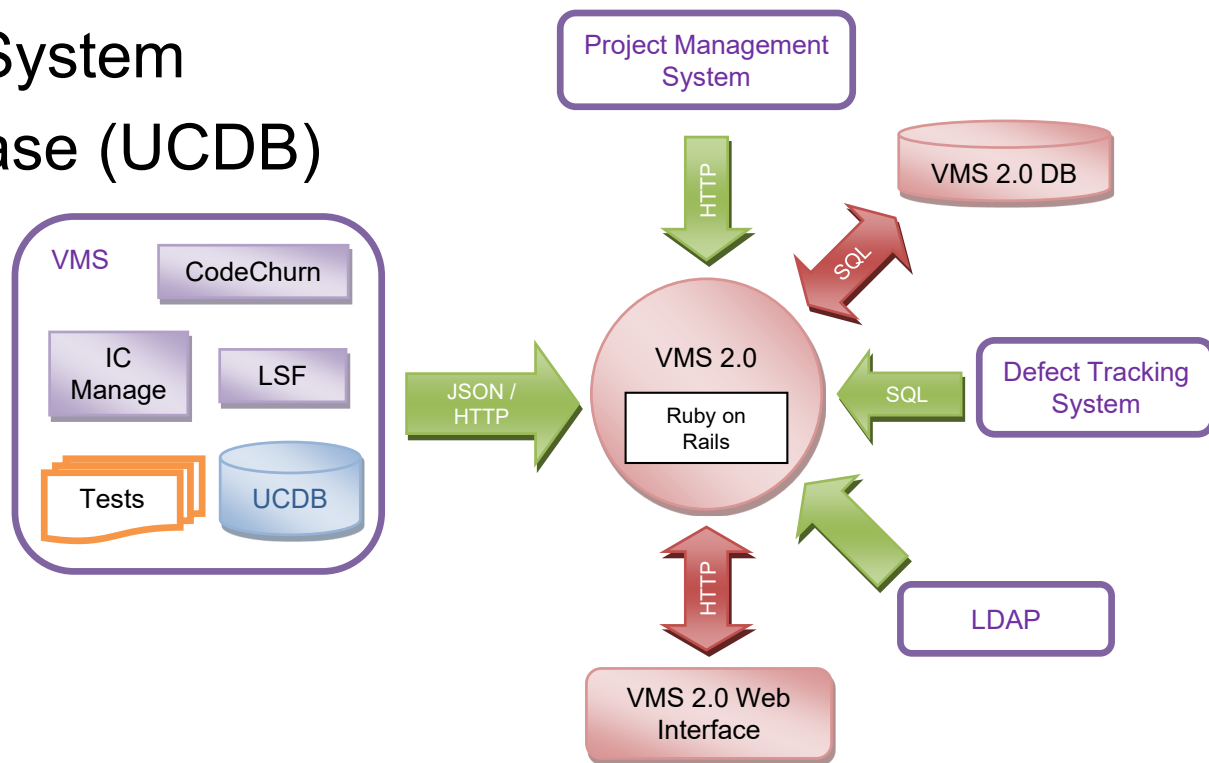
VMS 2.0 Provides

- Customizable landing page / dashboard
- Customizable plots
 - Trends
 - Choice of metrics
 - Multiple metrics



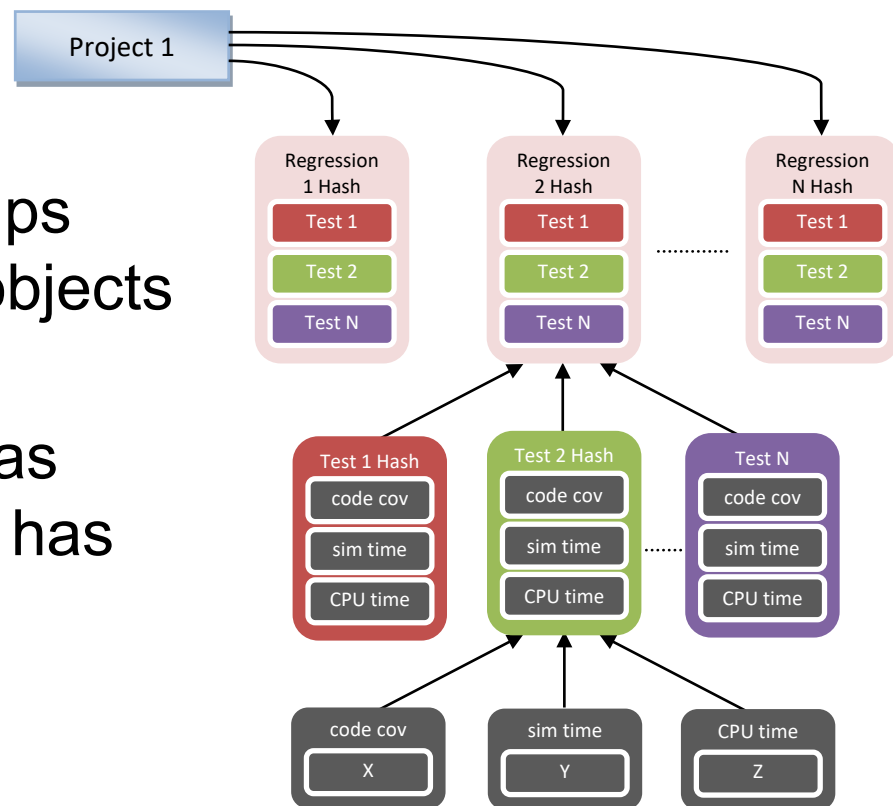
Gathering Information

- 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

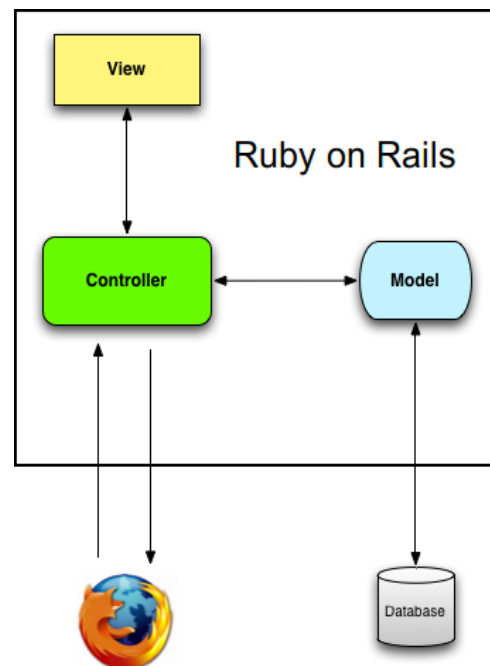


Ruby on Rails

- 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


Development VMS Metrics Repository
PERFORM


David Crutchfield (Logout)

[Home](#)
[All IP Projects](#)
[All Chip Projects](#)
[VMS 2.0 Help](#)


David Crutchfield (**3** subscriptions)

No.	Project	IPS1	IPS2	IPS3	IPS4	CDT	Subscription
1	project_1	1535	1540	1550	1603	Defect Tracking Report	Unsubscribe
2	project_2	1540	1550	1603	1610	Defect Tracking Report	Unsubscribe
3	project_3	1547	1604	1614	1620	Defect Tracking Report	Unsubscribe

[About](#)
[Contact](#)
[File SPR](#)

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)

← Previous 1 2 3 Next →

No.	Project	IPS1	IPS2	IPS3	IPS4	CDT	Subscription
1	project_1	1535	1540	1550	1603	Defect Tracking Report	Subscribe
2	project_2	1540	1550	1602	1610	Defect Tracking Report	Subscribe
3	project_3	1547	1604	1614	1620	Defect Tracking Report	Subscribe
4	project_4	NONE	NONE	1550	1601	Defect Tracking Report	Subscribe
5	project_5	1550	1610	1622	1641	Defect Tracking Report	Subscribe

← Previous 1 2 3 Next →

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)

← Previous 1 2 3 4 5 6 Next →

No.	Project	Launch	PR1	PR2	PR3	PR4	PR5	CDT	Subscription
1	Chip 1	1451	1512	1512	1519	1553	1553	Defect Tracking Report	Subscribe
2	Chip 2	1516	1525	1525	1526	1610	1613	Defect Tracking Report	Subscribe
3	Chip 3	1441	1441	1441	1508	1538	1549	Defect Tracking Report	Subscribe
4	Chip 4	1422	NONE	NONE	1438	1549	1722	Defect Tracking Report	Subscribe
5	Chip 5	1433	1445	1502	1514	1550	1609	Defect Tracking Report	Subscribe
6	Chip 6	1439	NONE	NONE	1440	1518	1618	Defect Tracking Report	Subscribe

← Previous 1 2 3 4 5 6 Next →

Regression Dashboard

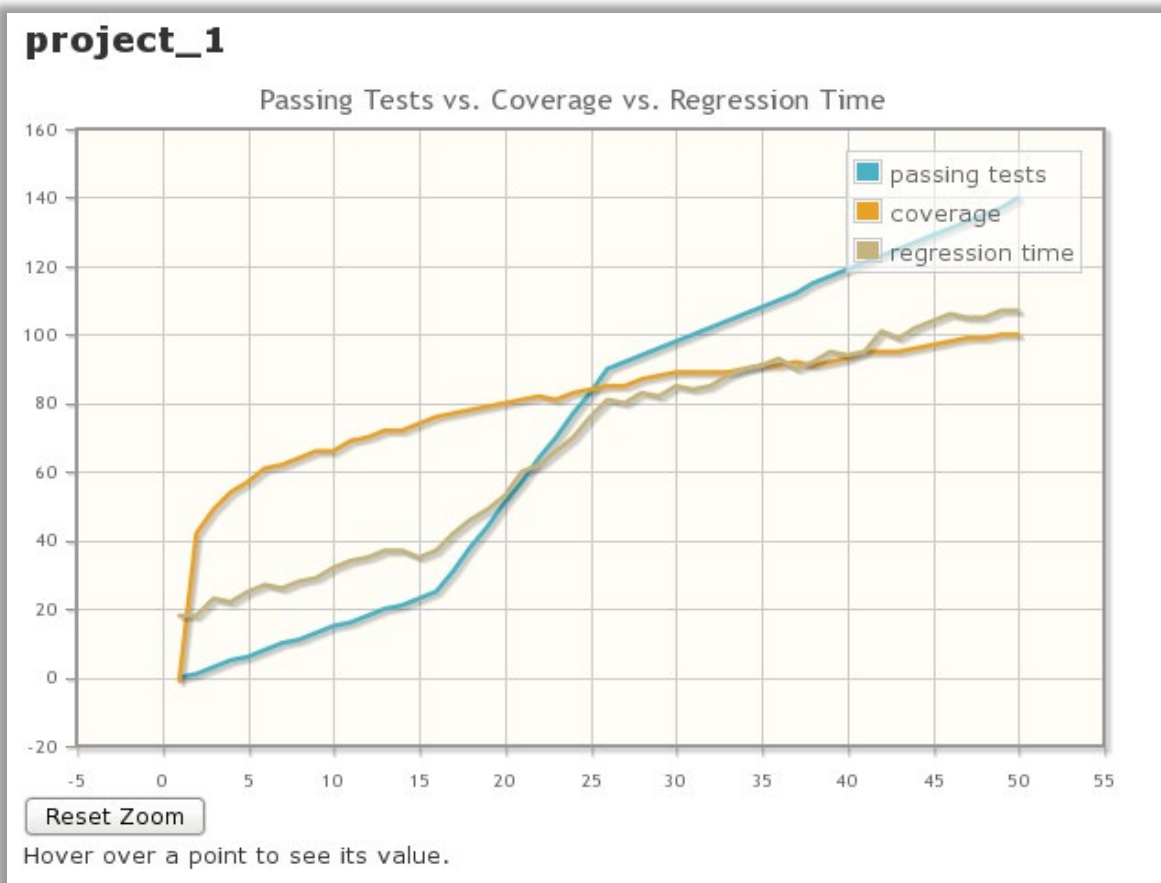
- Provides table of default metrics
 - Total coverage
 - Number of tests
 - Design / Test bench churn
 - Run times
 - Host / License utilization

50 regressions associated with this project: project_1

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

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

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

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?

