SERVICE VIRTUALIZATION:

Reduce the time and cost to develop and test modern, composite applications

Business white paper

Table of contents

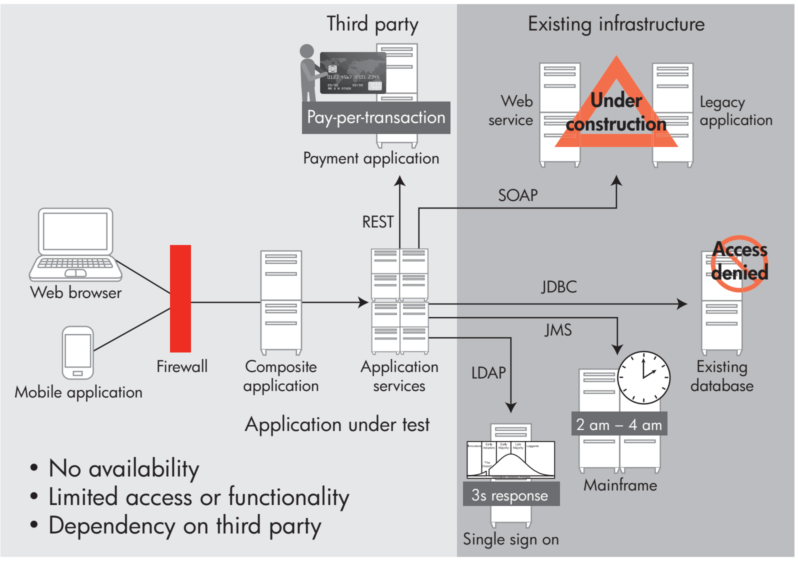


Figure 1: The challenges of building and testing a complex, composite application in a distributed environment include dealing with services that are unavailable, services with limited access, and third-party dependencies.

Why you need service virtualization

Today, everything is mobile, connected, interactive, immediate and fluid. To gain competitive advantage in the face of rapidly changing technology and business dynamics, enterprises must accelerate time to market. In an age of "instant" expectations, organizations have to be able to respond quickly and effectively to changing business needs.

To help the business achieve these goals, application teams need to deliver quality software on time and on budget. Software developers and testers must speed up application projects and lower the costs of development integration and testing. However, in the effort to reduce costs and meet tight deadlines, application teams often need to make tradeoffs on the expected level of performance and functionality, which can lead to a sacrifice in application quality.

To do this rapid, agile development, many organizations are moving to composite applications which are made up of smaller units of functionality that can be developed and tested in quickly achieved increments. This approach promotes agility because the smaller units of functionality, or services, can be developed in parallel, accessed from third parties, and reused in multiple projects. One downside to this approach is that the entire process can slow down—and become more costly—if there are dependencies on services that are not available when developers and testers need them.

HP Service Virtualization software allows developers and testers access to limited or unavailable services in a simulated, virtual environment. This helps application teams to lower costs and reduce testing times by finding defects earlier in the application life cycle when they are easier, faster and less expensive to fix. It helps improve quality by enabling quality assurance (QA) teams to test what otherwise couldn't be tested. It also helps isolate problems that are based on dependencies between services in composite applications. This helps significantly reduce delays and manage the costs and complexity of composite application development and testing.

The challenges of composite applications

Composite applications are composed of services, which are applications, systems or components that expose their functionality by remotely accessible interface in order to allow integration with other applications and systems. These may be web services, commonly known as Simple Object Access Protocol (SOAP) or Representational state transfer (REST), and are usually accessible over HTTP/HTTPS.

Unavailable services

There are many situations in which a service may be unavailable for development or testing efforts. Some applications may be unavailable because they haven't been developed yet or are only partially developed. Some may be time-limited because they are sandbox services or scheduled services shared by many departments. Some specialized applications may be unavailable because they are behind a firewall. Some third-party services, such as a payment service or shipment service for a shopping application, may have pay-per transaction access fees that are cost-prohibitive for development and testing purposes. And some services that use older technologies for application integrations (such as Common Object Request Broker Architecture (CORBA), Distributed Component Object Model (DCOM), and remote Enterprise JavaBean (EJB)/Remote Method Invocation (RMI) running on top of TCP/IP and other network layers) may be available but difficult and time-consuming to configure.

For services that are available for functional testing, there may be restrictions or outright bans on performance testing. This is true in particular for services delivered via the cloud and third-party sandbox services.

Functional and performance testing challenges

Unavailable services can have a huge impact on an application's quality and schedule. If the application team chooses to wait until all services are available, delivery of the application may be significantly delayed. If the team moves forward without access to certain services, they may see inconsistent and inaccurate results that will compromise application quality. In addition, the team may only be able to perform client-side testing, which is insufficient for assessing the quality of an application.

In terms of load testing, if a third-party service is unavailable, the testing team cannot find out what will happen to the performance of the application under test (AUT) if the service does not meet its service-level agreement (SLA) for performance. Performance testers may need to tune the application by adjusting the SLA to speed up the overall transaction time, but that is nearly impossible to do if they do not have access to the third-party service. In addition, performance testers need to test the entire business process through multiple performance characteristics, such as peak and non-peak hours, but this is difficult to do when services are time-limited or unavailable.

Issues with services that are available

Sometimes services are available, but they may slow down test execution because they are used heavily by other teams, set up on slow hardware, or have performance defects. A service may require complicated provisioning of test data to multiple distributed systems before test execution, which can delay testing. Or a service might be available, but a defect in the service might block completion of an end-to-end process test and thus prevent defect detection in later parts of a multi-step business process.

Even when services are available, they may have dependencies that are not available. For instance, the AUT may be dependent on a third-party payment service which itself may be dependent on a different third-party service such as a credit card processing service. A service may be dependent on other in-house teams or may require data that is unavailable. In each of these cases, the QA team may not understand the dependencies or may not have a way to work around the unavailable dependencies.

HP Service Virtualization

HP Service Virtualization helps organizations address the challenges of developing and testing composite applications when they have limited access to services or the services are unavailable. By simulating the unavailable services in a virtual environment, HP Service Virtualization enables functional and performance testing to take place earlier in the application lifecycle, saving time and reducing the cost of developing and testing modern applications.

Developers can dramatically speed up prototyping by quickly creating functional mockups and using simulated models of unavailable or constrained services. Developers can also reuse emulated services from other projects or use local simulation copies of third-party services to jump-start the development process.

Testers can use HP Service Virtualization to overcome dependency barriers, eliminate wait cycles during test preparation, and reduce the time required for complete end-to-end testing. Testers can modify the performance behavior of simulated services to test otherwise hard-to-achieve limit conditions. HP Service Virtualization provides support for large-scale load testing environments to enable performance testers to optimize application performance and find the best deployment configuration.

HP Service Virtualization is easy to use. Wizards help make initial implementation, configuration and setup of virtual services simple. HP Service Virtualization's unique data-oriented approach enables the creation of a virtual service with no scripting. A new virtual service is created by entering known information about the service using a simple spreadsheet interface, by emulating the behavior and performance of an existing service through a unique "learn and simulate" capability, or by linking to external data sources. These easy-to-use modeling techniques help to speed up virtual environment preparation and simplify maintenance in rapidly changing environments.

Organizations that use HP Service Virtualization can get the most out of their virtual services by sharing and reusing them across stakeholders and teams. Broad reuse is enabled by integrations with HP market-leading application lifecycle management software products, including HP LoadRunner, HP Performance Center, HP Unified Functional Test, HP Service Test, HP Quality Center and HP Application Lifecycle Management.

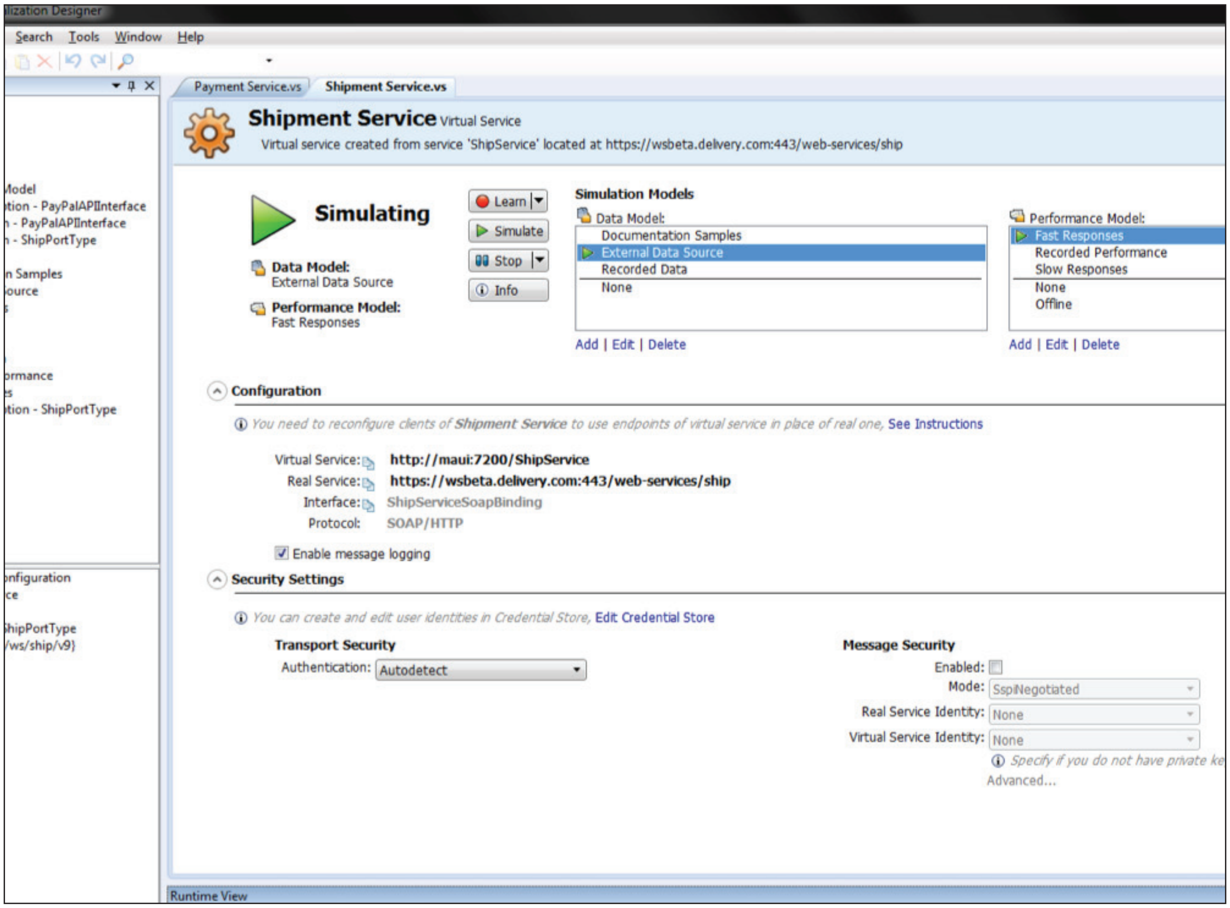
Key capabilities in HP Service Virtualization

HP Service Virtualization helps developers, performance testers, and functional testers reduce the time and risk involved in building and testing complex, composite applications.

For developers

Developers who work on today's composite applications often find themselves in a situation where they need to integrate with a service that is unavailable or difficult or expensive to access. This frequently happens when they need to perform a unit test of the larger application. Instead of facing delays, developers can use HP Service Virtualization to quickly create a model or simulated version of the service. Developers create a model using information about the service that they already have on hand in the service's documentation. HP Service Virtualization provides a wizard to create a skeleton of the model. The developer then populates the model with rules, parameters and messages from the documentation—using an interface similar to popular spreadsheet tools. With no scripting required, developers can quickly and easily create and maintain virtual service models.

Figure 2: HP Service Virtualization eliminates dependencies in composite application development and testing by simulating functional and performance behavior of otherwise unavailable application components.



Shipment Service Virtual Service

Virtual service created from service 'ShipService' located at https://wsbeta.delivery.com:443/web-services/ship

Simulating

Data Model: External Data Source

Performance Model: Fast Responses

Simulation Models

Data Model: Documentation Samples, External Data Source, Recorded Data, None

Performance Model: Fast Responses, Recorded Performance, Slow Responses, None, Offline

Configuration

You need to reconfigure clients of ***Shipment Service*** *to use endpoints of virtual service in place of real one,* See Instructions

Virtual Service: **http://maui:7200/ShipService**

Real Service: **https://wsbeta.delivery.com:443/web-services/ship**

Interface: ShipServiceSoapBinding

Protocol: SOAP/HTTP

☑ Enable message logging

Security Settings

You can create and edit user identities in Credential Store, Edit Credential Store

Transport Security

Authentication: Autodetect

Message Security

Enabled: □

Mode: SspiNegotiated

Real Service Identity: None

Virtual Service Identity: None

Specify if you do not have private ke

Advanced...

As development progresses, the developer can tweak the model to test different scenarios. HP Service Virtualization provides an accuracy metric to measure how close the virtual service simulation is to the original or modeled service behavior.

HP Service Virtualization helps developers create integrations to other application components and expose virtualized components to the QA team. Developers can also make virtualized services available to other teams for reuse in additional projects.

For functional testers

Functional testers need to validate the functionality of newly developed applications. HP Service Virtualization enables them to start their testing earlier in the project cycle, before all services are available.

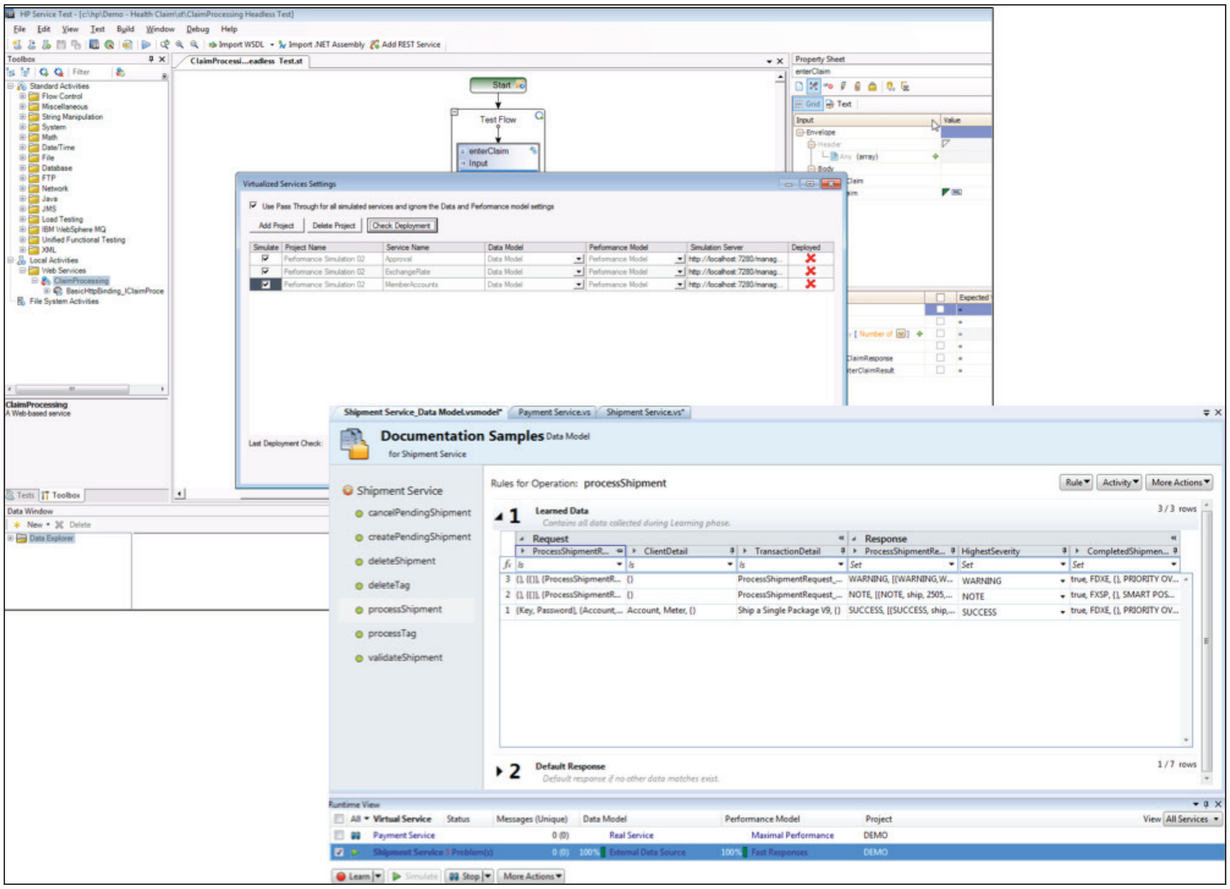
Functional testers create simulation models in HP Service Virtualization by emulating the behavior of an existing service, recording its communications with the AUT (through application programming interfaces) and end user (through the user interface). They then expose the models from HP Service Virtualization in HP Unified Functional Testing to test the behavior of the composite application or in HP Service Test to test at the service level. When testing an end-to-end flow, functional testers may discover that the test fails because of a defect in a real service. By swapping in a virtualized service to replace the broken one, the testers can work around the defect to continue the testing process without delay.

Functional testers also need to perform regression testing to validate upgrades and changes in applications and their components. If a component has changed, functional testers can use HP Service Virtualization to verify that its original behavior did not change and that the new version will not break any of the communications in the larger composite application.

Functional testers need to test composite applications from the end-user point of view with different test data for different scenarios. They also need to understand the dependencies and boundaries of the components that make up the composite application under test.

HP Service Virtualization integrates with HP Test Data Management software to enable functional testers to use sanitized and preprocessed data from various data sources to create simulation models of virtualized services.

Figure 3: HP Service Virtualization allows script-less, spreadsheet-like definition of virtual service behavior and control of virtual service models from functional testing tools.



Virtualized Services Settings

☑ Use Pass Through for all simulated services and ignore the Data and Peformance model settings

Add Project | Delete Project | Check Deployment

Simulate	Project Name	Service Name	Data Model	Peformance Model	Simulation Server	Deployed
☑	Peformance Simulation 02	Approval	Data Model	Peformance Model	http://localhost:7280/manag...	✗
☑	Peformance Simulation 02	ExchangeRate	Data Model	Peformance Model	http://localhost:7280/manag...	✗
☑	Peformance Simulation 02	MemberAccounts	Data Model	Peformance Model	http://localhost:7280/manag...	✗

Rules for Operation: processShipment

1 Learned Data — 3 / 3 rows
Contains all data collected during Learning phase.

	Request			Response		
	ProcessShipmentR...	ClientDetail	TransactionDetail	ProcessShipmentRe...	HighestSeverity	CompletedShipmen...
fx	Is	Is	Is	Set	Set	Set
3	{}, {[]}, {ProcessShipmentR...	{}	ProcessShipmentRequest_...	WARNING, [{WARNING,W...	WARNING	true, FDXE, {}, PRIORITY OV...
2	{}, {[]}, {ProcessShipmentR...	{}	ProcessShipmentRequest_...	NOTE, [{NOTE, ship, 2505,...	NOTE	true, FXSP, {}, SMART POS...
1	{Key, Password}, {Account,...	Account, Meter, {}	Ship a Single Package V9, {}	SUCCESS, [{SUCCESS, ship,...	SUCCESS	true, FDXE, {}, PRIORITY OV...

2 Default Response — 1 / 7 rows
Default response if no other data matches exist.

Runtime View

Virtual Service	Status	Messages (Unique)	Data Model	Performance Model	Project
Payment Service		0 (0)	Real Service	Maximal Performance	DEMO
Shipment Service	3 Problem(s)	0 (0)	100% External Data Source	100% Fast Responses	DEMO

View All Services

Using HP Service Virtualization, models can be created by recording the communication between the application under test and the real service. The integration with HP Test Data Management enables the extraction and creation of test data sets that can be used in a shared test environment. Using these capabilities combined with the test automation features of HP Application Lifecycle Management, functional testers can automate client-side testing using specific data and virtualized components. The testers can select which virtualized profiles to use during automated tests.

For instance, an application for an international market may require testing with a different data set for each supported language. Testers create a data set for each language using HP Test Data Management. They use HP Service Virtualization to create virtualized profiles to populate the data sets into the simulation of the virtualized components. This setup enables HP Unified Functional Testing to automatically run each of the appropriate tests, switching profiles to walk through each of the data sets. This automation increases test coverage and accelerates the validation of the AUT's functionality across each language.

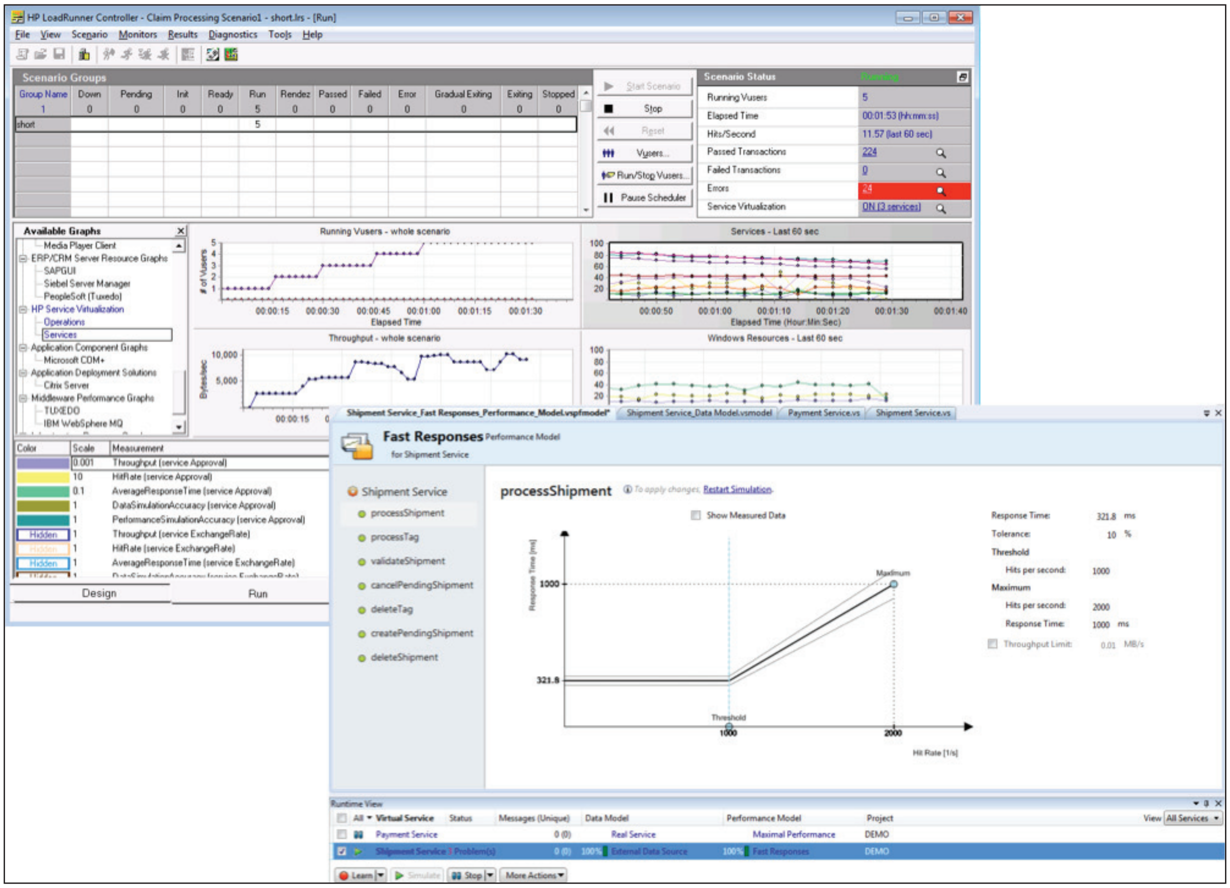
Integration of HP Service Virtualization with HP Application Lifecycle Management enables functional testers to consolidate results and report to management. HP Application Lifecycle Management and HP Quality Center automate the reporting process with test coverage reports that help keep management and other stakeholders informed of testing status.

For performance testers

Performance testers need to understand how composite applications will perform under load, but this can be difficult to do when services are unavailable or difficult to access. This is a more urgent problem for performance testers because many services that are available for functional testing may not always be available for performance testing, particularly for services offered via the cloud. In addition, even when services are available, performance testers need to know how they will perform under a variety of conditions. And it can be challenging for performance testers to isolate the performance behavior of the AUT.

HP Service Virtualization enables performance testers to quickly create a virtual service model through an easy-to-use recording process. By recording the

Figure 4: Users can graphically edit performance characteristics of virtual services to load test composite applications under a variety of conditions and access them during load testing scenarios.



HP LoadRunner Controller - Claim Processing Scenario1 - short.lrs - [Run]

Scenario Groups

Group Name	Down	Pending	Init	Ready	Run	Rendez	Passed	Failed	Error	Gradual Exiting	Exiting	Stopped
1	0	0	0	0	5	0	0	0	0	0	0	0
short					5							

Scenario Status: Running

Running Vusers	5
Elapsed Time	00:01:53 (hh:mm:ss)
Hits/Second	11.57 (last 60 sec)
Passed Transactions	224
Failed Transactions	0
Errors	24
Service Virtualization	ON (3 services)

Color	Scale	Measurement
🟪	0.001	Throughput (service Approval)
🟨	10	HitRate (service Approval)
🟩	0.1	AverageResponseTime (service Approval)
🟫	1	DataSimulationAccuracy (service Approval)
🟦	1	PerformanceSimulationAccuracy (service Approval)
Hidden	1	Throughput (service ExchangeRate)
Hidden	1	HitRate (service ExchangeRate)
Hidden	1	AverageResponseTime (service ExchangeRate)

Fast Responses Performance Model
for Shipment Service

processShipment — To apply changes, Restart Simulation.

□ Show Measured Data

Response Time: 321.8 ms
Tolerance: 10 %
Threshold
Hits per second: 1000
Maximum
Hits per second: 2000
Response Time: 1000 ms
□ Throughput Limit: 0.01 MB/s

Runtime View

□	All	Virtual Service	Status	Messages (Unique)	Data Model	Performance Model	Project
□		Payment Service		0 (0)	Real Service	Maximal Performance	DEMO
☑		Shipment Service	1 Problem(s)	0 (0)	100% External Data Source	100% Fast Responses	DEMO

communication between the AUT and a real service, HP Service Virtualization creates a model of the service that includes performance characteristics. Once the model is built, the performance tester can adjust the performance characteristics of the model to enable testing under different load conditions. Performance characteristics, such as response time, tolerance and scalability, are adjusted using an intuitive user interface. The performance characteristics are viewed as a graph that can be tweaked to simulate different conditions. There is no scripting required.

Performance testers use this technique to create multiple models for one service in order to simulate different load scenarios. They then expose the models from HP Service Virtualization in HP LoadRunner or HP Performance Center in order to load test the composite application. This enables an end-to-end test of an entire business process across different performance characteristics, such as peak load and non-peak load hours. This gives performance testers the ability to get as close as possible to a real load testing scenario for complex, composite applications and test various limiting performance scenarios.

If a real service is available but has a defect that prevents the completion of a multi-step process, performance testers can use HP Service Virtualization to model the service without the defect. Using the virtualized service enables the performance testers to proceed with testing the end-to-end process without having to wait for the defect to be addressed.

Similarly, if a service in the test environment is slowing down test execution due to defects, installation on slow hardware, or heavy usage by other teams, performance testers can use HP Service Virtualization to avoid delays. By creating a model of the service and changing its performance parameters, testers can speed up testing iterations.

HP Service Virtualization was designed to handle large-scale performance testing to emulate the real-world production environment. It allows performance testers to simulate negative behavior or a breakdown of dependent systems. By enabling performance testers to change backend performance characteristics, it allows them to optimize application performance and find the best deployment configuration.

Key takeaways

Building and testing today's composite applications is challenging. When services that underlie the applications are unavailable, the risks to software quality and on-time delivery increase dramatically. HP Service Virtualization helps developers and testers mitigate these risks in the following ways:

Reduce software business risk: HP Service Virtualization makes it possible for developers and testers to simulate the real-world behavior and performance conditions of unavailable or restricted access services without massive amounts of coding. This enables testing to occur earlier and more often in the application lifecycle when it is easier and less expensive to fix defects.

Minimize delays: HP Service Virtualization enables QA teams to deliver test results faster across more testing scenarios. With the flexibility to modify upstream test loads and virtual downstream system responses, QA teams can test more thoroughly against possible operating conditions without delaying the process. HP Service Virtualization enables faster and easier problem isolation, replication and identification—even when QA and development teams do not have access to all components in a live environment for debugging.

Increase software quality: HP Service Virtualization enables broader test coverage by enabling QA teams to change the functional and performance responses of virtual components. This gives QA teams control over testing backend systems that are not accessible for certain testing scenarios.

Meet SLAs for performance and quality: HP Service Virtualization helps application teams ensure that complex distributed applications achieve quality and performance goals by testing SLAs at all levels of the application.

Maximize team productivity and support globalization: By eliminating the problems caused by limited availability services, HP Service Virtualization provides testing teams with 24x7 testing capabilities even against shared and scarce resources.

For more information, visit **www.hp.com/go/sv** or the community page at **www.hp.com/communities/SV**.

4AA3-6375ENW, Created September 2011