Darron May
Mentor Graphics Inc
Darron_May@mentor.com

Are you really confident that you are getting the very best from your verification resources?

Fritz Ferstl
Univa
fferstl@univa.com

Abstract

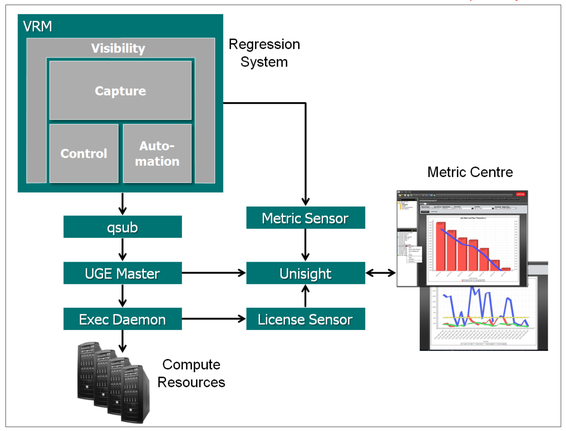
Getting the very best from your verification resources requires a regression system that understands the verification process and is a tightly integrated with Workload Management and Distributed Resource Management software. Both requirements depend on visibility into available software and hardware resources, and by combining their strengths, users can massively improve productivity by reducing unnecessary verification cycles.

Introduction

Smart combination of the grid and regression management systems to ensure that every verification cycle is a valid cycle requires:

- Best in class regression and grid systems
- Communication between the systems
- Visibility into metrics to allow measurement of both verification and resource effectiveness

The complete System



- Typical Regression Limitations
  - Expensive to create & maintain
  - No alternative but to home grow
  - Script based, with little separation between configuration and control
  - Hidden data & automation opportunities
- Verification Run Management Benefits
  - Focus resources on verification, not infrastructure development
  - Simplify set-up & Maintenance
  - Continual development & support

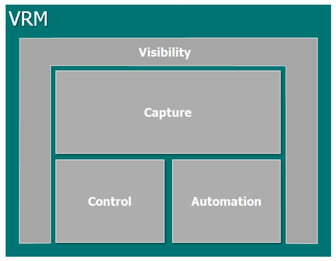
Not getting the most from technical infrastructure used in a product development cycle can be expensive. The cost, which include software licenses, fixed assets, and wasted time, often can exceed that of the original investment in the infrastructure.

Regression System

Twin demands increased quality and shortened design cycles put pressure on IP and SoC houses to leverage automation throughout their design and verification processes. Using a purpose-built regression system can give verification engineers maximum productivity while reducing the maintenance burden. User productivity can be boosted in most aspects of verification management including capacity, performance, resource usage, turnaround time, preparation, maintenance, results and coverage analysis.

- Structured Design
  - Separates capture from control
  - Layered providing expandability
- Improved Throughput
  - Resource balancing
  - Actions Optimization
- Improved Turn-around
  - Actionable data
  - Communication
  - Maintenance

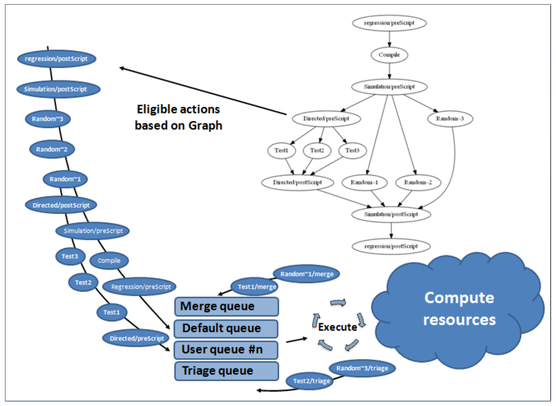
Major functions of VRM



Capture supports inheritance

```
<runnable name="regression" type="group" sequential="yes">
  <members>
    <member>Verification</member>
    <member>TrendAnalysis</member>
    <member>OptimizeSmokeTests</member>
    <member>Reports</member>
  </members>
</runnable>
<runnable name="Verification" type="group" sequential="no" base="testsetup">
  <members>
    <member>Directedtests</member>
    <member>Randtest</member>
    <member>Seedtests</member>
    <member>Formal</member>
  </members>
</runnable>
<runnable name="Directedtests" type="group">
  <parameters>
    <parameter name="TESTLIST">InitialTest ModeTwoTest DataTest FifoTest TxDataTest SyncTest</parameter>
  </parameters>
  <members>
    <member>datatests</member>
    <member>registertests</member>
  </members>
</runnable>
<runnable name="datatests" type="task" if="{(%DATATESTS%)} eq {yes}" foreach="(%TESTLIST%)" index="testname"/>
<runnable name="registertests" type="group">
  <members>
    <member>CPURegisterTest</member>
    <member>VariableTest</member>
  </members>
</runnable>
```

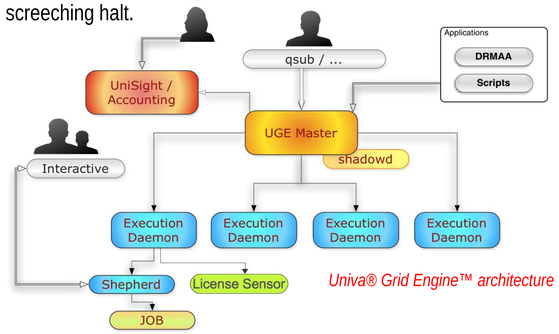
Control and Automation of the process



- Run until …
  - Coverage, no improvement, time, number cycles
- Automation of …
  - Pass/Fail, merging, ranking, re-run, clean up, seed management

Workload and Distributed Resource Management

Software for Workload Management (WLM) and Distributed Resource Management (DRM) has become a fundamental building block of compute farms or grids and large-scale technical computing data centers. It is as crucial as the networking infrastructure or file sharing services and provides a similar type of service (and potential bottleneck) to data centers that a conveyor provides to the assembly line in an industrial manufacturing factory: if it slows down, the production gets severely impeded, and if it stops, then everything comes to a screeching halt.



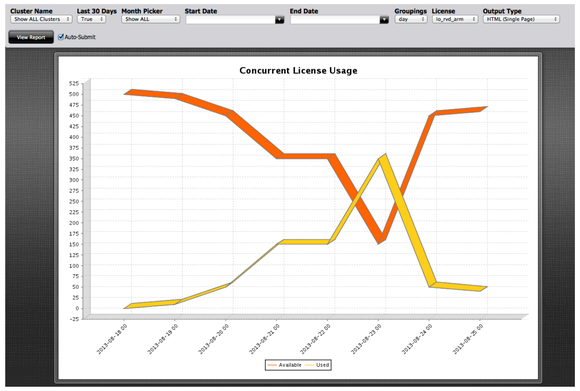
Univa® Grid Engine™ architecture

- Dependable
  - Avoiding costly downtimes
- Responsive and Scalable
  - Meet throughput requirements and keep utilization near the optimum
- Flexible
  - Adapt to the changing infrastructure complexities and allow for implementation of policies reflecting the operational goals of an organization

Managing Software Licensing

Organizations struggle to get a grip on license utilization. Many organizations have subsidiaries across the world, each with their own pool of licensed software. Asking questions like “Where is excess capacity?” or “Where is insatiable demand?” is a key benefit using license orchestration software in conjunction with the WLM/DRM system.

Impact of license usage orchestration



Integration Between Regressions And Execution

The regression system separates execution from capture with a particular method describing transparent ways to execute the actions or jobs.

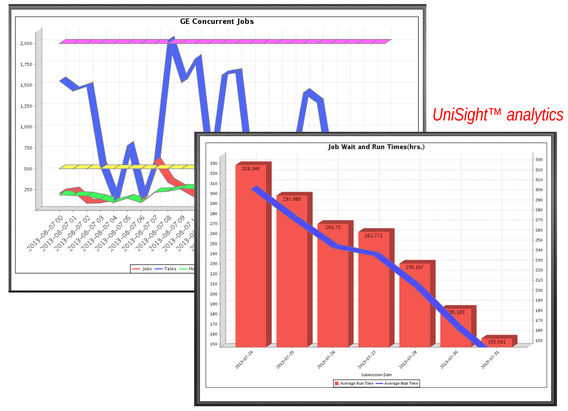
The ‘qsub’ command is the executable used to submit actions to the grid and by setting the ‘maxarray’ attribute to more than one will cause the regression system to pack the actions into arrays for submission to the grid.

Grid specific method

```
darronm@rocket:/mnt/hgfs/scratch
File  Edit  View  Terminal  Tabs  Help
<runnable name="simulate" type="group" sequential="no">
  <method name="grid" gridtype="uge" maxarray="(%ARRAY%)"
          mintimeout="3000" if="{(%MODE:%)} eq {grid}">
    <command>qsub (%GRIDOPTS%) (%WRAPPER%)</command>
  </method>
</runnable>
                                                100,1         99%
```

Monitoring Metrics

Having pertinent reporting and monitoring tooling entails three aspects: gathering comprehensive metrics, providing analytics to distil useful reports from that data, and having a user interface allowing for easy navigation.



UniSight™ analytics

Results

This paper has detailed how this can make massive improvements to the productivity and throughput of regressions with some real examples shown in the table below.

Industry	Sub process	Productivity	Pre-VRM/Analysis	With VRM/Analysis	Benefits
IP Developer	Nightly regression test time	Throughput	28 hours	2.5 hours	9 X faster
	Results and Coverage Analysis	Turn-around	2 hours	20 minutes	6 X faster
	Regression file cleanup	Capacity	15 minutes	30 seconds	30 X faster
Automotive	Nightly regression test maximum	Throughput	40 tests	320 tests	8 X more tests
	Nightly regression test setup time	Turn-around	30 minutes	2 minutes	15 X less time
	Nightly regression addition time	Turn-around	60 minutes	5 minutes	12 X less time
	Nightly regression Script Files	Turn-around	10 files	1 file	10 X easier
	Nightly regression Results Analysis	Turn-around	> 1 hour	<1 minute	60 X faster
Memory	Project #1 regression time	Throughput	120 hours	23 hours	> 5 X faster
	Project #2 regression time	Throughput	17.5 days	2 days	> 8 X faster
	Regression file cleanup	Capacity	Space exceeded	Complete	Regressions Complete
Semiconductor	System regression test redundancy	Throughput	38.9 hours	9.4 hours	> 4X faster
Gaming	Reduce runtime variation	Throughput	33 hours	11 hours	3 X faster
Micro-processor	Job clubbing	Throughput	350 minutes	125 minutes	~ 3X faster