Choosing the Right Service Virtualization Solution

EXECUTIVE SUMMARY

There are many factors to consider when choosing a service virtualization solution. These factors range from choosing an appropriate starting point based on team size to selecting the right deployment model that will accommodate your teams scaling.

The purpose of this article is to outline the various deployment options and ownership models, contrast the differences between them, and provide a launching point for you to select the correct service virtualization solution to fit your organization's needs.

REASONS FOR ADOPTING SERVICE VIRTUALIZATION

Typically, organizations adopt service virtualization for the following reasons:

AGILE

As organizations make the move to agile, development timeframes become considerably compressed. We need to be able to creatively develop our application without constraints. By leveraging service virtualization, we can accomplish more in this reduced timeframe because we have the support of a virtual environment to remove agile bottlenecks incurred when dependent APIs are unavailable or unreliable.

CONTINUOUS TESTING

Since agile planning considerably accelerates the software development process, modern application-testing teams "bear the burden" of this decrease in the cycle time between release iterations. This begs the question: how, in this shortened cycle, do you test the release candidate effectively? Continuous testing says, "let's test on demand," and instead of waiting until the evening or the end of the sprint to run the tests, you test the application continuously. Parasoft tools are designed with this in mind. They have intuitive automation APIs and integration with tools like Jenkins and Docker. This allows you to build an infrastructure today that will support your continuous testing efforts going forward.

SHIFTING LEFT

Enabling teams to start their testing efforts earlier has significant advantages, regardless of the development methodologies you subscribe to. Service virtualization allows development teams to create prototype services for scoping. These same artifacts can be leveraged by test to design and execute tests now.

PERFORMANCE

It is difficult to isolate performance issues in a large, constantly-evolving environment. Service virtualization allows you to simulate realistic performance behavior of specific components and reduce them from the overall environment baseline to properly gage specific performance. Furthermore, it is hard to provision adequate hardware that performs realistically under load. Using virtual services you can simulate any application or network performance characteristics without having the actual (usually expensive) infrastructure in place.

PARASOFT®

Perfecting Software

HOW TO GET STARTED WITH SERVICE VIRTUALIZATION

There is a misconception that service virtualization is a gigantic behemoth, that takes significant investment in time and capital to get started. In reality, service virtualization is as complicated as you want it to be. Companies can download a free service virtualization solution (Parasoft Virtualize Community Edition) and have services up and running in a matter of moments. For example, check out this video: Testing when the API Isn't Ready. You can get going fast with Virtualize Community Edition and with simplicity – the effort is the same as getting started with an open-source tool. You can make simple mocks and stubs that will enable you to take advantage of all the modernization areas above – of course, that is where the similarity to open-source ends.

With Virtualize, you can go way past simple mocks and stubs. The solution is designed with scaling in mind. Your SDLC landscape is going to change, and Virtualize will grow with you. It will have the functionality you need at each stage, as you move beyond simple services into creating intelligent, data-driven simulations from traffic recordings and desire more complex customization options such as performance and dynamic data. As your deployment needs grow, Virtualize can grow with you. You can create assets from a centralized browser or on local desktops but the collaboration will remain the same, and when it comes time to ramp up for performance testing, the solution will easily scale.

FREE VS. PAID SERVICE VIRTUALIZATION

If the answers to the questions in the column to the right are all "yes, I need the simple stuff," then you can start with the free Parasoft Virtualize Community Edition. With the Community Edition (CE), you get a free tool that enables you to quickly create HTTP-based virtual assets that can range in complexity from simple stubs to full, dynamic data-driven assets. With native support for WSDL, Swagger, and RAML, it is very easy to take Web Services and REST descriptions and 'at the click of a button' create assets. Additionally, because CE is built on top of the core Parasoft Virtualize platform, you can seamlessly leverage assets created in CE with the Virtualize Professional Desktop. This opens up the door to reuse all your hard work across additional protocols, advanced tooling, and greater capacity (50k hits/day).

The Virtualize Professional Desktop also comes with Parasoft SOAtest, the powerful API and web testing solution, already built in. So regardless of where you start, your test and virtual assets can be used across any Parasoft solution as you continue to scale and grow your testing efforts.

HOW ARE THE SERVICES GOING TO BE CONSUMED?

Once the initial virtual services are created, there are different consumption methodologies we can position depending on a series of contributing factors: team size, frequency of access, and level of testing maturity (automated vs. continuous). Consumption activities are distinctly different from creation activities because of network topology. An organization will apply different variants of the architecture, depending on how and from where the services will be invoked.

Development

Development teams take the best advantage of local topology. Developers like to have "private Islands" or local environments that have all of the components they need. This gives them creative freedom to develop whatever, whenever. The initial desire for a development team is to have local deployments of Virtualize running, while sharing their virtual services to source control. This technique works with teams of relatively small sizes, about 10 or less. These teams should get started using the free Virtualize Community Edition. The solution allows each developer to consume 11,000 hits a day (more than sufficient for development), and it has all of the tools and capabilities that developers need to create services for today's modern systems.

CHOOSING BETWEEN A FREE OR PAID SERVICE VIRTUALIZATION SOLUTION

When choosing between a free or paid virtualization solution, you need to ask yourself some questions about what you are emulating:

- What protocols and message formats are needed? is it just Web Services or REST XML or JSON over HTTP or do you need more MQ, JMS, MQTT, ISO8683, FIX, special custom format?
- How much control do you need over the performance characteristics? Simple response timing or do you need to model load dependent behavior?
- It is 'simple static stub?' Or do the responses need to be data driven or dynamic from incoming payload?
- How often are you going to want to access the virtual services per day (>11k hits/day)? How much throughput are you expecting for your services (> 11tps)?

As you start making these virtual services available to the larger development community, you need to start thinking about a consolidated consumption server. The size of the server will be based on the amount of services you have deployed and the performance you expect from them. A Runtime Server will allow 11 transactions per second. For a medium to large development team, this should support daily ad hoc testing as well as nightly runs. Assets that were created will be checked into source control along with the code, and deployed to the runtime server for overnight execution.

As the team starts to grow in size, or as additional teams come on board, there are two migration paths that can be taken.

1. Get a beefier consolidated server such as a performance edition
2. Scale horizontally with Parasoft cloud-based deployments

While the consolidated Performance Server allows for ease of environment assembly through automation, the challenge associated with a single powerful machine is one of congestion. Not from the perspective of incoming traffic, but in terms of configuration activity. As more and more development teams deploy virtual services to the same server, naming conventions and path configurations start to become complex. In some cases, the time that you save by not configuring your applications dynamically to point to transient virtual infrastructures is lost because of the time that it takes recovering from collision caused by multiple teams tied to the same endpoint. By taking advantage of on-demand servers – whether it's on premise, through Docker, or in the cloud through the AWS and Azure solutions – development teams can create multiple "private islands" as needed to support their effort.

Testers

Testers have the same complexity challenge illustrated above. The major difference with development teams is that they will get to this level of congestion much quicker. The reason for this is due to regression testing. Testing teams have to keep different versions of virtual services active to allow for backwards compatibility and regression testing. Additionally, different development teams might desire to consume the same virtual asset with vastly different data sources. As testing teams start to grow configurations, scaling becomes the primary driver for their deployment structure.

Parasoft recommends, as testing teams are starting to consume virtual assets, that they deploy all of their virtual assets to a consolidated server. This can take two paths:

1. Dedicated or dynamic centralized runtime servers per silo
2. Dedicated or dynamic cloud deployment machines

A simple runtime server has a low startup cost and can service unlimited hits a day. Depending on how each silo will be consuming their virtual services, this may be more than sufficient. As additional silos want to consume virtual services, the organization can simply horizontally scale by grabbing new runtime server licenses through Parasoft. The same strategy can be streamlined by taking advantage of the AWS and Azure on-demand offerings without involving Parasoft. Theoretically the consumption allowance is the same, but the cost can be reduced because the cloud servers only charge for runtime. A typical deployment environment such as that would have a centralized runtime library server that could double as the stage server and, as applications need to be tested, the required virtual services can be deployed through the service virtualization thin client interface. The cloud-based infrastructures can be provisioned prior to deployment, utilized, and subsequently decommissioned. This allows for rapidly deploying and destroying virtual environments to support the testing teams and can easily be scaled as required.

Performance

Performance testing is one of the areas where service virtualization has its greatest advantage. Since you have the ability to model realistic performance behavior for your virtual services, the performance teams can leverage artifacts that have been created by the development and testing teams, and simply apply their application performance SLA to the virtual services profile. This means that performance teams don't have to take a lot of time creating virtual services. Their consumption activities, however, are distinctly different.

In the case of performance, it doesn't really matter what the team size is. What becomes important instead is the expected transactions per second. For performance testing, a team requires a performance server. These machines can handle ~2000 transactions per second, depending on the quantity and complexity of the virtual services deployed.

In the early stages of performance testing, as you have less than 500 virtual assets, if the aggregate expected performance does not exceed ~2000 transactions per second, one performance server should be able to accommodate this behavior characteristic. As the quantity and complexity of the virtual services increase, you will need to add additional performance servers. Parasoft Virtualize supports clustering, so it's as easy as adding an additional performance server and directing all traffic at the load balancer.

For peak performance, we recommend that your virtual servers are hosted by cloud providers such as AWS or Azure. This makes reconfiguring the underlying hardware trivial. There is a cost consideration for MIPS on and off the cloud provider, so one must consider the frequency from where the calls will originate. Organizations are beginning to move their development activities to the cloud, so this cost should be reduced over time.

CREATION

Anyone can build virtual assets, but what do the asset-creation workflows look like? There are recommended adoption strategies based upon which team you are targeting for the overall effort.

Development-Focused

Development teams are best positioned to generate the initial assets because they have a deep understanding of how the application and its dependencies interact with each other. Developers tend to be early adopters, only bringing in new tools if they are functionally sound, and tend to shy away from solutions that require commercial licensing.

For these reasons, development-focused teams can start with the Virtualize Community Edition, which is available for free and has been tailored to modern technology standards such as REST (RAML, SWAGGER).

Test-Focused

On the other side of the coin, organizations have a lot to gain by having testing teams create, and more importantly extend, virtual services. Testing teams should start by creating their assets on a Virtualize Professional Desktop, which includes more exotic protocols, message types, and advanced workflows like Parasoft Change Advisor.

In addition to creation activities, testing teams benefit from collaboration. The Virtualize thin client interface allows teams to create and share artifacts from their browser. Behind the scenes, the artifacts are stored in source control and hosted on a shared centralized virtualization server. As testing teams start to adopt continuous testing, they will have a relationship between the test cases and the virtual assets. Both artifacts will be checked out, on demand, as a part of test execution. The Virtualize ecosystem can facilitate this relationship and seamlessly deploy the artifacts as a part of the team's existing automation workflows.

Center of Excellence

Center of excellence models start becoming a requirement if the number of individual asset creators across an organization begins to exceed 100 users. The center of excellence team is the keeper of best practices when creating virtual services. They also hold the governance model and administer the continuous testing infrastructure. They are an enablement team that will reach out to other teams as required, provide them access to the software, teaching them how to build an initial round of virtual services, and supporting them through complex virtual asset builds.

When your organization has matured to a center of excellence model, you will typically have your asset creators using the Virtualize thin client interface, as it is the easiest way to distribute the software. The thin clients will be connected to a centralized staging server. This centralized server can start out as a runtime as it is your sandbox. All assets created initially go to this stage machine so that they can be validated and approved. Once approved, they are checked into source control and promoted via automation to remote Virtualize servers, depending on how many silos the consumption activities have been federated to – the number of these servers depends upon the organization's desire for environmental independence.

SUMMARY

Because software is rarely a one-size-fits-all solution, choosing a solution that will scale with your organization is the most important factor for tool selection. Because Parasoft Virtualize Community Edition is free, requires no commitment, and will allow you to scale in many directions after you get started, it is the best suggestion we can make, so you can quickly get started with a deployment solution that will integrate with your current process and position you for the future.

www.parasoft.com

Parasoft Headquarters:
+1-626-256-3680

Parasoft EMEA:
+31-70-3922000

Parasoft APAC:
+65-6338-3628

PARASOFT®
Perfecting Software