

ARCHITECTING AN AUTOMATED TEST ENVIRONMENT

By Larry Cantwell and Amit Rao,
ONPATH Technologies

In today's product development environments there is a driving need to test products and services faster while increasing test coverage and product quality.

These two seemingly conflicting objectives are being accomplished by network and application product companies through well-architected automated test environments. The ROI for implementing test lab automation is compelling in terms of product development time saved, efficient sharing of test equipment, and last but not least in reduction of costly field defects. This paper will provide discussion and guidelines on the key components that must be considered when architecting and implementing an automated test lab environment.

WHY TEST LAB AUTOMATION?

Before getting into the design of an automated lab scheme, let's spend a moment taking a look at some key drivers of automation trends in development organizations:

- The pure scale of global networking today along with state-of-the-art technologies such as Service Oriented Architecture (SOA) and Virtualization have driven exponential growth in the scale of the lab test setups required for product development companies.
- Product and service lifecycles have shortened significantly in recent years requiring companies to reduce time to market significantly without sacrificing functionality or quality.
- Product development has become a global effort for most organizations as offshore resources are combined with traditional stateside teams to optimize development costs and enable 24x7 test operations.
- Typical test configurations have expanded from 10's of devices to 1000's of devices in many cases. This drives a requirement for organizations to share equipment across multiple development efforts and be able to dynamically reconfigure test network topologies on a daily basis.
- Mergers and acquisitions have given companies many disparate labs with gaps in test equipment coverage.

TEST LAB AUTOMATION CHALLENGES

With the above trends in mind, test lab automation has become an essential aspect in most product development and service deployment scenarios. However, companies face several challenges in implementing such a solution

- Multiple test and verification routines
- Poor equipment utilization and control
- Limited remote access and sharing
- Simulating equipment and cable failures
- Manual and error-prone processes
- Expensive tools and reduced budgets
- Long setup and teardown times
- Range of networks, speeds and media

THE LAB AUTOMATION SOLUTION

In many organizations, dedicated projects have been put together to focus exclusively on developing and maintaining the automated test environment. The overall ROI on this investment is generally very compelling in terms of the following:

- Increased capability to share test equipment, software and devices, resulting in a significant reduction of development capital spending.
- Significantly (orders of magnitude!) reduced test lab setup and tear down time resulting in reduced operational expenses and reduced time to market.
- Significant reduction in product field defects through:
 - Increased scalability testing
 - Expanded hardware, software and application interoperability testing
 - Increased regression testing coverage
 - Increased ability to do stress and corner case testing

The architecture and implementation of an automated test environment breaks down into several primary elements. These elements are as follows and are depicted in *Figure 1*:

- A Physical Layer Switching infrastructure (PLS) to provide automated (dynamically configurable) connectivity between devices and applications under test. This acts as the plumbing or glue that ties the automated test environment together.
- A soft test framework that connects and integrates the elements of the automated test environment through common messaging and APIs supported by the various test products.
- A top level graphical management application that provides a single point of control over the automated test lab environment.
- The actual network elements, endpoint devices and applications to be tested.

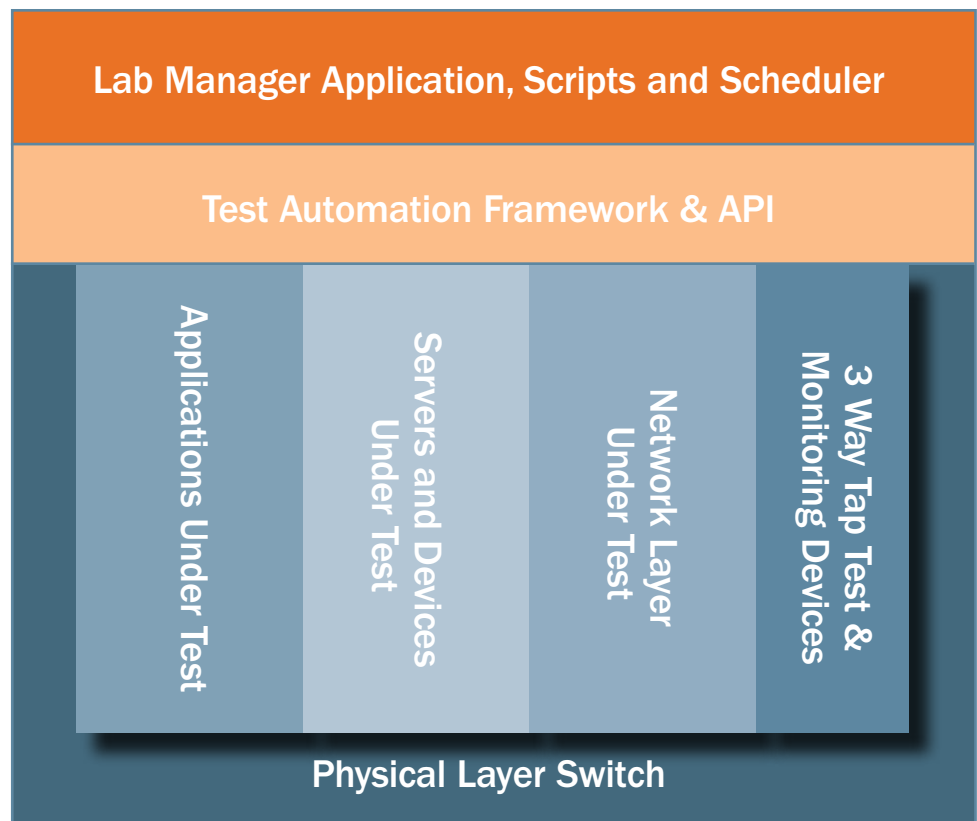


FIGURE 1. Key automated test lab components

Automated Physical Layer Test Infrastructure

The first element in architecting an automated test environment is automating the physical infrastructure itself. Whether a company is producing software applications, middleware or network and storage hardware equipment, one reality holds true: the connectivity of all of the elements (both hardware and software) required to test the end product must be able to be dynamically configured into a target test configuration.

Traditionally, test labs may have spent days, weeks, or in some cases months manually cabling test devices, monitoring equipment, servers, etc. together through patch panels and point-to-point connections in preparation for a specific product test cycle. Aside from the day-to-day price paid in time to market relative to the critical development path as well as inefficient use of both human and test equipment resources, this approach often resulted in an over-encumbering 'rats nest' of cables either below the raised test floor or in overhead cable raceways. This approach is also prone to human error, and often times, it takes significant cycles to bring up a given test configuration once cabled. The physical patching depicted in the graphic in *Figure 2* illustrates this problem and how it is addressed in an automated test lab environment.

The key element that must be included in the automated physical layer test infrastructure is a physical layer switch (PLS) or switches. The scalable physical layer switching infrastructure is the backbone of the automated test lab. It effectively eliminates the need to move cables around as test topologies are set-up and torn down. The notion here is to cable up your test lab 'plumbing' once and then modify it dynamically on the fly at the click of a button from a client terminal that exists anywhere in the world.

Why use a physical layer switch instead of a layer 2/3 switch? The vastly accepted approach is to use a physical layer switch because you will want your physical interconnect within the automated test lab to act like an 'automated wire'. In many cases, layer 2/3 switches will change the network dynamic in terms of blocking, throughput, upper layer protocols and services. In some cases the layer2/3 switches themselves may be the devices under test so you certainly would not want to use the device itself for interconnect of the test network.

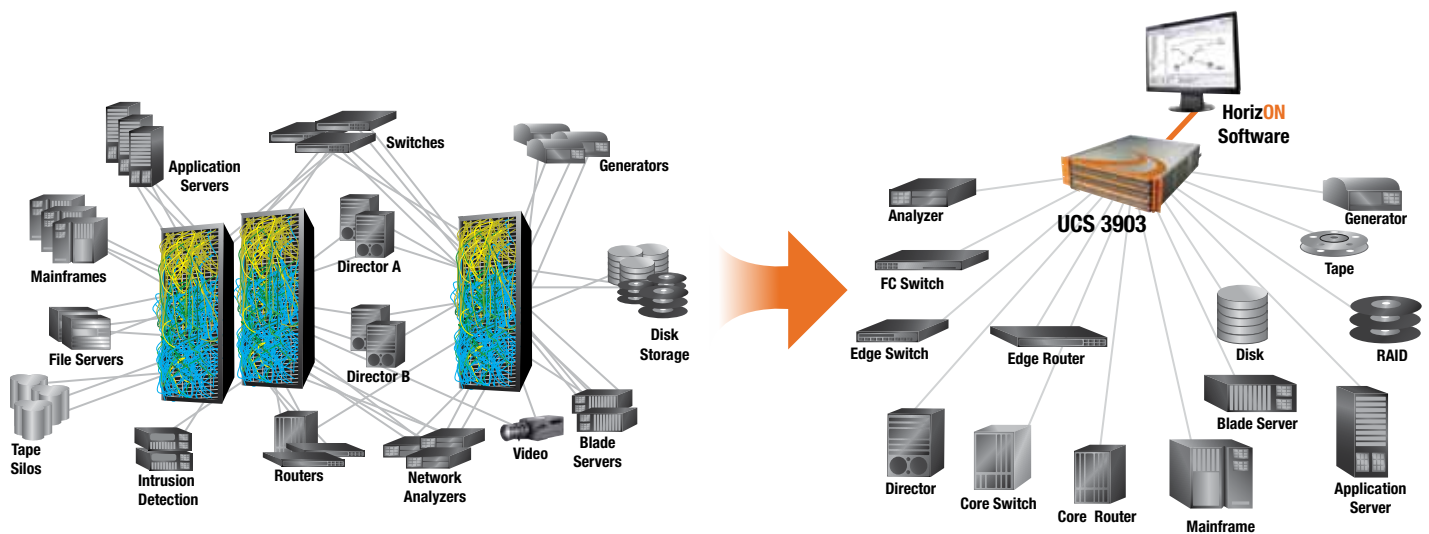


FIGURE 2. Automated lab environment transition

PLS Attributes

In this section of the document, we will further explore the use of the PLS to automate devices, test equipment and raise application connectivity as the fundamental building block of an automated test lab environment.

SCALABILITY AND NON-BLOCKING

First and foremost, the PLS must provide scalable, non-blocked connectivity for the automated test lab environment. Typical automated test lab environments today have 1000+ test devices, any number of which can be required for a given test topology. Since PLS's operate at the physical layer (almost as an automated patch panel in the most basic sense), it is important to select a PLS solution that can scale to this magnitude of test device connectivity in a fully non-blocked fashion. Often, 1st generation deployments will fall into a trap of building the automated physical test infrastructure out of smaller PLS products. This can lead to blocking and scalability issues down the road.

ABILITY TO APPLY TEST TOPOLOGIES REMOTELY

The PLS solution must allow each test stakeholder to be able to configure, schedule and non-disruptively overlay test topologies dynamically and remotely. Since one of the key value propositions of the automated test environment is the capability to locate test devices and equipment in a central location and share it across a global development team, this functionality must be provided such that the test engineer can schedule and apply their test topologies from a remote graphical client regardless of their position on the planet relative to the actual equipment they require for a given test.

ABILITY TO CENTRALIZE USER OF ANALYZERS AND PROBERS TO PASSIVELY 'TAP' THEM INTO CIRCUITS

A fundamental attribute of the automated lab physical layer connectivity is the ability to centralize costly monitoring and probing equipment such that it becomes a pool of resources available to all test stakeholders. When selecting the PLS component of your automated environment, ideally this functionality is best made available through a three-way passive tapping functionality. Many PLS units offer this capability to passively tap probe and test equipment into any circuit on demand. This approach is optimal because it eliminates the active disruption that test equipment can impose on a circuit as well as enabling the ability to share test equipment efficiently and in a virtual fashion where the test equipment need not be local to the site that is having the problem.

ABILITY TO PROVIDE SECURE USER ACCESS

A fundamental attribute of the automated lab physical layer connectivity is the ability to centralize costly monitoring and probing equipment such that it becomes a pool of resources available to all test stakeholders. When selecting the PLS component of your automated environment, ideally this functionality is best made available through a three-way passive tapping functionality. Many PLS units offer this capability to passively tap probe and test equipment into any circuit on demand. This approach is optimal because it eliminates the active disruption that test equipment can impose on a circuit as well as enabling the ability to share test equipment efficiently and in a virtual fashion where the test equipment need not be local to the site that is having the problem.

ABILITY TO DIAGNOSE AND INJECT ERRORS

A test network by definition is prone to errors. These errors can occur in the form of physical issues that can be experienced in any network. Given the nature of the environment, errors can also be caused by the ‘developmental’ devices under test. Uncertainty in the environment is often exacerbated by the fact that developmental devices and applications are present. Often, organizations that are embarking on a test automation pilot project will experience some hesitation on the part of the development and test engineers due to concerns that automation deployments can be unstable particularly as they grow in scale. For these reasons, it is important that the PLS be able to monitor and detect low level physical errors and circuit degradation in a real time fashion. Further, the extent to which the PLS can aid in fault isolation and distinguish where in the test network that the error is occurring can provide significant benefit and should be a consideration in choosing a PLS solution. Finally, it is often desirable in test environments to be able to inject low level data errors (CRC, encoding, etc) in order to see how the test network and device under test (DUT) is able to react. A device that can provide this advanced capability is also highly desirable.

Automated Test Management Framework

Once the PLS ‘plumbing’ is properly architected, the other fundamental component of the Automated Test Lab is the software framework and user management interface that will be the control plane of the lab. This section of the paper will explore this aspect of the architecture.

COMMON APIs AMONG TEST EQUIPMENT AND APPLICATIONS

First and foremost, when considering the architecture for your automated test lab, one must ensure a common fit between the products you purchase and/or develop to build that automated test environment. It is typical for an automated test environment to be sourced with soft and hard components from multiple test automation vendors. Initiatives such as the TesLA are geared toward allowing these companies to work together to define common APIs so that management and control of the automated test network is done seamlessly through messaging and databases that share standardized formats. These initiatives will help to expedite the deployment of automated test infrastructures and minimize the amount of custom framework and infrastructure work required. It is important when architecting an automated test environment to ensure that the various components from the automation vendors are following these initiatives and that they will integrate together with relative ease. Taking care toward this end can reduce lab automation deployment complexity considerably.

TEST AUTOMATION MANAGEMENT APPLICATION

A test automation management application (TAMA) must be selected or developed. There are several applications of this type on the open market. Ideally, this application will fit into the overall lab automation architecture in such a way that it provides a single point of control for the infrastructure, test equipment and for running the test jobs themselves. It is required that the test framework itself has a well-defined control plane that interconnects all of the elements in the automated test framework to the TAMA. This is typically achieved by building an I.P management framework that is reliable and distinct from any test paths. This section describes some of the key components of the Test Automation Management Application (TAMA).

ABILITY TO BUILD AND APPLY TEST TOPOLOGIES

One thing that is common to any test environment is that there is a pool of devices and applications that can be made available for the test users to build a test topology. These devices can include servers, storage, switches and routers, protocol analyzers and applications. The TAMA must keep track of the inventory of all of these devices and applications and provide the ability for test users to graphically draw the test topology they wish to have by dragging and dropping devices from the pool and then interconnecting them as desired through the PLS infrastructure. A simple test topology is shown in *Figure 3*.

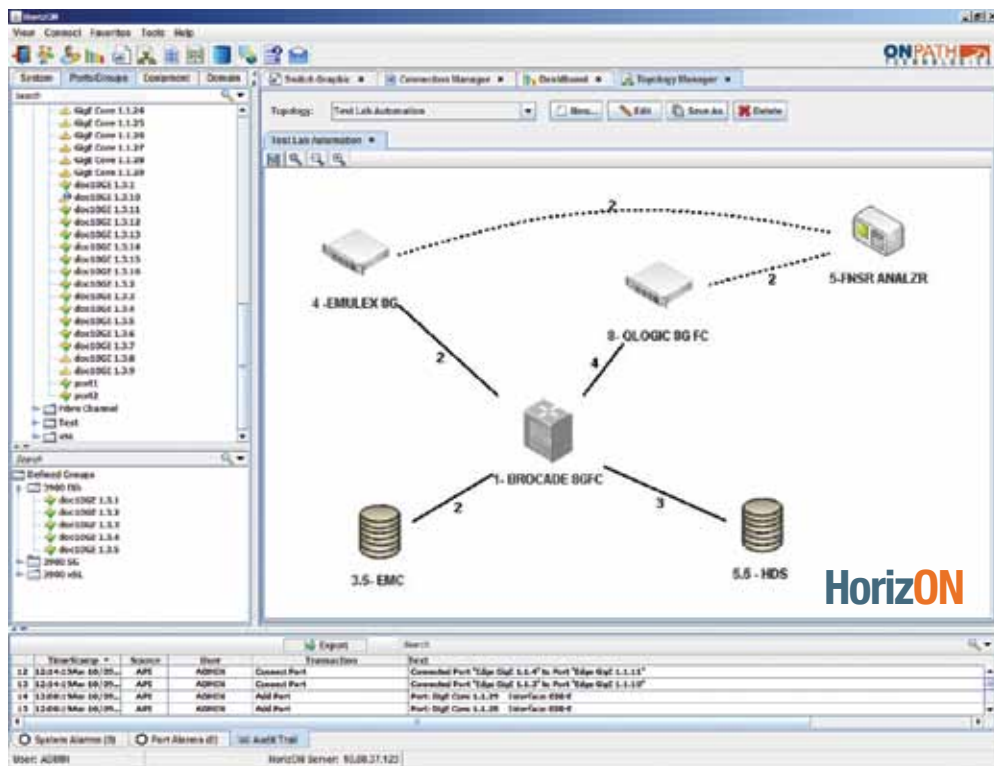


FIGURE 3. Sample test topology

ABILITY TO SCHEDULE AND RESERVE TEST TOPOLOGIES INCLUDING DEVICE APP. AND THEIR CONNECTIVITY

Another fundamental capability of the TAMA is that it must provide the capability to schedule and reserve the equipment involved in a test topology. Often times, this will involve the automated intelligence to look into the future to reserve the required resources at a time furnished by the test user or perhaps at a time when the required resources become available. This carries with it the need for intelligence within the reservation application to invoke test job queuing, resolve resource conflicts and schedule resources based on various sub-algorithms such as earliest possible available. The most common graphical user

interface used for this operation is a calendar based front end client that provides an hour by hour, day by day view of test lab resources and their availability. This is analogous to scheduling human resources for a meeting on an application such as MS Outlook Calendar. The reservation system will use the non-blocked PLS architecture described above to provide seamless connectivity for the test topology.

DEVICE DISCOVERY AND INVENTORY

The TAMA must be aware of the devices available in the test network pool. This should be furnished via a concise hierarchical inventory of devices that are known entities in terms of capabilities and attributes. Ideally, the TAMA will be able to auto-discover all devices in the test pool via the out of band test management framework. The resources and their attributes are stored in database that can be easily accessed, queried and augmented through a graphical client.

DASHBOARD SHOWING RESOURCES AVAILABLE AND THEIR STATUS

From a user perspective, the usability of the Automated Test Lab will be driven by how easily devices and applications within the environment can be reserved and utilized. The TAMA should provide a top level dashboard view that clearly shows the available devices and their alarm status as well as utilization. It is highly desirable to have both a real-time and historical view of test device utilization. The real-time view will allow individual users to easily visualize what resources are available for them to build their test topology. The historical utilization view will allow test managers to easily see which devices within the automated test lab are over-utilized and under-utilized for the purpose of capacity planning and test network optimization.

ABILITY TO PROVIDE TEST QUEUING

Many test labs have built-out sophisticated test regression scripts over the course of time. As individual developers make changes to software and hardware modules, it is desirable to run system level regression with the subject module changes in isolation before releasing those changes to be merged in with other developers modules. This approach restricts the scope of debugging down to the module that changed to allow developers to quickly isolate what they did to 'break' the system. Test automation makes this objective much more achievable because system regression scripts can be selected ala carte and run in sequence on individual module changes by queuing up the test jobs and running them 24x7. This eliminates the requirement for test engineers to be present 24x7 to manually set up and execute regression tests. The end result is much more efficient use of test resources and of course early isolation of bugs.

COMMON TEST SCRIPTING LANGUAGE

In most test lab environments, batteries of test scripts have been developed over time. These scripts are generally the functional test that is being performed. Often times, these scripts configure network devices, invoke test pattern generation, create fault scenarios, etc. Some common scripting languages are PERL and PYTHON. These scripting languages are compatible with most common operating systems and TAMAs.

ABILITY TO 'PUSH' TEST CONFIGURATIONS OUT TO THE TEST NETWORK DYNAMICALLY

The TAMA must be able to configure a given test topology, connect the pieces together via the PLS and finally 'push' the test configuration out to the subject test topology. This 'push' includes things like configuring routers, applications, adapters and operating systems. In more recent years, many companies are also starting to use virtualization software to greatly increase the scalability of their automated test lab. An example of this would be pushing multiple images of an application under test out to a pool of servers for a given test. The next test user that comes along may be able to use virtualization applications to push additional test jobs out to the same server farm without impacting other test jobs that are already running.

CONCLUSION

In this document we have explored the key considerations when architecting an Automated Test Lab. Careful consideration to the architectural components discussed in this paper will give rise to a scalable automated test infrastructure that will enable organizations to enhance productivity, optimize sharing of expensive tools and scale the scope of testing in the face of reduced budgets and staff.



ABOUT ONPATH TECHNOLOGIES

ONPATH Technologies is the leading provider of scalable connectivity and monitoring solutions for high-performance networks. With a history spanning over 25 years, ONPATH is a spin-out of Brocade Communications' physical layer switch business. ONPATH's Universal Connectivity System™ and HorizON™ Software deliver an advanced platform that automates and secures data center and test infrastructure to help network managers conserve time, increase utilization, and save money compared to manual patching or complex mesh switching architectures. Our patented switching technology and advanced software deliver the industry's most scalable, secure connectivity solution with the most in-depth view of your network. Scalable from 8 to 4,096 non-meshed ports, independent of your speed or protocol requirements, ONPATH delivers the flexibility and security required by today's data centers and the investment protection necessary for those of tomorrow. ONPATH currently has over one million installed ports throughout Fortune 1,000 and Government customers.

For more information on how ONPATH can help you with your initiatives, contact us today for an engineering application review or network consultation.

2000 Lincoln Drive East
Marlton, NJ 08053
609.518.4100
info.request@onpathtech.com
www.onpathtech.com