



STORAGE DEVELOPER CONFERENCE

SNIA ■ SANTA CLARA, 2015

Parallelizing a Distributed Testing Environment

Teague Algie

Cleversafe

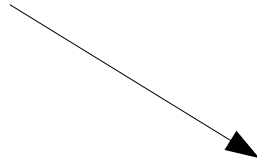
Role of Integration/Regression Tests

Merging features

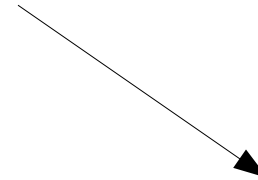
Releases

History

Manual Testing



Basic Assistance Framework



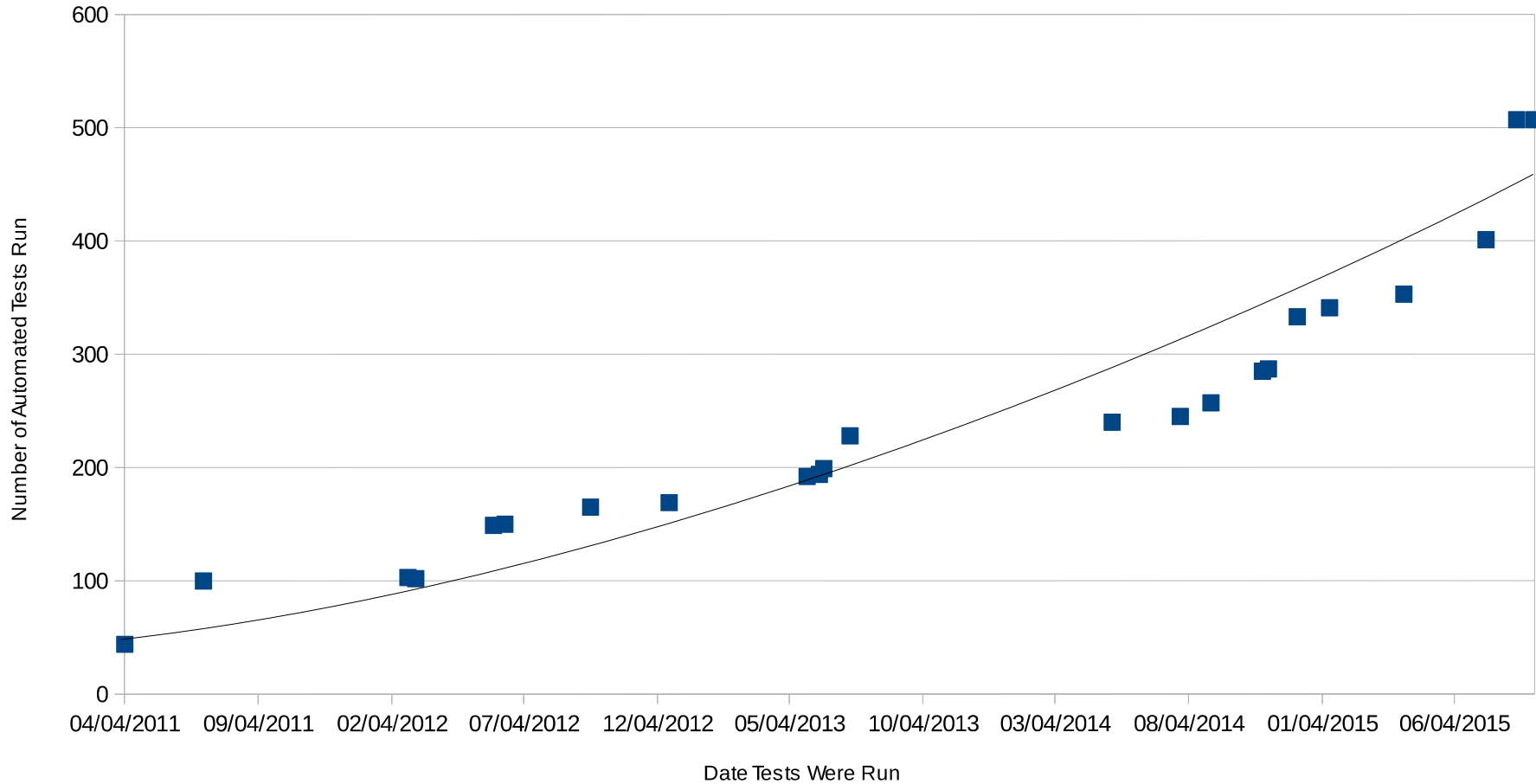
Full Product Framework

+

Automated Test Runner

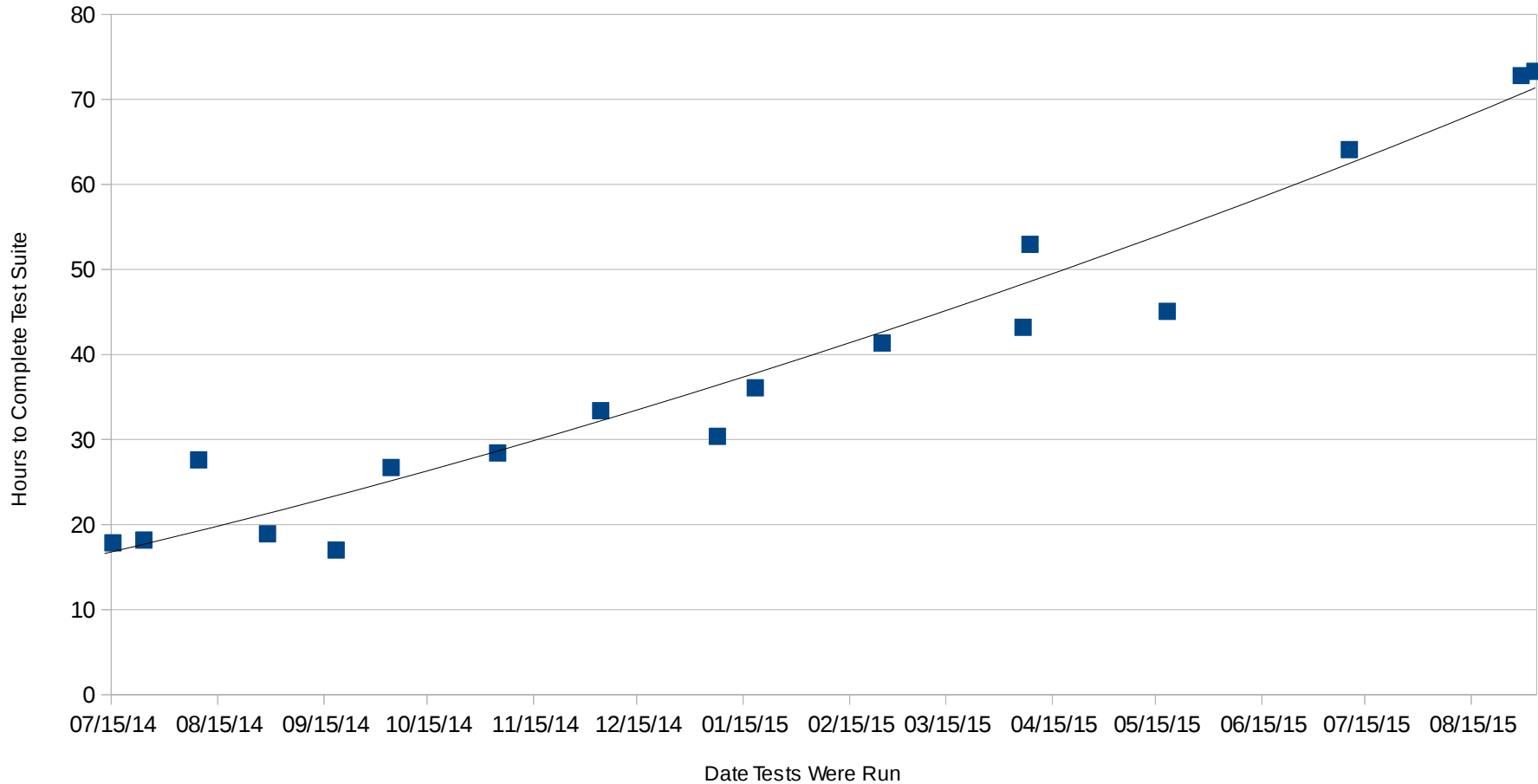
How Many Tests Are There?

Tests Run in the Regression Suite



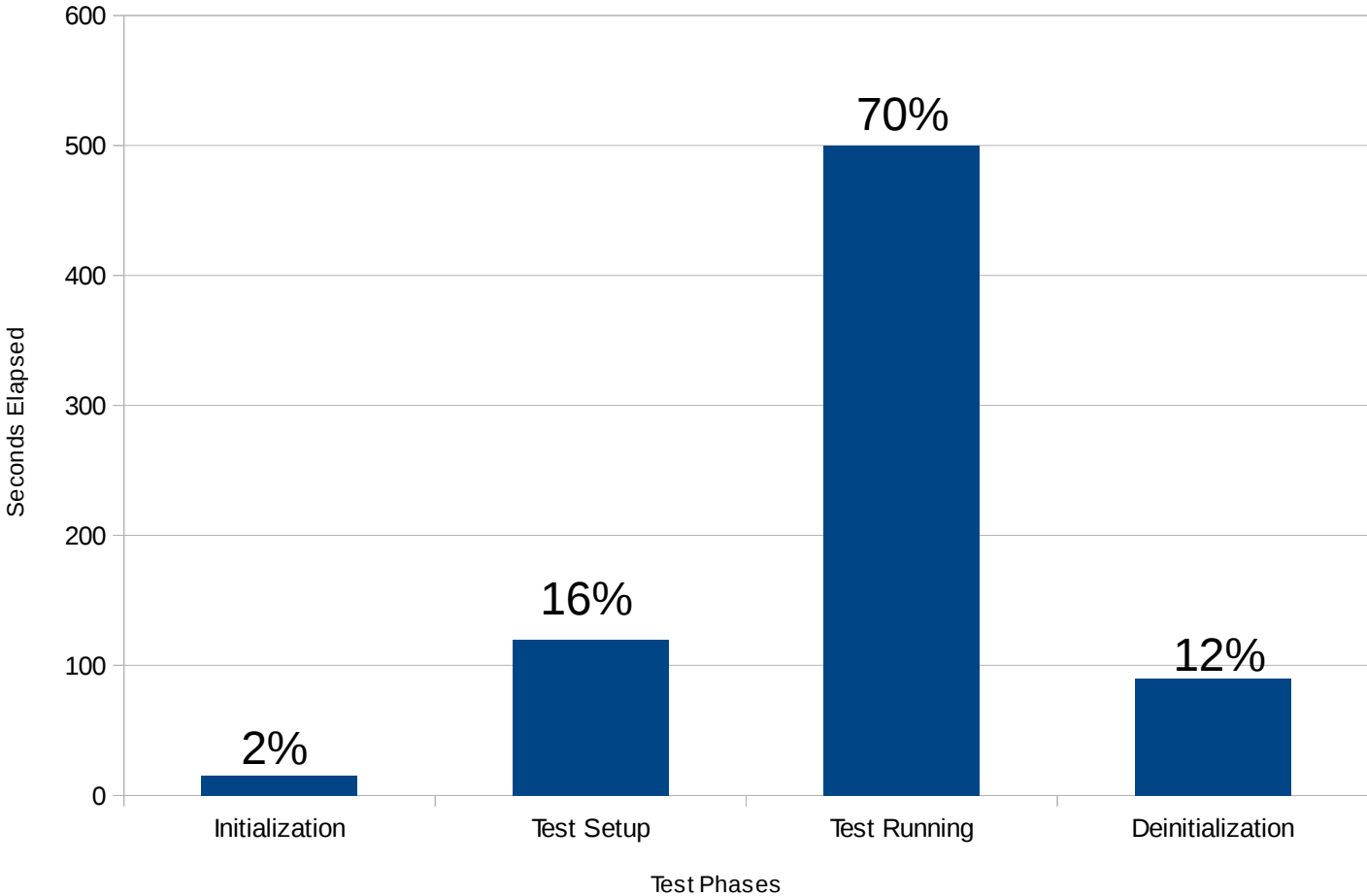
How Much Time is Spent Running Tests?

Hours Spent Running Tests, by Date



Where is Time Spent per Test?

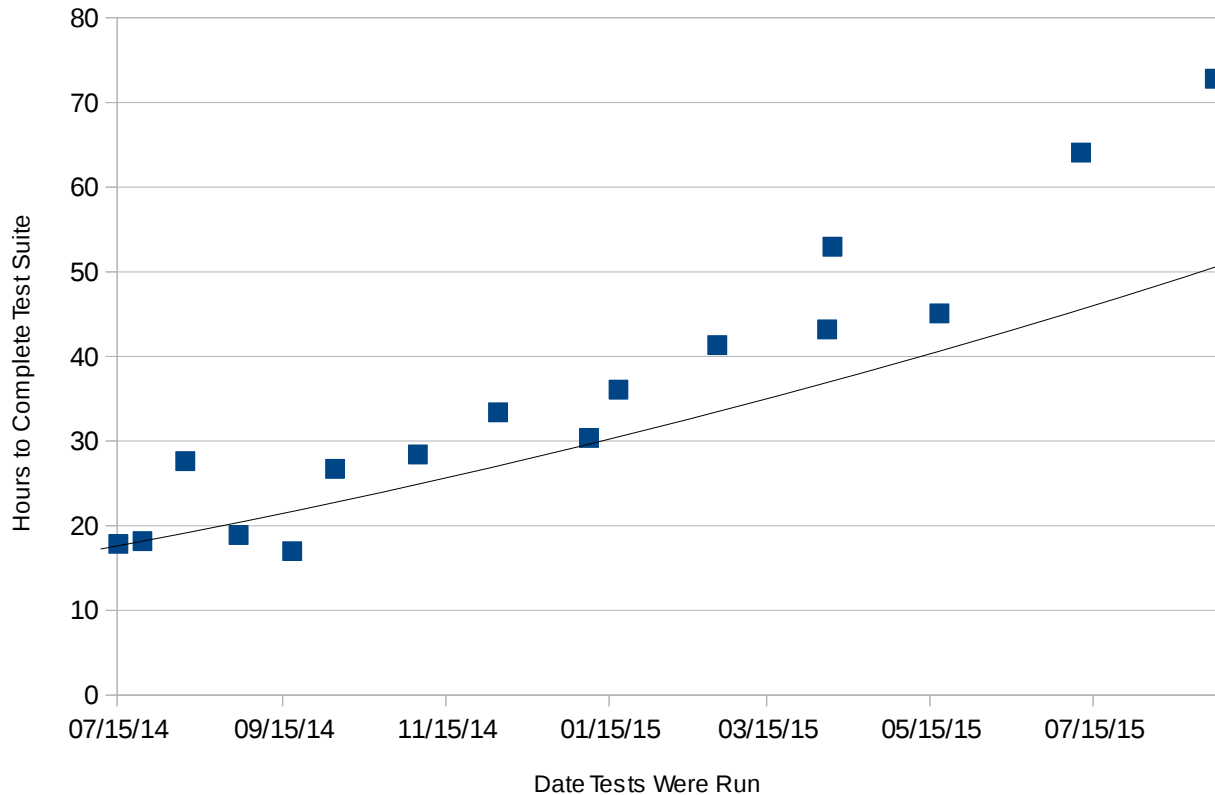
Test Phase Time Breakdown



Decreasing Per-Test Runtime

Not an effective solution in the end

Hours Spent Running Tests, by Date



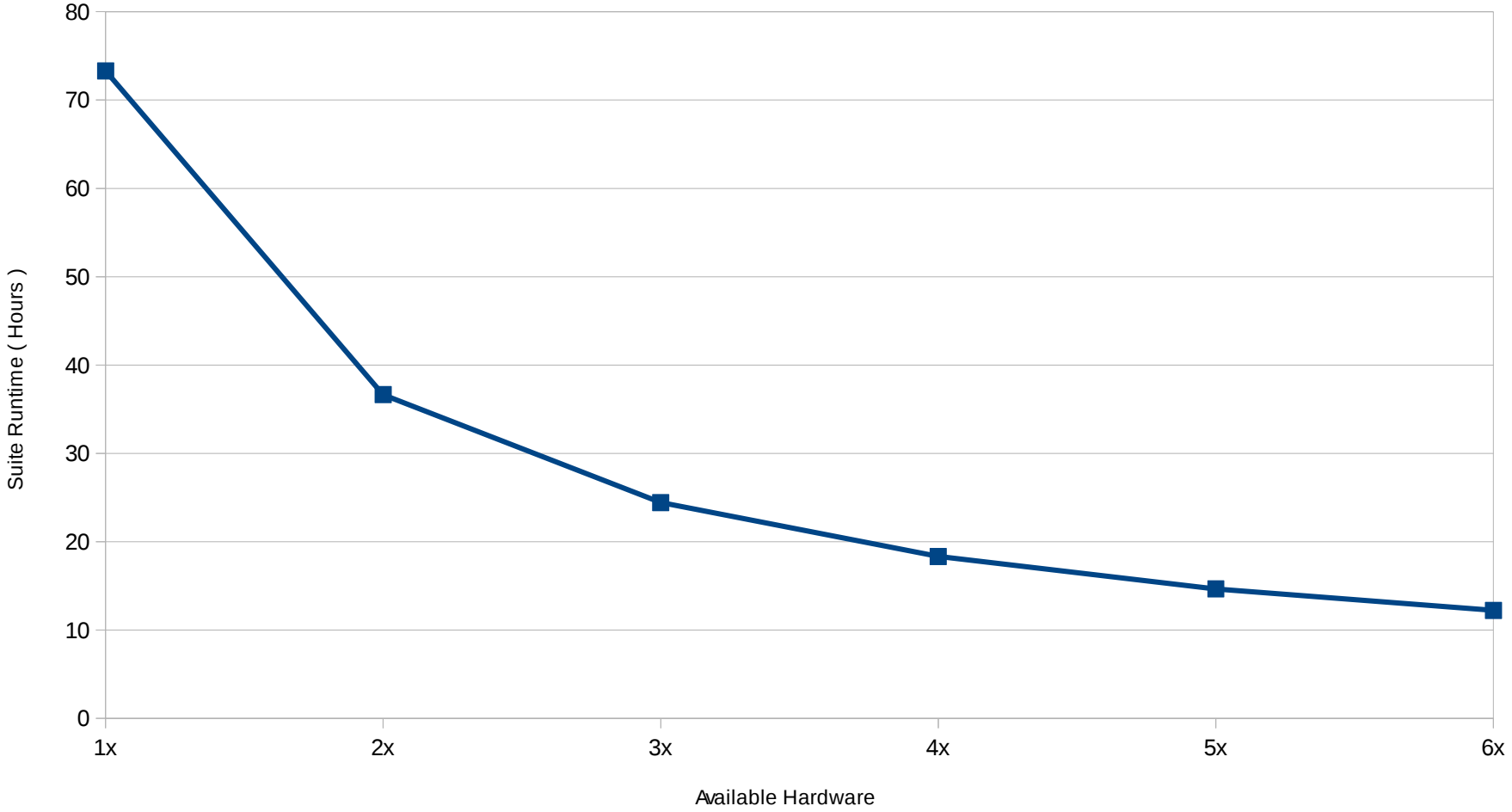
- Expensive
- Impermanent
- Fragile

Serial Testing Does Not Scale

- Long turnaround before releases
- Decreased iterations on tests
- Unfortunate surprises

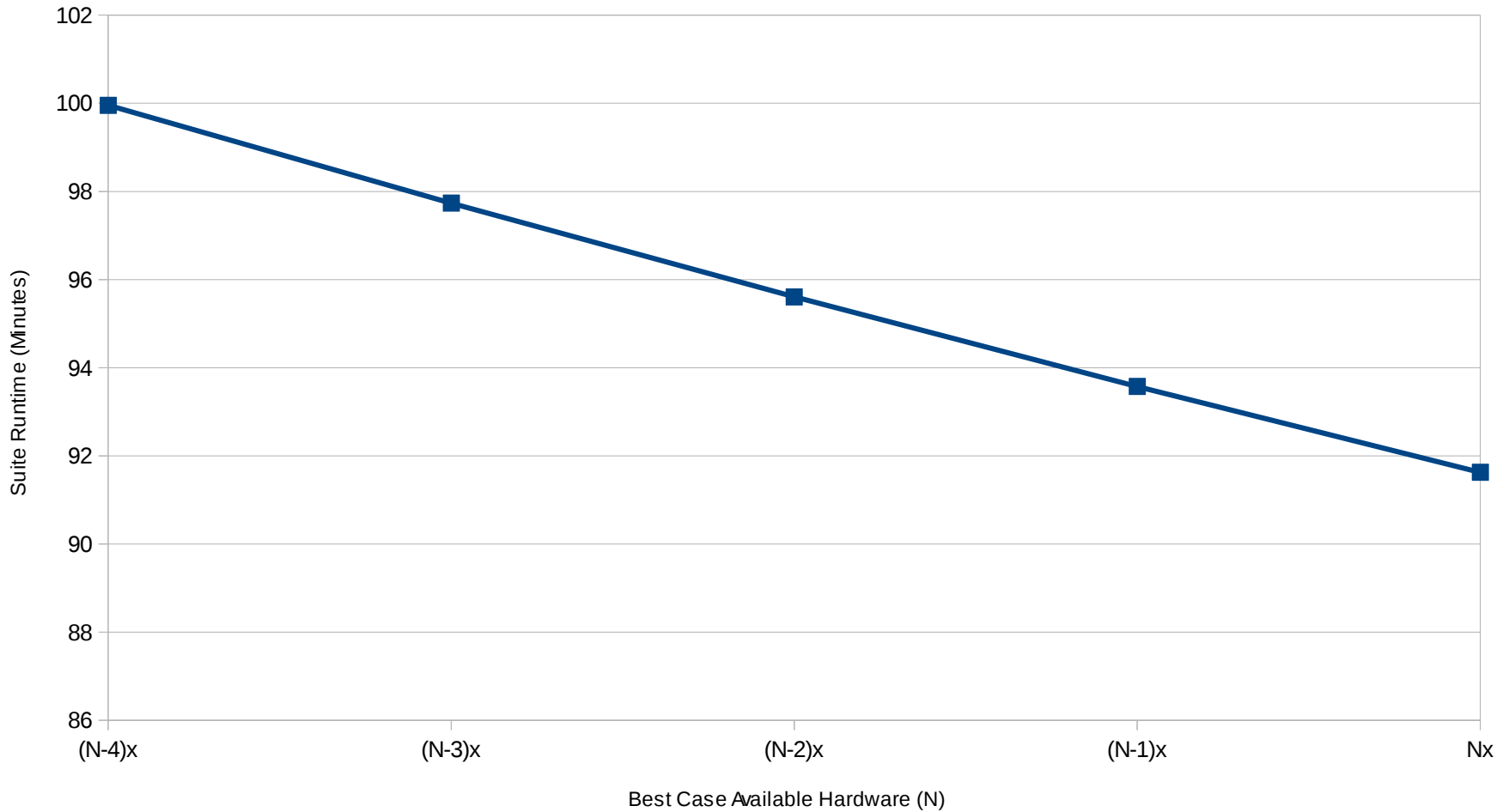
Total Runtime with Added Hardware

Decreasing Job Runtime by Adding Hardware



Best Case: Unlimited Hardware

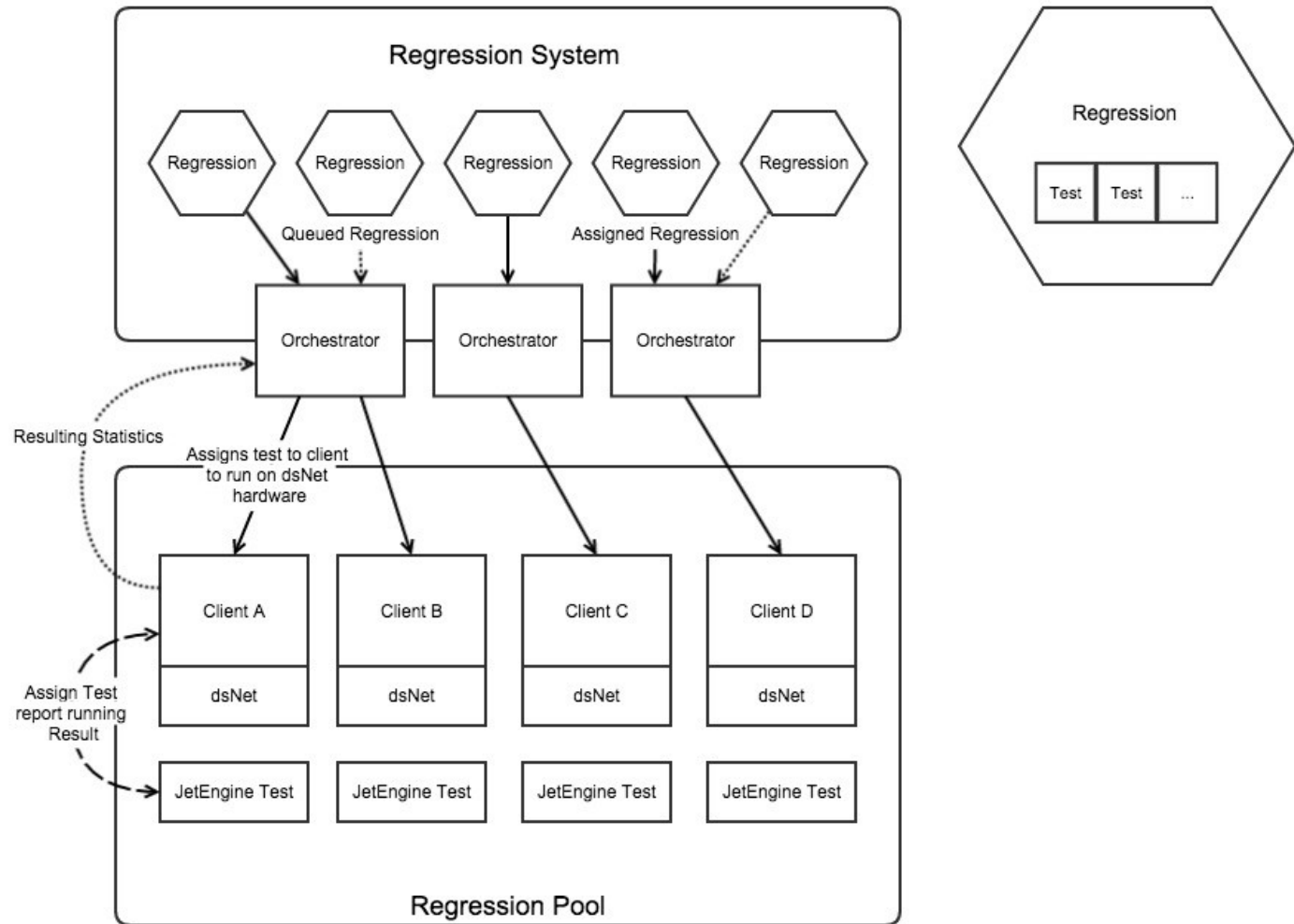
Decreasing Job Runtime by Adding Hardware



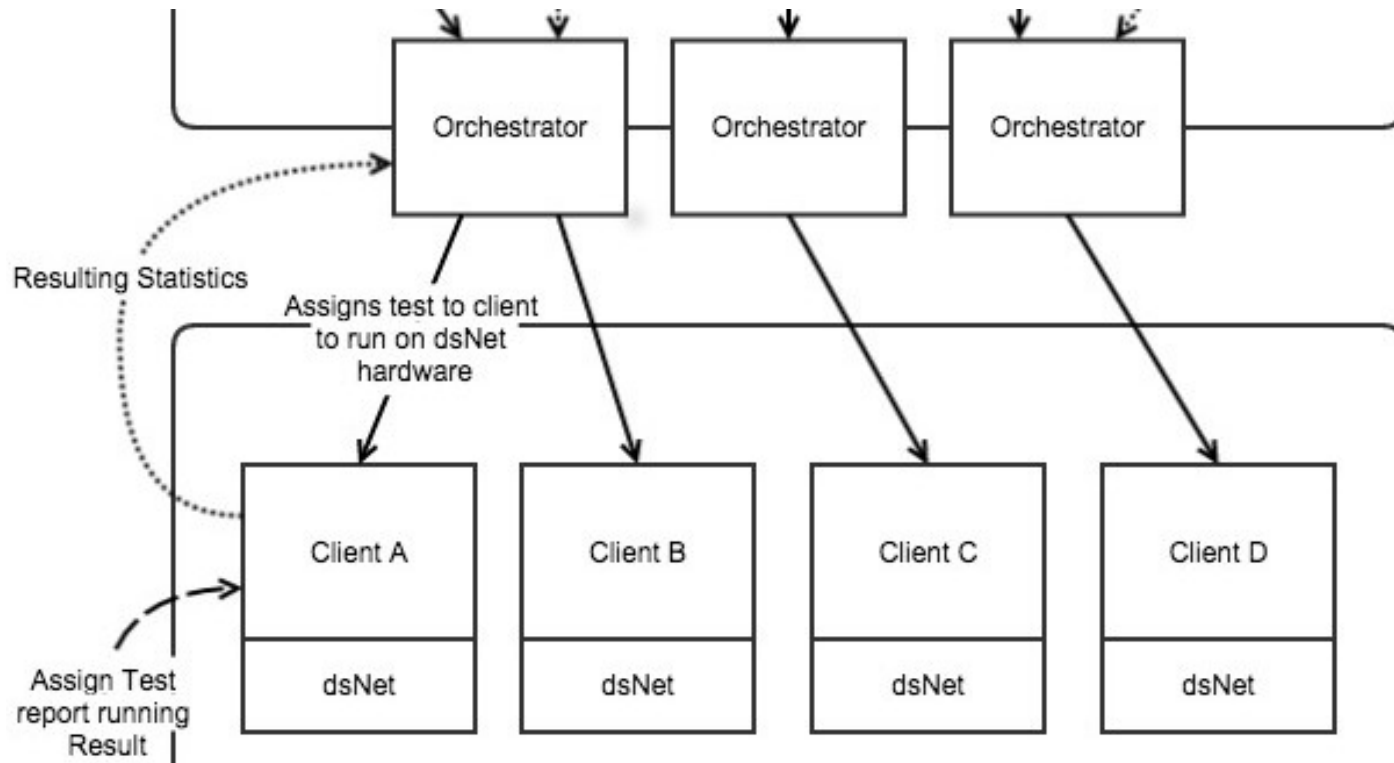
Requirements for Distributing Test Load

- Maintaining state when pieces go down
- Handling failures of individual components
- Maximizing hardware utilization
- Using resources intelligently

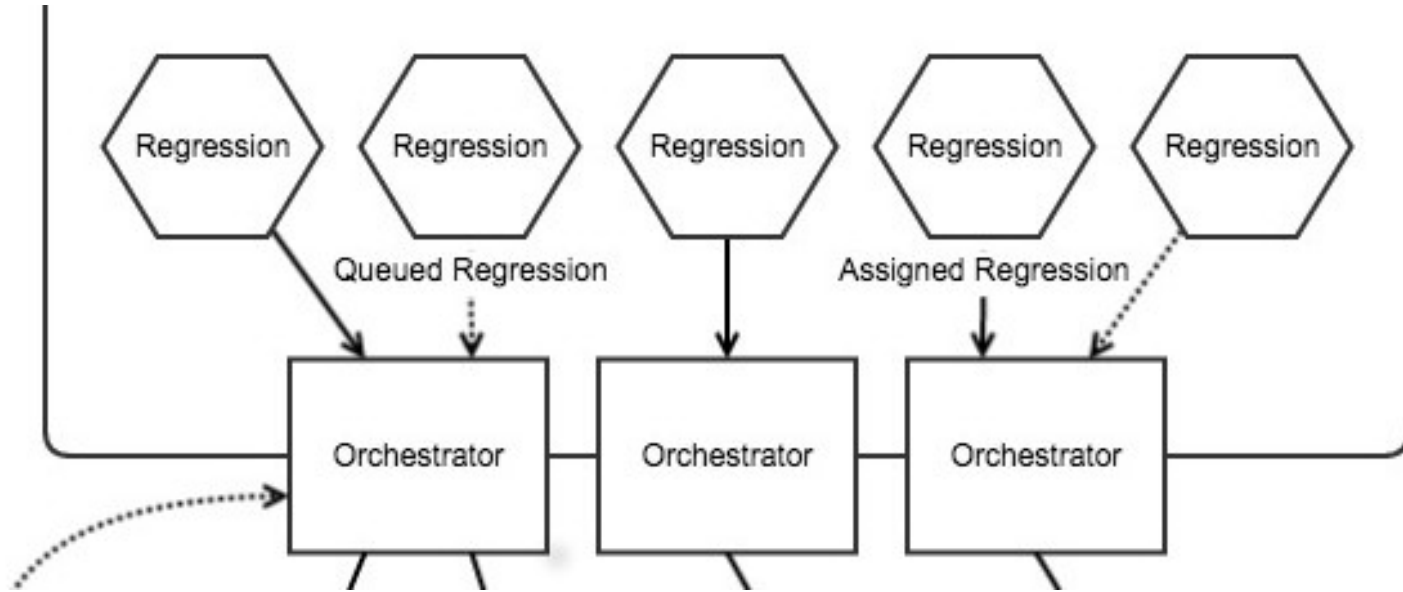
Parallel Jetengine is Born!



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 accomodated

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.