Parallelizing a Distributed Testing Environment

Teague Algie

Cleversafe
Role of Integration/Regression Tests

Merging features

Releases
History

Manual Testing

Basic Assistance Framework

Full Product Framework

+ Automated Test Runner
How Many Tests Are There?

Tests Run in the Regression Suite

Number of Automated Tests Run

Date Tests Were Run

04/04/2011 09/04/2011 02/04/2012 07/04/2012 12/04/2012 05/04/2013 10/04/2013 03/04/2014 08/04/2014 01/04/2015 06/04/2015
How Much Time is Spent Running Tests?

Hours Spent Running Tests, by Date

Date Tests Were Run

Hours to Complete Test Suite
Where is Time Spent per Test?

Test Phase Time Breakdown

- Initialization: 2%
- Test Setup: 16%
- Test Running: 70%
- Deinitialization: 12%
Decreasing Per-Test Runtime

Not an effective solution in the end

- Expensive
- Impermanent
- Fragile
Serial Testing Does Not Scale

- Long turnaround before releases
- Decreased iterations on tests
- Unfortunate surprises
Total Runtime with Added Hardware

Decreasing Job Runtime by Adding Hardware

Suite Runtime (Hours)

Available Hardware

1x 2x 3x 4x 5x 6x
Best Case: Unlimited Hardware

Decreasing Job Runtime by Adding Hardware

Suite Runtime (Minutes)

Best Case Available Hardware (N)

(N-4)x (N-3)x (N-2)x (N-1)x Nx
Requirements for Distributing Test Load

- Maintaining state when pieces go down
- Handling failures of individual components
- Maximizing hardware utilization
- Using resources intelligently
Parallel Jetengine is Born!
Handling Client Death
Handling Orchestrator Failure
Scheduling

- All failure states are allowed within the boundaries of the design.
What Else Do We Get?

- Schedule on specific hardware
- Non homogeneous devices accommodated
Lessons Learned

• Scale is always an issue
• The answer is always to split up the problem
• Be resilient to failures
• Use **expensive** hardware as little as possible
In Practice

- Real world results for 1 test vs serial TBA
- 2 tests vs serial TBA
- N tests TBA
It works!

- In conclusion, do this.