



## Customer Case Study

# Pillar Data Systems tests application-aware storage products using the SwiftTest 3000.



Pillar Data Systems, provider of application-aware storage systems, enhances performance testing capabilities by implementing the SwiftTest 3000. Sorely in need of additional load testing capacity, the Pillar test team finds that the SwiftTest 3000 provides all the traffic and test configuration flexibility they need to put their efficient, high-capacity Pillar Axiom systems to the ultimate test.

### About Pillar Data Systems

Pillar Data Systems designs and builds cost-effective networked storage solutions with the highest utilization rates in the market.

The company's flagship product, the Pillar Axiom, is a true application-aware system, enabling multiple tiers of storage on a single platform—each with a unique level of service. The result is an ultra high-capacity storage system which consolidates assets, modularly scales capacity, and reduces management complexity.

### Pillar test environment

The Pillar test environment includes racks of midrange white boxes, switches, and tape libraries—a full complement of servers, storage, and support hardware needed to adequately stress the Pillar product line before going to market.

The team needs to test both the performance and the unique application-aware functionality of the Axiom product line, using high volume CIFS and NFS traffic.

### Challenges at Pillar

- **Limited load testing capacity.**  
*Performance is insufficient to adequately test Axiom systems.*
- **Desire greater granularity and test sophistication.**  
*Testing the unique features of the application-aware Axiom requires extremely detailed and flexible tests.*
- **Difficult, time-consuming virtual server management.**  
*Rampant VM client sprawl.*

### Changes with SwiftTest

- **Unparalleled load testing.**  
*Can validate the extreme performance of the Axiom.*
- **Comprehensive, flexible tests.**  
*Able to fully test the features that make the Pillar Axiom unique.*
- **No virtual machine sprawl.**  
*SwiftTest 3000 replaced all of the virtual machine clients, and more.*

Colin Hutcheson is the Pillar test architect responsible for designing test processes and systems to best put the Pillar Axiom through its paces.

### The virtual machine farm

Additionally, Pillar runs a virtual server farm to generate load and run test scenarios. Colin has set up the environment to be able to create virtual machines that are each configured for a different scenario. He would deploy a fleet of virtual machines to run each test case—easy to deploy but hard to clean up. Every test case requires a client and disparate application layer test suite.

### Challenges at Pillar

At the heart of the Pillar test system were four Dell 2950 servers with 8GB RAM. This horsepower was no longer enough, and Colin was planning to augment or replace them with quad core, dual processor machines to bring performance of his test system up to snuff. Hardware was tapped out.

The virtual environment was also problematic. Deploying fleets of virtual machines from scratch for each new

test was time-consuming. Clean-up needed to be double-checked and the VM memory footprint would outgrow the resources on the VM host. The system was simply difficult to manage, prone to sprawl, and couldn't deliver enough high-volume CIFS and NFS traffic—it could not adequately test the unique features of the Axiom.

The challenges with tools included needing to write scenarios from scratch, write wrappers around existing code, and revise manually any time a variable changed. Steve Downer, director of test engineering, found that his test case volume was growing exponentially, and his team was increasingly racing around trying to get as many tests running as he could in the shortest time possible.

### Enter SwiftTest

In the fall of 2009, Colin was seeking load testing tools and found SwiftTest. The SwiftTest 3000 performance and depth and breadth of features fit Pillar's needs just at the time Colin was considering a significant investment in new white box servers to expand his testbed.

Instead of new servers, a SwiftTest 3000 was added to the Pillar environment. Not only were they quickly able to increase testing productivity, Colin and Steve found the SwiftTest team to be

---

*“To truly understand Axiom’s performance, we have to simulate customer client machines that are always getting faster and more aggressive. SwiftTest can simulate the faster and badder client—it’s great, like turning on a firehose.”*

*- Colin Hutcheson, Test Architect*

---

More than a simple load generator, the system provides protocol-level control and semantic definition that enables the broadest possible range of test cases.

The SwiftTest 3000 features 8 singly configurable 1G ports and an additional 1G management port. It supports multiple authentication types and originates both NFS and CIFS traffic, simulating a great number of clients and scenarios at once. It easily handles multiple scenarios with varying lists of commands within each of 5 protocols (CIFS v1, v2, and NFS v2, v3, v4)—all running at the same time without operational overhead.

The SwiftTest 3000 is the clear choice for companies in the storage industry, because SwiftTest knows storage better than any other test tool vendor. It is the industry's elite testing tool due to the SwiftTest focus on storage-specific features; high-volume traffic handling, customer-centric graphic interface, and prebuilt tests and wizards.

### The Results

High volume CIFS and NFS traffic is now putting realistic stress on the Pillar Axiom with no more VM housekeeping headache. Having the SwiftTest 3000's 8 ports on different networks allows for a more representative picture of how production clients would address a storage device. Simulating connection scaling no longer requires adding in sets of clients, which had been impossible before with Pillar's VM client setup.

Control of test cases and scenario set-up is easy and fast for Steve's team, who can use sample tests and wizards, quickly assemble actions, and change parameters such as number of users—all in a graphical UI.

They now have easy access to stats that were only usually available via network traces and client OS data. And they get real-time results as part of the scenario instead of requiring extra triage cycles. Even customer log scenarios are easily reverse engineered for targeted tests.

Now the Pillar test team can perform tests that exceed the demands their customers put on the Axiom. And they can do it with greater control, in less time, and using fewer resources.

---

*“Aggregation of results is our enemy. We want to see those results from individual components as they all have differing effects and interactions on a storage array. With SwiftTest we can trace, track, and compare anything—all the protocol-level details we want.”*

*- Steve Downer, Director of Test Engineering*

---

responsive and flexible, with in-depth knowledge of storage and testing.

### The SwiftTest 3000

The SwiftTest 3000 is a high-performance load-generating system designed for comprehensive storage network testing.

### Test Development Environment

The SwiftTest 3000 includes a graphical application, the SwiftTest Test Development Environment, that provides flexible, granular configuration options and test results analysis.

Testers can now find problems, modify parameters, and quickly see results using the SwiftTest interface. Problem isolation is fast and accurate, leading to much-appreciated efficiency in working with development engineers.

---

### Why Buy the SwiftTest 3000

- **Time savings in executing multiple test suites.**  
*Single-screen, graphical control of test tools enables fast settings changes.*
  - **Ease of test case creation and control at the protocol level.**  
*Granular visibility and results are available for individual components.*
  - **Consistent, repeatable results.**  
*Test clean-up is automatic. Start tests at the same point with a clean slate.*
- 



3255 Scott Blvd., Suite 2-101 • Santa Clara, CA 95054-3013  
408.716.7117 • [www.swifttest.com](http://www.swifttest.com)