



Adobe[®] LiveCycle[®] Enterprise Suite on VMware[®] Infrastructure

June 2010

DEPLOYMENT AND TECHNICAL CONSIDERATIONS GUIDE

vmware[®]



Table of Contents

Introduction.....	1
VMware Infrastructure and Adobe LiveCycle ES2 Overview.....	1
Adobe LiveCycle ES2 Overview.....	1
VMware Infrastructure 3 and VMware vSphere 4.....	2
LiveCycle ES2 Architecture and Deployment Strategy.....	3
Testing Process and Results.....	4
Testing Methodology and Overview.....	5
2-vCPU Tests:.....	5
Hardware Configuration.....	5
ESX Host Configuration.....	6
Virtual Machine Configuration.....	7
Workload Used.....	7
Results Observed.....	8
Reference Architecture.....	20
Workgroup and Smaller-Size LiveCycle Deployments.....	20
Departmental and Medium-Size LiveCycle Deployments.....	21
Enterprise and Larger-Size LiveCycle Deployments.....	22
Deployment Best Practices.....	23
Licensing.....	23
Technical Support.....	23
Conclusions.....	24
Resources.....	25

Introduction

This document provides basic guidance to those interested in running Adobe® LiveCycle® Enterprise Suite (ES2) on VMware® Infrastructure 3 or VMware vSphere™ 4. It provides basic information on the architecture of LiveCycle ES2, as well as the value of utilizing the VMware platform to deploy LiveCycle ES2 solutions. The document also discusses results of recent testing conducted jointly by VMware and Adobe, characterizing the performance and functionality of LiveCycle ES2 on VMware infrastructure version 3.5 update 3 and vSphere 4.0. Finally, the document outlines some best practices for utilizing the two product sets together in a datacenter.

VMware Infrastructure and Adobe LiveCycle ES2 Overview

VMware and Adobe have worked together to provide official support of LiveCycle Enterprise Suite (ES2) solutions on VMware virtual infrastructure. Both Adobe and VMware have active virtualized LiveCycle ES2 deployments running on VMware virtual infrastructure and there are many customers that have also successfully deployed LiveCycle ES2 in VMware virtualized environments.

Adobe LiveCycle ES2 Overview

Adobe LiveCycle ES2 software can help you extend the value of existing back-end systems by enabling developers to build and deploy applications quickly and easily, and by empowering business users to manage application environments based on their specific needs. With Adobe LiveCycle ES2, you can make it easier for people to interact with information through intuitive user experiences, improve efficiencies through business process automation, and enhance customer service through personalized communications management. With LiveCycle ES2, you can

- **Deliver applications that people want to use** – Build and deploy easy-to-use, personalized, and interactive applications that extend processes beyond your firewall in an engaging way. LiveCycle ES2 helps companies deliver easy-to-use, personalized, and interactive applications that connect to automated processes, improving customer self-service and employee productivity. People interact with these applications using the freely available and ubiquitous Adobe Flash® Player and Adobe AIR® clients, enabling organizations to increase adoption rates and user satisfaction.
- **Improve operational efficiency** – Boost your profit margins and reduce total cost of ownership through better management and automation of critical business processes (with tools such as Adobe Reader and LiveCycle Workspace). By connecting people and disparate back-end systems with a single, streamlined solution, organizations can increase process efficiencies and visibility.
- **Communicate effectively with customers and partners** – Respond quickly to customer demands, complex market conditions, and evolving regulatory requirements with personalized and more secure communications. LiveCycle ES2 helps companies engage customers, improve operational efficiencies, and reduce costs by automating the generation of personalized electronic and paper documents. By targeting communications more effectively, companies can improve customer satisfaction to develop and retain profitable, long-term relationships.

LiveCycle ES2 leverages free and ubiquitous client runtimes so that your applications are easily accessible and immediately usable by customers, constituents, partners, and suppliers.

LiveCycle Foundation provides the underlying server capabilities that enable the deployment, execution, and management of your LiveCycle ES2 applications. LiveCycle Foundation consists of:

- **Service container** – Provides the common runtime environment to support all LiveCycle modules and associated services.
- **Foundation components** – Enable integration with common IT infrastructures.
- **Administration tools** – Help you monitor and control server activity and easily deploy and manage LiveCycle modules.
- **Central repository** – Allows you to store information and assets shared by all aspects of your LiveCycle ES2 implementation.

LiveCycle ES2 applications flexibly integrate services provided by a number of different modules that are available, which provide various Internet application services, document services such as forms automation, security, and communications management, and business process management. Examples include LiveCycle Data Services to streamline the development of rich Internet applications with a high-performance server framework, LiveCycle Rights Management and Digital Signatures to manage usage rights to sensitive information in files and integrate digitally signed and certified documents within critical business processes, and LiveCycle Output that dynamically generates personalized documents on demand.

NOTE: For a complete list of modules available for use in Adobe LiveCycle ES2 applications, refer to product information on the Adobe web site, located at: <http://www.adobe.com/products/livecycle/>

Adobe provides a number of different tools enabling developers to easily create and modify existing LiveCycle applications via model-driven development to deliver RIAs, forms, and personalized documents:

- **LiveCycle Workbench ES2** – Included in every LiveCycle ES2 installation, LiveCycle Workbench ES2 is an integrated development environment (IDE) that allows developers, designers, and business analysts to work together collaboratively on form design and process modeling.
- **LiveCycle Designer ES2** – Also included in every LiveCycle ES2 installation and working in conjunction with LiveCycle Workbench ES2, LiveCycle Designer ES2 provides robust and intuitive tools for creating interactive forms and personalized document templates.
- **Adobe Flash Builder™** – You can easily integrate Adobe Flash Builder software (now in beta release) with the LiveCycle Workbench ES2 environment. Using Flash Builder, you can develop your own RIAs or create custom additions to forms created in LiveCycle Designer ES2 and viewed with Adobe Flash Player software.

VMware Infrastructure 3 and VMware vSphere 4

VMware's leading virtualization solutions provide multiple benefits to IT administrators and users. VMware virtualization creates a layer of abstraction between the resources required by an application and operating system, and the underlying hardware that provides those resources. A summary of the benefits this abstraction layer provides include the following:

- **Consolidation:** VMware technology allows multiple application servers to be consolidated onto one physical server, with little or no decrease in overall performance.
- **Ease of Provisioning:** VMware virtualization encapsulates an application into an image that can be duplicated or moved, greatly reducing the cost of application provisioning and deployment. With the addition of VMware vCenter™ management software, administrators can leverage templates to create golden masters of virtual machine environments that can be duplicated or moved. This process can greatly reduce the cost and time of application provisioning and deployment as well as make it easier to scale LiveCycle ES2 deployments.

- **Manageability:** Virtual machines may be moved from server to server with no downtime using VMware vMotion™. vMotion also simplifies common operations such as hardware maintenance performed during planned downtime.
- **Availability:** VMware High Availability (HA) can shorten unplanned downtime and provide higher service levels to an application. In the case of an unplanned hardware failure, VMware HA restarts any affected virtual machines on another host in a VMware cluster.

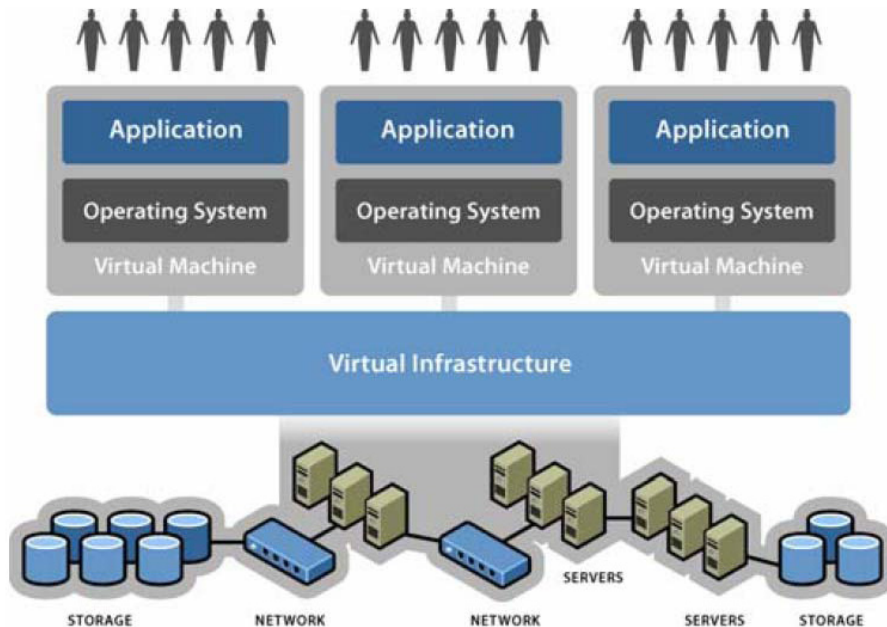


Figure 1. VMware vSphere Virtual Infrastructure

LiveCycle ES2 Architecture and Deployment Strategy

LiveCycle ES2 is an enterprise SOA platform that lets developers easily integrate a myriad of enterprise services into a smaller number of composite services by means of orchestration tools. LiveCycle itself provides a large number of document-centric services.

LiveCycle ES2 is a J2EE application that is supported on three popular J2EE application server platforms. These are IBM WebSphere, Red Hat JBoss and Oracle WebLogic. Figure 2 below provides a diagram of the LiveCycle ES2 system architecture and the various components and services provided in a LiveCycle platform deployment.

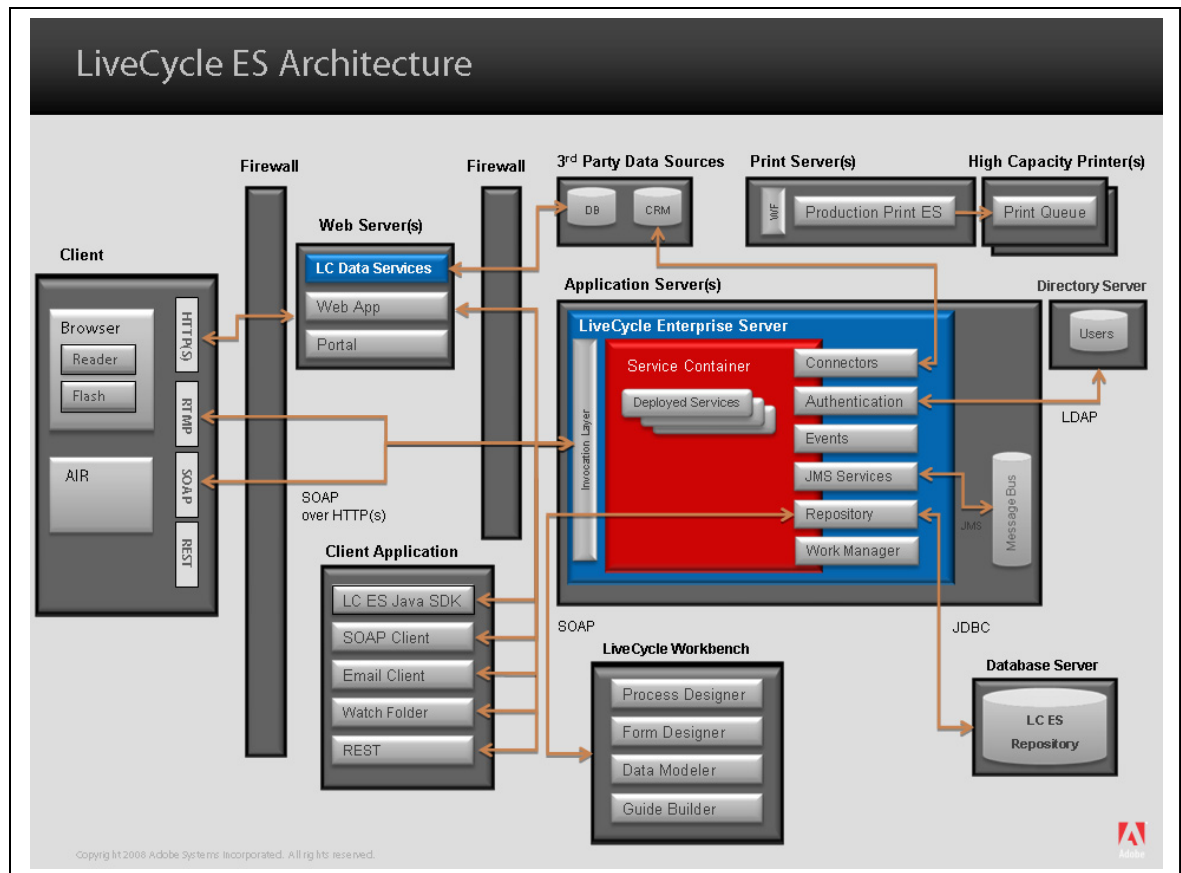


Figure 2. LiveCycle ES Architecture

Here is some other key information about the LiveCycle ES2 Platform:

- All services provided by LiveCycle ES2 have IIOP, REST, SOAP, Flex Remoting, Watched Folder and E-Mail endpoints for programmatic invocation. REST and SOAP endpoints are available to .NET applications.
- Authentication and authorization is achieved by integration with corporate directories that support LDAP v3 (ES2 SDK available). These include Microsoft Active Directory, SunONE Directory Server, Novell eDirectory and IBM Tivoli Directory Server.
- Data persistence is provided by a Relational Database Management System (RDBMS). Supported RDBMS are Oracle, Microsoft SQL Server, IBM DB2 and Oracle MySQL.
- Integration functionality for mail systems and JMS systems is built-in.
- Connectors are available for Enterprise Content Management (ECM) software from IBM (FileNet and Content Manager), EMC (Documentum) and Microsoft (SharePoint).

Testing Process and Results

VMware and Adobe conducted a number of different tests to characterize the performance of LiveCycle ES2 on VMware virtual infrastructure and to also test the operation of various VMware product features in a virtualized LiveCycle environment. The following sections summarize the configurations tested, workloads used in testing, and the results.

Testing Methodology and Overview

The primary objectives of testing were to determine Adobe LiveCycle performance characteristics and show that Adobe LiveCycle can scale linearly and be run more resiliently on VMware virtual infrastructure. (We also wanted to ensure that customers could successfully deploy their mission-critical LiveCycle applications in the same virtual environments.) Testing also validated the operation of VMware features including vMotion, VMware DRS, VMware Fault Tolerance (FT), and VMware HA with LiveCycle deployments in a virtual environment.

Two groups of tests (with a total of eleven individual tests) were run to collect results in different testing scenarios. The first group of tests was run on one vCPU; the second group was run on two vCPUs.

1-vCPU Tests:

- **Test 1: Baseline Test** – to obtain basic measurements about how LiveCycle performs on VMware virtual infrastructure.
- **Test 2: VMware vMotion** – to determine the impact of vMotion on a live running LiveCycle virtual machine.
- **Test 3: VMware DRS** – to observe how the performance of LiveCycle virtual machines is optimized using VMware DRS.
- **Test 4: High Availability Test 1** – to observe how high availability is provided by VMware HA without restarting the LiveCycle application.
- **Test 5: High Availability Test 2** – to observe how high availability is provided by VMware HA and also observe what happens when the LiveCycle application is restarted to restore full capacity.
- **Test 6: Fault Tolerance** – to observe how high availability is provided by VMware FT without using application cluster HA capabilities or functionality.

2-vCPU Tests:

These tests scaled up resources from the 1-vCPU test to see how the LiveCycle virtual machine behaves with one additional vCPU.

- **Test 7: Baseline Test** – to observe the resource utilization with a larger-sized LiveCycle virtual machine.
- **Test 8: VMware vMotion** – to determine the impact of vMotion on a live-running LiveCycle virtual machine.
- **Test 9: VMware DRS** – to observe how the performance of LiveCycle virtual machines is optimized with two vCPUs.
- **Test 10: High Availability Test 1** – to observe how high availability is provided by VMware HA with two vCPUs.
- **Test 11: High Availability Test 2** – to observe how high availability is provided by VMware HA and also observe what happens when the LiveCycle application is restarted to restore full capacity with two vCPUs.

Hardware Configuration

The testing environment consisted of two ESX servers that host a two-node Adobe LiveCycle cluster in both 1-vCPU and 2-vCPU configurations to perform tests as described in the previous section. On the same ESX hosts, there is also a load balancer virtual machine and a MySQL virtual machine that respectively direct incoming requests to LiveCycle virtual machines in round-robin fashion and serve as the data store for the LiveCycle application.

The following diagram provides details about the setup of ESX host servers and virtual machines used to perform the one-vCPU and two-vCPU tests described earlier.

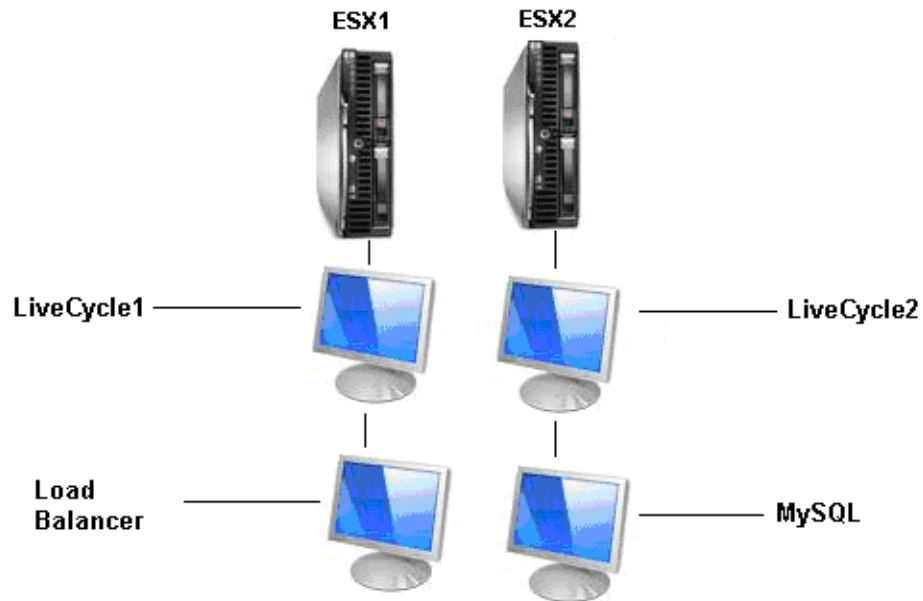


Figure 3. Hardware Configuration for LiveCycle ES2 Testing

ESX Host Configuration

The following table describes the configuration of ESX host servers and storage in the LiveCycle test configurations.

Table 1. ESX Host Hardware

HARDWARE	CONFIGURATION
Server	Two HP Proliant BL 460c G6 servers. Each server is equipped with: <ul style="list-style-type: none"> • 2.8 GHz dual quad-core Intel Xeon E5560 • 48 GB RAM
Storage	NetApp FAS 3020: <ul style="list-style-type: none"> • Aggregate of 21 disks • RAID DP • 4 Gbps fibre connectivity • 10K RPM disks • FC protocol

Virtual Machine Configuration

The following table describes the configuration of virtual machines running on ESX host servers in the LiveCycle test configurations.

Table 2. LiveCycle Virtual Machine Configuration

VIRTUAL MACHINE	HARDWARE CONFIGURATION
Cluster - Two LiveCycle virtual machine instances (on Suse Linux Enterprise Server 10 with JBoss 4.2 as J2EE Application Server)	<ul style="list-style-type: none"> • 2 vCPUs • 4 GB memory • 1 Ethernet card • 30 GB storage (1 x 70 GB) • vmdk only
One Load Balancer (on Apache web server)	<ul style="list-style-type: none"> • 1 vCPU • 1 GB memory • 1 Ethernet card • 30 GB storage (1 x 50 GB) • vmdk only
One MySQL instance	<ul style="list-style-type: none"> • 1 vCPU • 2 GB memory • 1 Ethernet card • 30 GB storage • vmdk only

Workload Used

The workload used for testing included common critical tasks performed by a typical LiveCycle application to ensure the majority of key LiveCycle components were involved in our test. The number of concurrent users remained the same for both 1-vCPU and 2-vCPU tests to see how the system responded with varying resources being assigned to the virtual machines. The load was generated using the LoadRunner program running on a separate machine from the two ESX host server machines.

Adobe Technical Marketing (eTech) has developed a benchmark orchestration of several LiveCycle ES services chained (orchestrated) together as a single service. It is primarily an index of the performance of the processors on the System Under Test (SUT). This means that the number, clock speed and other capabilities of the processors will have significant impact on the reported transaction throughput numbers.

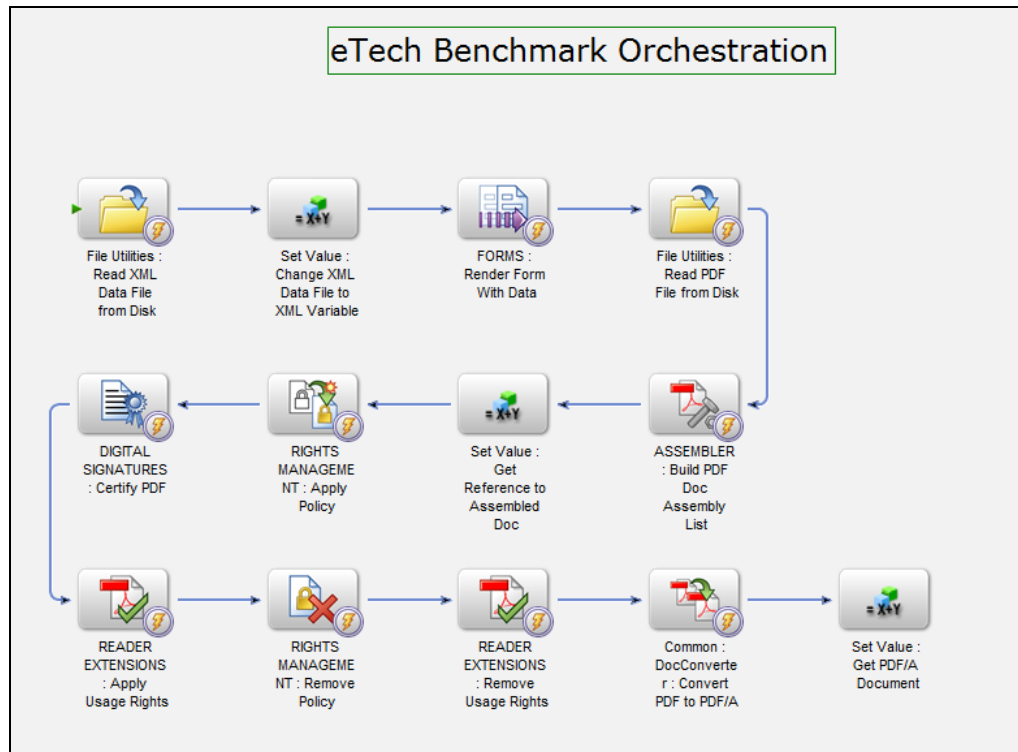


Figure 4. eTech LiveCycle Benchmark Orchestration

As the illustration of the benchmark orchestration in Figure 6 shows, the following actions are performed serially:

- Read an XML file from the server file system. This XML file contains form data.
- Set the contents of this XML file to a process variable of type 'XML'.
- Pass this data to the Forms ES component of LiveCycle along with a form template (.XDP) from the LiveCycle Repository (database).
- Keep the resulting PDF form in a process variable of type 'document'.
- Read another PDF file from the server file system.
- Using Assembler ES, combine the previously created PDF form and the PDF into a single PDF.
- Apply a Rights Management ES policy to the combined PDF.
- Certify this PDF with a Digital Signature using a Document Signing Credential kept in the LiveCycle Trust Store.
- Apply Reader Extensions rights to the PDF.
- Remove the Rights Management Policy that was previously applied.
- Remove the Reader Extensions rights that were previously applied.
- Convert the PDF to the PDF/A archival format and keep the result in a process variable of type 'document'.

Results Observed

Based on the results of our testing, we found that virtual deployments of Adobe LiveCycle scale very linearly as more resources are assigned to virtual machines. Also, LiveCycle runs well in conjunction with VMware HA, vMotion, VMware DRS and VMware FT to provide higher resilience and efficiency to customer applications. So, customers can have the peace of mind to know they can deploy their mission-critical LiveCycle production applications in a VMware virtual infrastructure environment and get the performance and reliability they need.

Test 1: Baseline Test (One vCPU)

The first baseline test was performed with two LiveCycle application virtual machines configured with one vCPU each. The goal was to test the number of transactions processed.

Table 3. Test 1 Baseline Results

VM	CPU			Memory		Network		Disk				Kernel		Guest	
	(%)			(mb)		Packets/sec		I/O (sec)		Latency (ms)		Latency (ms)		Latency (ms)	
	Avg	Max	Rdy	Avg	Max	Recd	Txd	Avg	Max	Avg	Max	Avg	Max	Avg	Max
LiveCycle 1	93	100	0	3995	4055	243	112	5	41	1	23	0	0	1	23
LiveCycle 2	100	100	0	3991	4055	422	110	8	32	1	19	0	0	1	19
LB	1	2	0	420	481	392	115	1	7	1	33	0	0	1	33
MySQL	1	6	0	944	1065	254	163	23	48	2	13	0	0	2	13

Average response time was 10.81 seconds with CPU and Memory utilization above 90%. The test configuration was an optimal setup for two concurrent users.

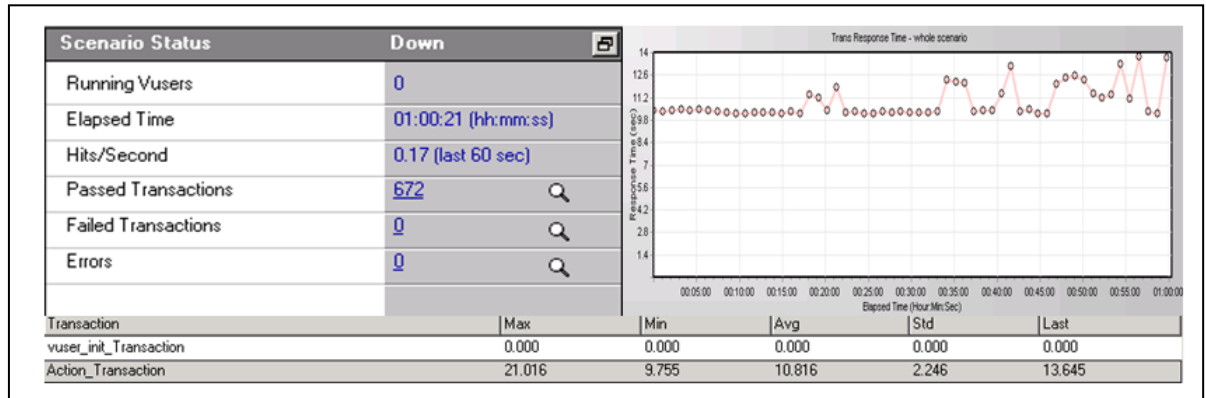


Figure 5. Graph of Test 1 Baseline Results

Figure 5 shows a total of 672 transactions were processed with no failures reported.

Test 2: VMotion (One vCPU)

This test validated VMware vMotion functionality and recorded the transaction response time. Two 1-vCPU LiveCycle instances were configured on separate ESX hosts.

Table 4. Test 2 Results

VM	CPU			Memory		Network		Disk				Kernel		Guest	
	(%)			(mb)		Packets/sec		I/O (sec)		Latency (ms)		Latency (ms)		Latency (ms)	
	Avg	Max	Rdy	Avg	Max	Recd	Txd	Avg	Max	Avg	Max	Avg	Max	Avg	Max
LiveCycle 1	100	100	0	3645	4014	225	108	6	42	0	2	0	0	0	2
LiveCycle 2	93	100	0	3850	4014	236	97	9	30	1	7	0	0	1	7
LB	1	2	0	430	492	369	112	1	8	0	5	0	0	0	5
MySQL	2	3	0	983	1044	396	154	22	42	1	4	0	0	1	4

As with Test 1, the CPU and Memory utilization was above 90%, as we kept the number of concurrent users at two. The average response time was reported at 10.907 seconds.

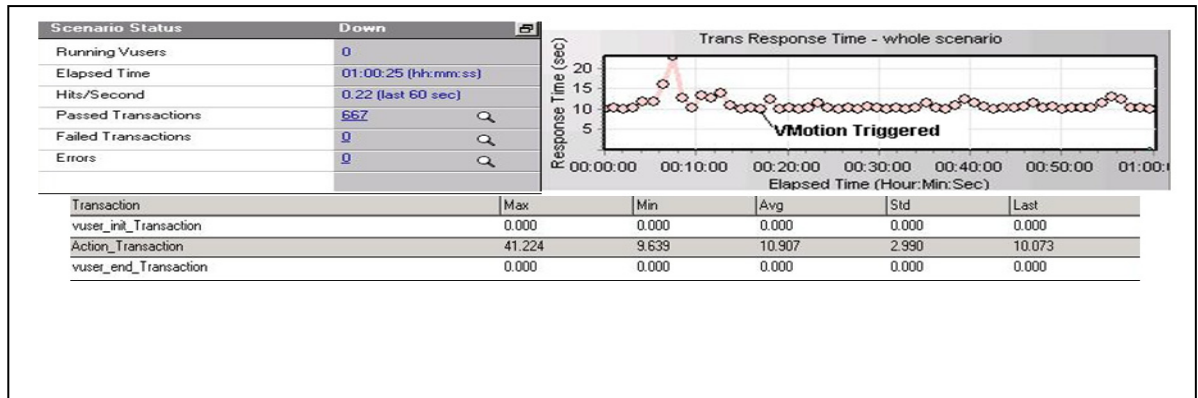


Figure 6. Graph of Test 2 Results

Figure 6 shows that the vMotion process took 25 to 40 seconds to complete. No transactions were failed during this test, since vMotion migrates virtual machines without service disruption, so there would be no impact on response times or transaction rates.

Test 3: DRS (One vCPU)

This test validated the VMware DRS feature operating with two 1-vCPU LiveCycle virtual machines. In this test, all (two) virtual machines were initially placed on a single host to saturate the host. After VMware DRS was triggered, one of the LiveCycle virtual machines was migrated to the other host.

Table 5. Test 3 Results

VM	CPU			Memory		Network		Disk				Kernel		Guest	
	(%)			(mb)		Packets/sec		I/O (sec)		Latency (ms)		Latency (ms)		Latency (ms)	
	Avg	Max	Rdy	Avg	Max	Recd	Txd	Avg	Max	Avg	Max	Avg	Max	Avg	Max
LiveCycle 1	94	100	0	3932	4096	228	105	7	18	0	7	0	0	0	17
LiveCycle 2	83	102	0	3979	4055	224	95	5	138	1	7	0	0	1	7
LB	1	3	0	389	440	361	108	1	5	0	17	0	0	0	17
MySQL	2	3	0	925	1044	380	148	22	48	0	7	0	0	0	7

Since VMware DRS does not introduce any significant overhead, the average response time stayed approximately the same at 10.94 seconds. The vMotion migration process was completed in 25 to 40 seconds.

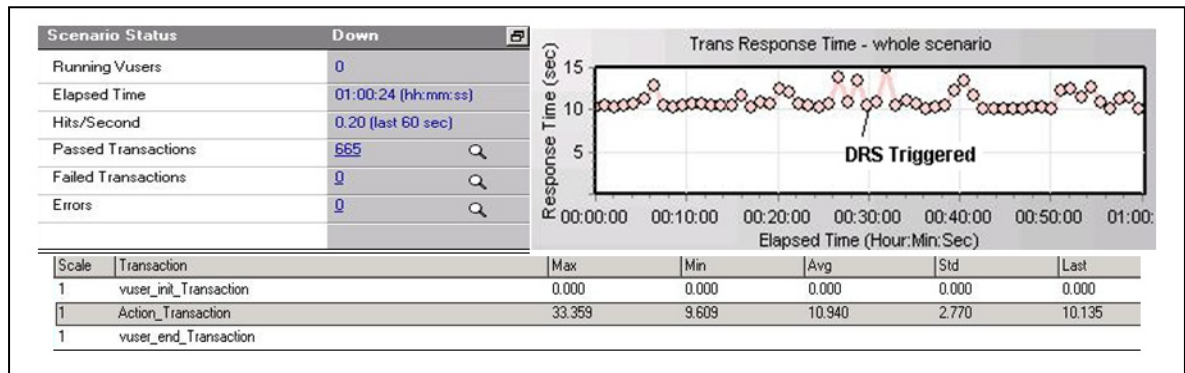


Figure 7. Graph of Test 3 Results

Figure 7 above indicates that there were a couple of short spikes in response time, but the system stabilized very quickly, and there were failed transactions.

Test 4: High Availability Test 1 (One vCPU)

Two 1-vCPU LiveCycle virtual machines (only one running) were purposely configured on two separate ESX hosts as part of the setup for this test. The High Availability Test was then performed by crashing the host where the LiveCycle virtual machine instance was running. The second virtual machine LiveCycle application was started post HA.

Table 6. Test 4 Results

VM	CPU			Memory		Network		Disk				Kernel		Guest	
	(%)			(mb)		Packets/sec		I/O (sec)		Latency (ms)		Latency (ms)		Latency (ms)	
	Avg	Max	Rdy	Avg	Max	Recd	Txd	Avg	Max	Avg	Max	Avg	Max	Avg	Max
LiveCycle 1	50	53	0	4005	4055	427	169	7	28	2	31	0	0	2	31
LiveCycle 2	99	100	0	3990	4044	245	110	10	32	1	18	0	0	1	18
LB	1	1	0	346	399	383	104	1	8	2	27	0	0	2	27
MySQL	2	3	0	1050	1270	357	119	19	41	2	28	0	0	2	28

In our test, VMware HA was triggered by crashing the ESX host that had the LiveCycle virtual machine instance running. (Until VMware HA was triggered, the load balancer was forwarding all requests to the active LiveCycle virtual machine.) The crashed virtual machine was then started on the other ESX host. In this scenario, we did not start the second host LiveCycle virtual machine application until post VMware HA transfer between the hosts.

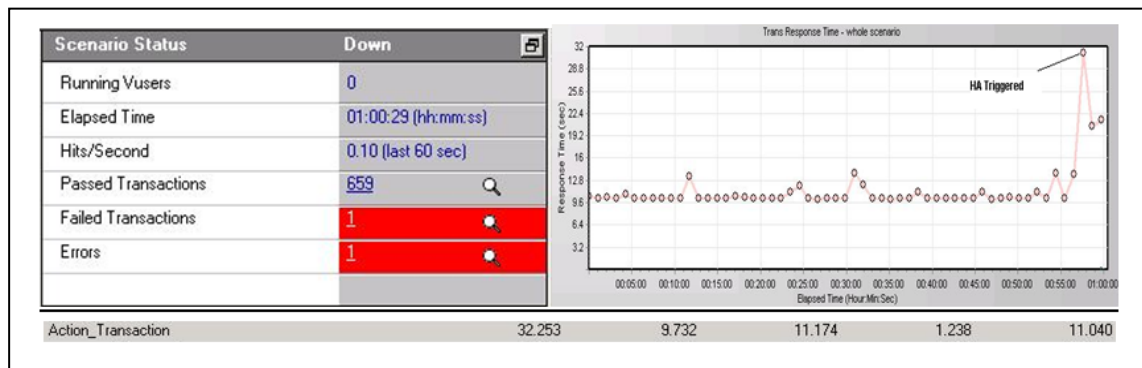


Figure 8. Graph of Test 4 Results

Figure 8 shows only one active transaction failed.

Test 5: High Availability Test 2 (One vCPU)

Two 1-vCPU LiveCycle virtual machines were purposely configured on separate hosts as part of the test setup. The High Availability test was then performed by crashing one of the hosts. The LiveCycle application was started immediately once the virtual machine came up on the second host to restore full capacity provided by the 2-node LiveCycle cluster.

Table 7. Test 5 Results

VM	CPU			Memory		Network		Disk				Kernel		Guest	
	(%)			(mb)		Packets/sec		I/O (sec)		Latency (ms)		Latency (ms)		Latency (ms)	
	Avg	Max	Rdy	Avg	Max	Recd	Txd	Avg	Max	Avg	Max	Avg	Max	Avg	Max
LiveCycle 1	52	55	0	4001	4049	430	175	7	25	2	29	0	0	2	29
LiveCycle 2	99	100	0	3990	4001	248	111	10	29	1	19	0	0	1	19
LB	1	2	0	346	425	390	110	1	8	2	27	0	0	2	27
MySQL	2	5	0	1049	1210	365	125	19	40	2	25	0	0	2	25

When HA was triggered, the active transaction failed while the virtual machine was rebooting. After the virtual machine was up on the second host, the LiveCycle application was manually started. During the test, we observed that the load balancer had started sending requests to the second host even before the LiveCycle virtual machine was completely started. Figure 9 below provides a graph of the results of Test 5.

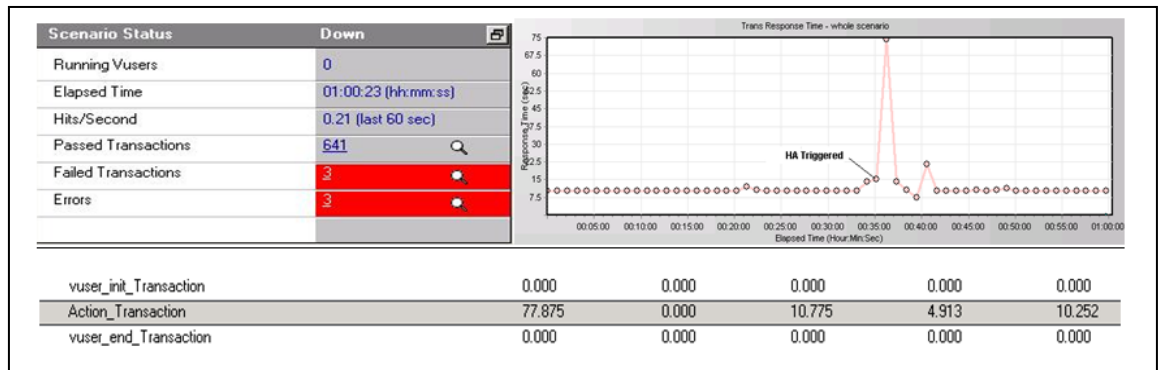


Figure 9. Graph of Test 5 Results

Figure 9 shows that more transactions failed here than during the previous test, because the load balancer started sending requests before the application on the restarted second LiveCycle instance was completely up and running.

Test 6: Fault Tolerance (One vCPU)

This test validated VMware Fault Tolerance (FT) by crashing one of the two 1-vCPU LiveCycle application virtual machines.

Table 8. Test 6 Results

VM	CPU			Memory		Network		Disk				Kernel		Guest	
	(%)			(mb)		Packets/sec		I/O (sec)		Latency (ms)		Latency (ms)		Latency (ms)	
	Avg	Max	Rdy	Avg	Max	Recd	Txd	Avg	Max	Avg	Max	Avg	Max	Avg	Max
LiveCycle 1	91	100	0	3976	4055	232	94	5	18	1	22	0	0	5	18
LiveCycle 2	100	101	0	3991	4055	250	109	8	32	1	19	0	0	1	19
LB	2	6	0	941	1065	415	161	23	48	2	13	0	0	2	13
MySQL	1	3	0	989	1055	406	159	23	307	1	7	0	0	1	7

There were no transactions lost during this test. The second VMware FT virtual machine started instantly, once the primary first virtual machine was down.

Figure 10 below provides a graph of the results of Test 6.

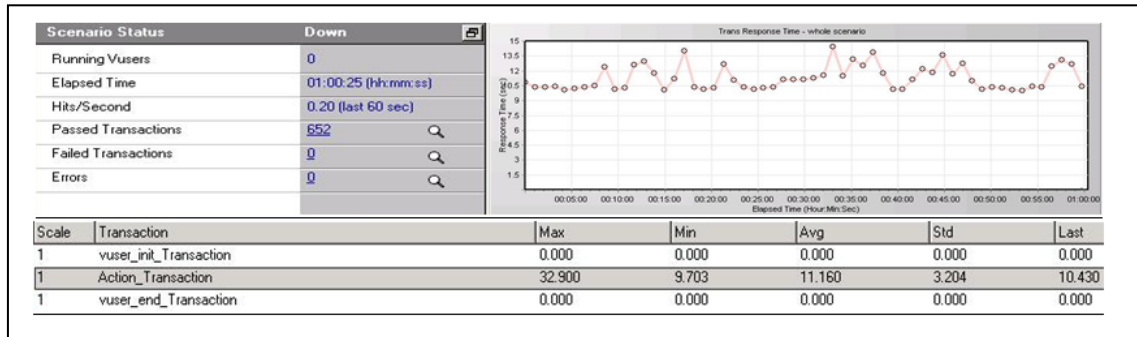


Figure 10. Graph of Test 6 Results

The graph in Figure 10 shows that no transactions failed during this test and the average number of transactions was in the mid-600 range.

Test 7: Baseline Test (Two vCPUs)

This second baseline test (in contrast to Test 1) was performed with two LiveCycle application virtual machines, each configured with two vCPUs. The goal was to measure the number of transactions processed.

Table 9. Test 7 Results

VM	CPU			Memory		Network		Disk				Kernel		Guest	
	(%)			(mb)		Packets/sec		I/O (sec)		Latency (ms)		Latency (ms)		Latency (ms)	
	Avg	Max	Rdy	Avg	Max	Recd	Txd	Avg	Max	Avg	Max	Avg	Max	Avg	Max
LiveCycle 1	49	56	0	3980	4055	225	91	5	23	1	25	0	0	1	25
LiveCycle 2	53	61	0	3972	4055	261	119	7	21	1	9	0	0	1	9
LB	1	2	0	417	481	396	114	1	8	1	47	0	0	1	47
MySQL	48	59	0	1041	1147	405	154	23	44	2	13	0	0	2	13

A total of 709 transactions were processed. There were no failed transactions. It is worth noting that CPU and memory utilization was around 50%, after doubling the number of vCPUs of the LiveCycle application virtual machines. In other words, the LiveCycle cluster could now handle twice the amount of the current workload.

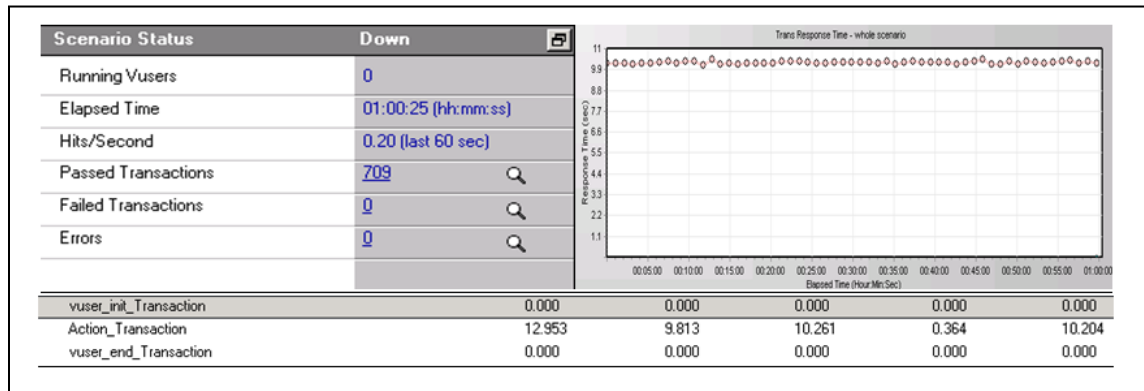


Figure 11. Graph of Test 7 Results

The graph in Figure 11 shows that there was a slight increase in the number of transactions processed when comparing to our 1-vCPU test.

Test 8: VMotion (Two vCPUs)

For this test, we validated and checked vMotion performance with two LiveCycle application virtual machines on each ESX host, and each virtual machine configured with two vCPUs (in contrast with Test 2).

Table 10. Test 8 Results

VM	CPU			Memory		Network		Disk				Kernel		Guest	
	(%)			(mb)		Packets/sec		I/O (sec)		Latency (ms)		Latency (ms)		Latency (ms)	
	Avg	Max	Rdy	Avg	Max	Recd	Txd	Avg	Max	Avg	Max	Avg	Max	Avg	Max
LiveCycle 1	45	55	0	3900	4055	195	165	6	25	0	6	0	0	0	6
LiveCycle 2	55	61	0	4005	4055	250	125	7	23	1	11	0	0	1	11
LB	1	2	0	434	492	400	113	1	9	0	18	0	0	0	18
MySQL	2	4	0	1046	1167	425	168	24	49	2	9	0	0	2	9

Completion of the vMotion process during this test took between 25 to 40 seconds and there were no transaction failures reported.

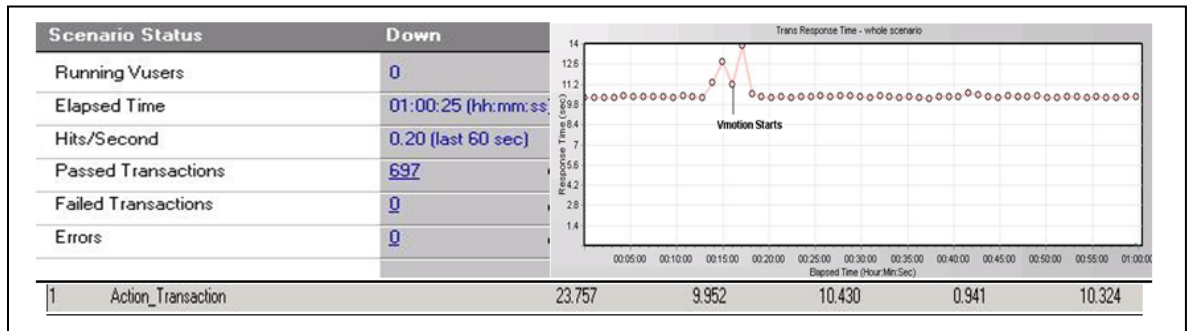


Figure 12. Graph of Test 8 Results

The graph in Figure 12 shows that there were a couple of spikes in response time during the vMotion transaction, but the system stabilized very quickly.

Test 9: VMware DRS (Two vCPUs)

For this test, we validated and checked performance of VMware DRS with two LiveCycle application virtual machines, each configured with two vCPUs. All the virtual machines were placed on a single host to saturate the ESX host.

Table 11. Test 9 Results

VM	CPU			Memory		Network		Disk				Kernel		Guest	
	%			mb		Packets/sec		I/O (sec)		Latency (ms)		Latency (ms)		Latency (ms)	
	Avg	Max	Rdy	Avg	Max	Recd	Txd	Avg	Max	Avg	Max	Avg	Max	Avg	Max
LiveCycle 1	49	58	0	3910	4001	190	164	5	23	0	5	0	0	0	5
LiveCycle 2	55	61	0	4005	4055	251	120	7	21	1	10	0	0	1	10
LB	1	2	0	434	492	400	113	1	7	0	20	0	0	0	20
MySQL	2	4	0	1046	1167	425	168	24	47	2	8	0	0	2	8

To perform the test, VMware DRS was triggered and one of the LiveCycle virtual machines was migrated to the other host.

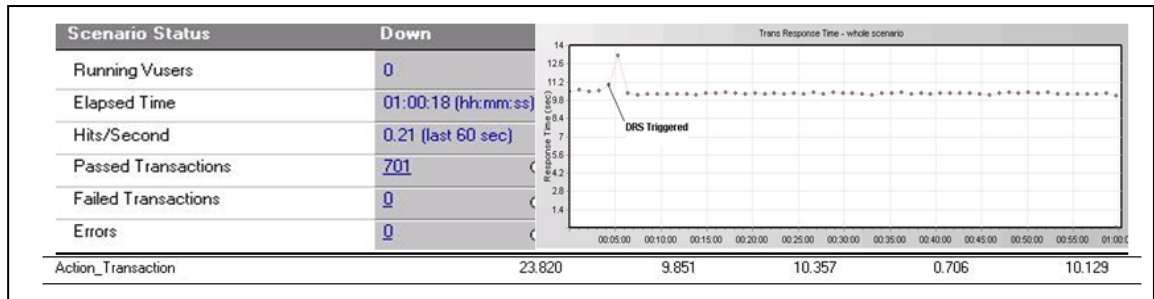


Figure 13. Graph of Test 9 Results

The graph in Figure 13 shows that there were no transactions failures reported. There was a single spike in response time due to a vMotion transaction triggered by VMware DRS.

Test 10: High Availability Test 1 (Two vCPUs)

For this test, we configured two LiveCycle application virtual machines, each with two vCPUs. To validate and measure VMware HA performance with this configuration, after the HA process completed, the LiveCycle application on the second virtual machine was not started. This step was taken to check that the load balancer was sending all the requests to the active LiveCycle application virtual machine.

Table 12. Test 10 Results

VM	CPU			Memory		Network		Disk				Kernel		Guest	
	(%)			(mb)		Packets/sec		I/O (sec)		Latency (ms)		Latency (ms)		Latency (ms)	
	Avg	Max	Rdy	Avg	Max	Recd	Txd	Avg	Max	Avg	Max	Avg	Max	Avg	Max
LiveCycle 1	49	58	0	4012	4055	190	375	7	25	1	7	0	0	1	7
LiveCycle 2	54	59	0	3985	3809	434	160	9	23	1	28	0	0	1	28
LB	1	2	0	439	502	391	115	3	9	2	30	0	0	2	30
MySQL	2	4	0	1009	1208	363	120	25	45	5	25	0	0	5	25

After VMware HA was triggered, the active transaction failed while the virtual machine was rebooting. It was found that the load balancer only sends all the requests to the active virtual machine (LiveCycle 1), as expected.

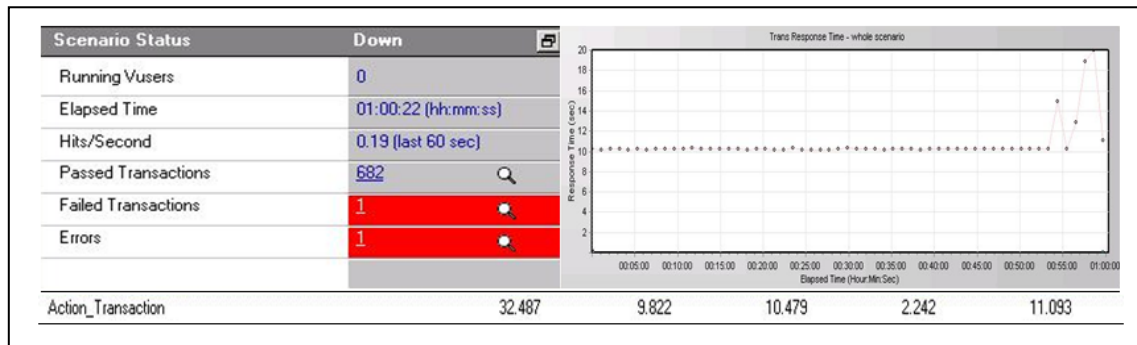


Figure 14. Graph of Test 10 Results

Like our 1-vCPU VMware HA test (Test 4), the graph in Figure 14 shows only one active transaction failed.

Test 11: High Availability Test 2 (Two vCPUs)

For this High Availability test, two LiveCycle application virtual machines were set up on each ESX host; each virtual machine was configured with two vCPUs. During the HA process, once the virtual machine was up and running on the second host, the LiveCycle application virtual machine on the second host was started immediately. This step was taken to check that the load balancer resumes sending the requests to both LiveCycle virtual machines to restore full capacity use.

Table 13. Test 11 Results

VM	CPU			Memory		Network		Disk				Kernel		Guest	
	(%)			(mb)		Packets/sec		I/O (sec)		Latency (ms)		Latency (ms)		Latency (ms)	
	Avg	Max	Rdy	Avg	Max	Recd	Txd	Avg	Max	Avg	Max	Avg	Max	Avg	Max
Live Cycle 1	51	58	0	4014	4058	192	376	7	26	1	8	0	0	1	8
LiveCycle 2	53	61	0	3989	3812	434	163	9	22	1	29	0	0	1	29
LB	1	3	0	440	504	392	118	3	11	2	32	0	0	2	32
MySQL	2	5	0	1010	1210	365	122	25	44	5	29	0	0	5	29

After VMware HA was triggered, the active transaction failed. When the virtual machine on the second host was up and running, the LiveCycle application was manually started. It was observed that, even before the LiveCycle application was completely started, the load balancer had started sending requests to it. It was also observed that the initial transactions failed, since the LiveCycle application on the second host was not completely up. Later, however, when the application was up again, there were no further transaction failures recorded.

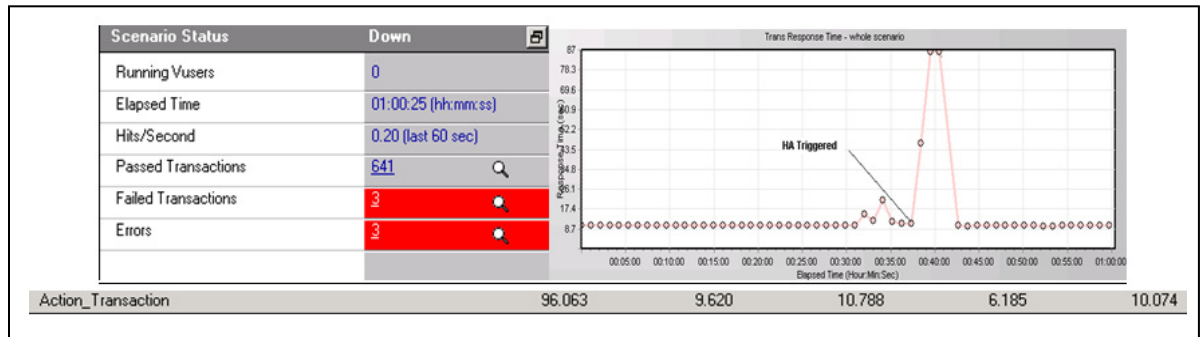


Figure 15. Graph of Test 10 Results

The graph of Figure 15 shows that more transactions failed than during the previous test, because the load balancer started sending requests to the application on the restarted LiveCycle virtual machine before it was completely up. That result is consistent with Test 5 performed with 1-vCPU virtual machines.

Reference Architecture

This section provides an overview how one might typically deploy Adobe LiveCycle ES2 in different-sized business environments.

Workgroup and Smaller-Size LiveCycle Deployments

Figure 16 shows a configuration for smaller-size LiveCycle ES2 deployments on VMware vSphere.

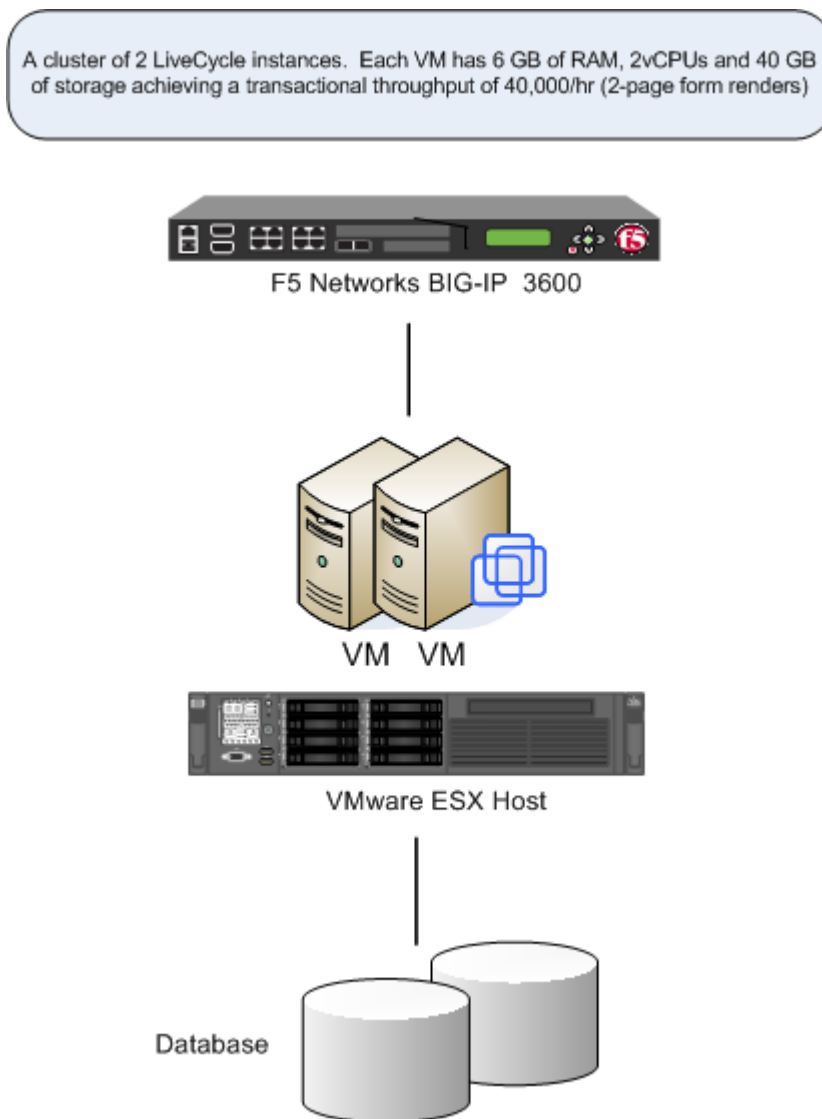


Figure 16. Workgroup and Smaller-Size Deployment Solution

This deployment topology is a good fit for small businesses as well as small workgroups within larger organizations with up to 200 users, which translates to a load of about 10 concurrent requests. Having at least two virtual machines ensures higher availability for the application. One virtual machine should be able to handle all of the load in case one of the virtual machines fails.

Departmental and Medium-Size LiveCycle Deployments

Figure 17 shows a deployment configuration of LiveCycle ES2 on vSphere for departmental or medium-size business applications serving up to 400 users with a 20 concurrent request load.

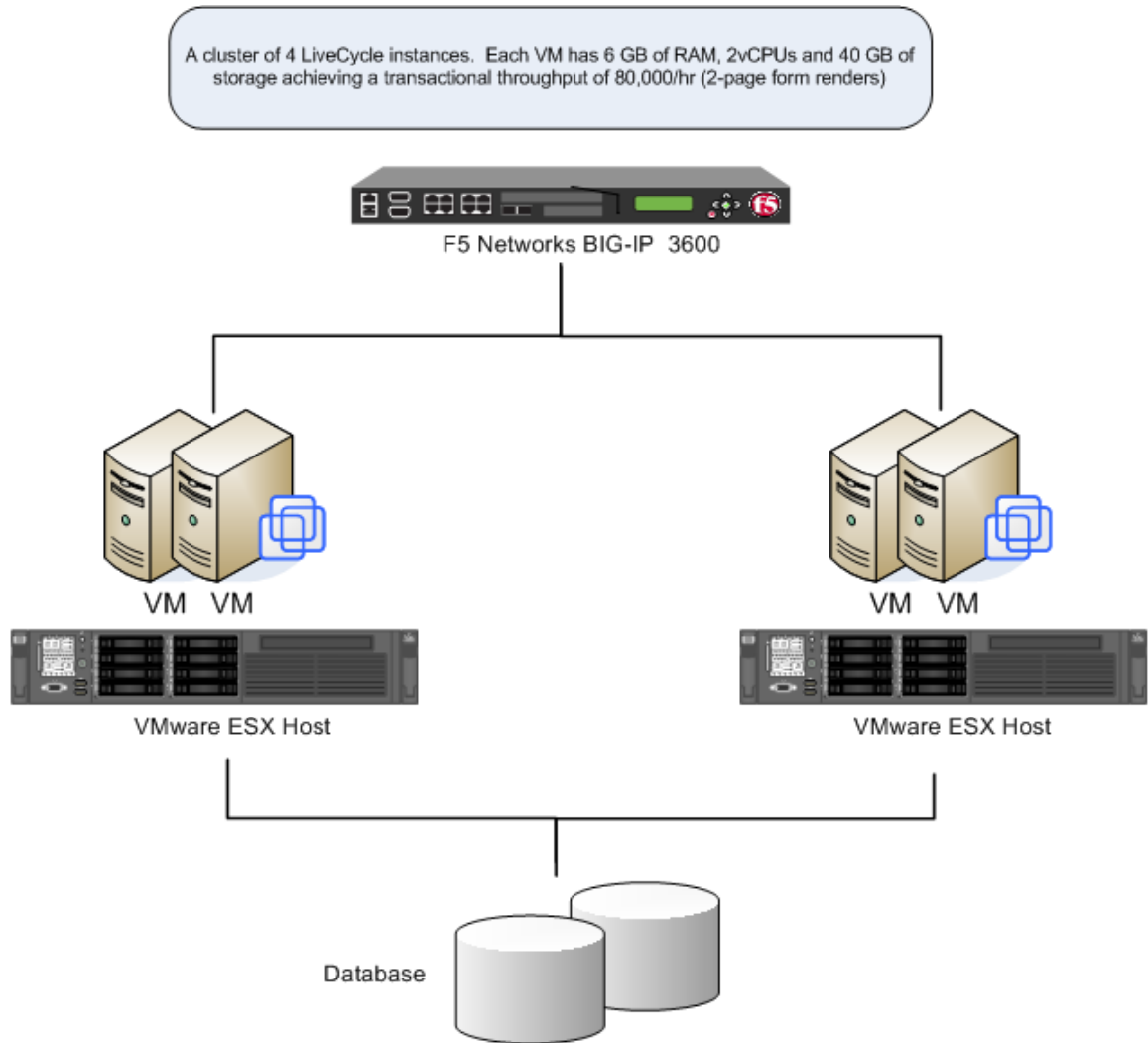


Figure 17. Departmental and Medium-Size Business Deployment Solution

Enterprise and Larger-Size LiveCycle Deployments

Figure 18 shows the configuration of a LiveCycle ES2 deployment on vSphere for Enterprise and larger-size business applications.

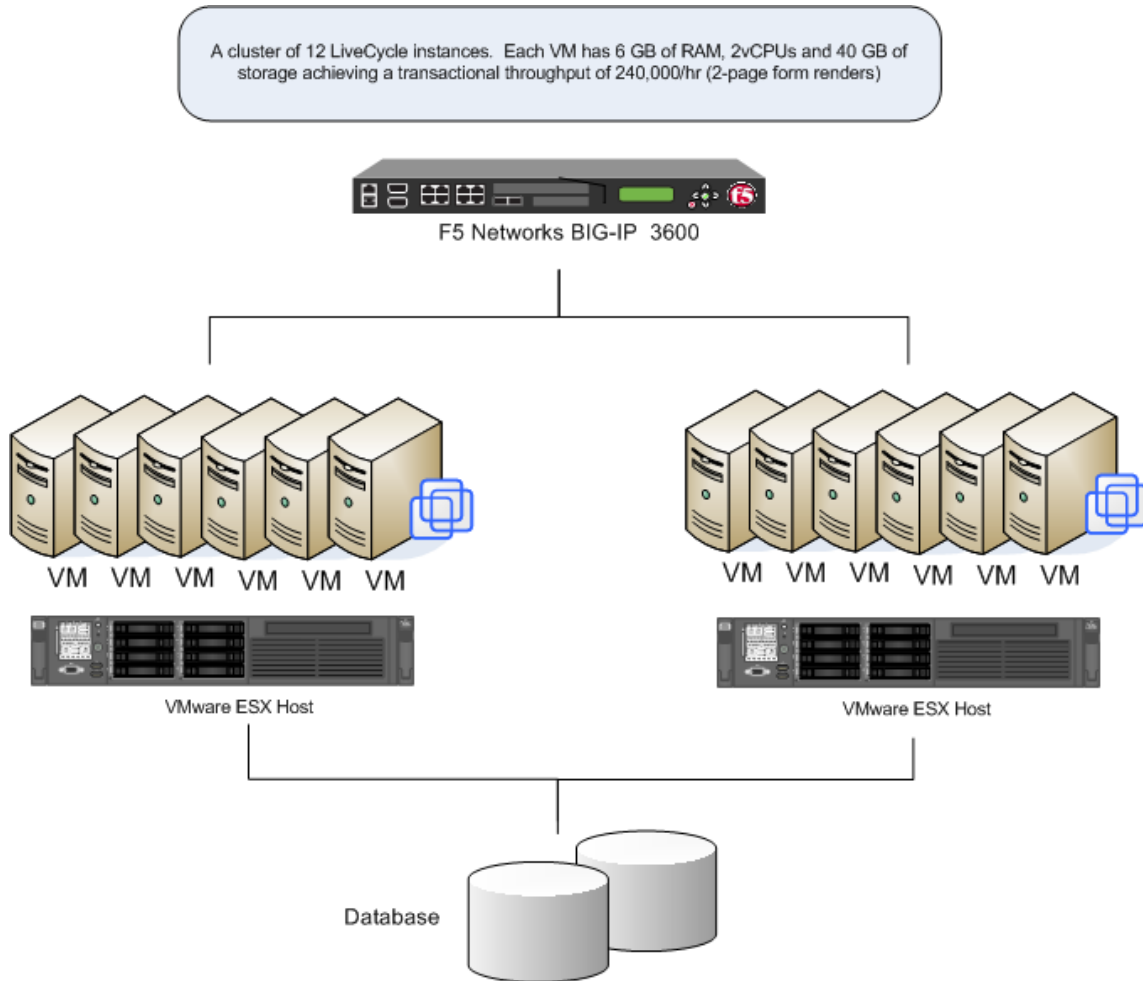


Figure 18. Enterprise and Larger-Size Business Solution

Because enterprise-sized LiveCycle deployments may experience a wide variety of workloads, it is recommended that the specific solution topology be sized during an iterative cycle of application performance testing. Long-lived orchestrations tend to be database and I/O-intensive, while short-lived orchestrations tend to be application server and CPU-intensive.

Deployment Best Practices

VMware best practices are outlined in the VMware documentation available at:

<http://www.vmware.com/support/pubs/>

VMware also offers specific documents on VMware vSphere 4 performance enhancements, networking and storage performance, best practices for VMware VMFS, and resource management with VMware Distributed Resource Scheduler (DRS). Additional specific recommendations for VMware infrastructure deployments are:

- Make sure that the host server and storage that you will be using to deploy VMware ESX are listed on the *VMware Systems and Storage Hardware Compatibility List (HCL)* available at:
<http://www.vmware.com/resources/compatibility/>
- For best performance, it is recommended that you run Adobe LiveCycle ES2 Server on VMware ESX 3.5, update 3, or later versions.
- Disconnect unused, unnecessary devices on both the guest and on the host. These include COM ports, LPT ports, Floppy drives, CD-ROM drives, and USB adapters. Disabling devices on the host frees IRQ resources and eliminates IRQ sharing conflicts that can cause performance problems.
- Make sure that you are running the latest version of VMware Tools in the guest operating system of virtual machines.
- Do not set resource reservations and limits unless required. Set the limit as “unlimited,” which is the default specified by VMware.
- To establish a network between two virtual machines that reside on the same ESX host, connect both virtual machines to the same virtual switch. If the virtual machines are connected to different virtual switches, traffic will go through “wire” and incur unnecessary CPU and network overhead.
- Using a SAN device for storage is recommended to best meet I/O requirements for applications and to leverage all VMware features and capabilities. Using iSCSI or NFS, respectively, provide the next best storage performance.
- If local disks must be used, VMware recommends placing the VMware ESX operating system on disks separate from the VMFS file system where the virtual machines reside.

Licensing

LiveCycle server products are licensed in a variety of options, all of which support VMware virtualization. License per CPU/Core or per Server virtualization options have specific technical constraints. With per CPU/Core virtualization licensing, a virtual processor is considered equivalent to a licensable core, so limiting the number of virtual processors to the number of licensed cores ensures compliance. With per Server licensing, each VMware instance is considered a licensable server. The other licensing options do not have specific requirements on how virtualization is configured.

Technical Support

For Technical Support, go to the main Support portal web site provided for the Adobe LiveCycle application located at: <http://www.adobe.com/support/programs/livecycle/>

Conclusions

Overall testing results show that LiveCycle ES2 running on VMware virtual infrastructure performs well. Furthermore, using virtual infrastructure for deployments has the potential to reduce cost, increase service levels and simplify the manageability of LiveCycle applications.

Customers deploying LiveCycle in a clustered configuration on VMware can scale out to handle increased workloads and also scale back when workloads are decreased. Combining VMware virtual infrastructure with other VMware vCenter management products can reduce the time and effort customers spend actively managing deployed applications.

Adobe and VMware continue their joint efforts and work together to validate how IT data centers can best meet enterprise SLA requirements by deploying Adobe LiveCycle ES2 in a vSphere virtualized environment. The performance results included in this paper prove that running LiveCycle ES2 inside VMware virtual machines can provide a highly efficient production-ready platform.

The reference architecture section provides information on how to size and design small, medium and larger-size Adobe LiveCycle ES2 deployments. Customers with more dynamic usage load patterns can deploy LiveCycle ES2 in a clustered configuration of VMware virtual machines to quickly scale out to meet increased workloads and then scale back when workloads are decreased. With VMware vCenter management capabilities such as high availability, distributed resource scheduling, and fault-tolerance, customers can rest assured that their LiveCycle ES2 application will run uninterrupted.

In summary, customers can have confidence that they will achieve significant cost savings and gain increased agility and productivity when deploying Adobe LiveCycle ES2 in a VMware virtual infrastructure environment.

Resources

Customers can find more information about VMware and LiveCycle ES2 products via the links listed below.

VMware Resources

- VMware official website:
<http://www.vmware.com/>
- VMware Infrastructure 3 and vSphere 4 product Web site:
http://www.vmware.com/products/data_center.html
- VMware download Web site:
<https://www.vmware.com/download/>
- VMware support Web site:
<http://www.vmware.com/vmtn/>
- VMware Performance Tuning Paper:
http://www.vmware.com/pdf/vi_performance_tuning.pdf
- System Compatibility Guide for a complete list of compatible hardware:
http://www.vmware.com/pdf/vi35_systems_guide.pdf
- Storage/SAN Compatibility Guide for a complete list of compatible storage devices:
http://www.vmware.com/pdf/vi35_san_guide.pdf
- I/O Compatibility Guide for a complete list of compatible networking devices:
http://www.vmware.com/pdf/vi35_io_guide.pdf

Adobe Resources

- Adobe LiveCycle ES2 Web site:
<http://www.adobe.com/products/livecycle/>
- Adobe Developer Center LiveCycle Portal:
<http://www.adobe.com/devnet/livecycle>
- Adobe LiveCycle ES2 Online Documentation Portal:
<http://www.adobe.com/devnet/livecycle/?view=documentation>
- Adobe LiveCycle ES2 API QuickStarts (sample code):
http://help.adobe.com/en_US/livecycle/9.0/programLC/help/index.htm?content=000004.html
- Adobe LiveCycle ES Platform Support Matrix
http://www.adobe.com/support/products/enterprise/knowledgecenter/softcomp2_0_1.pdf

