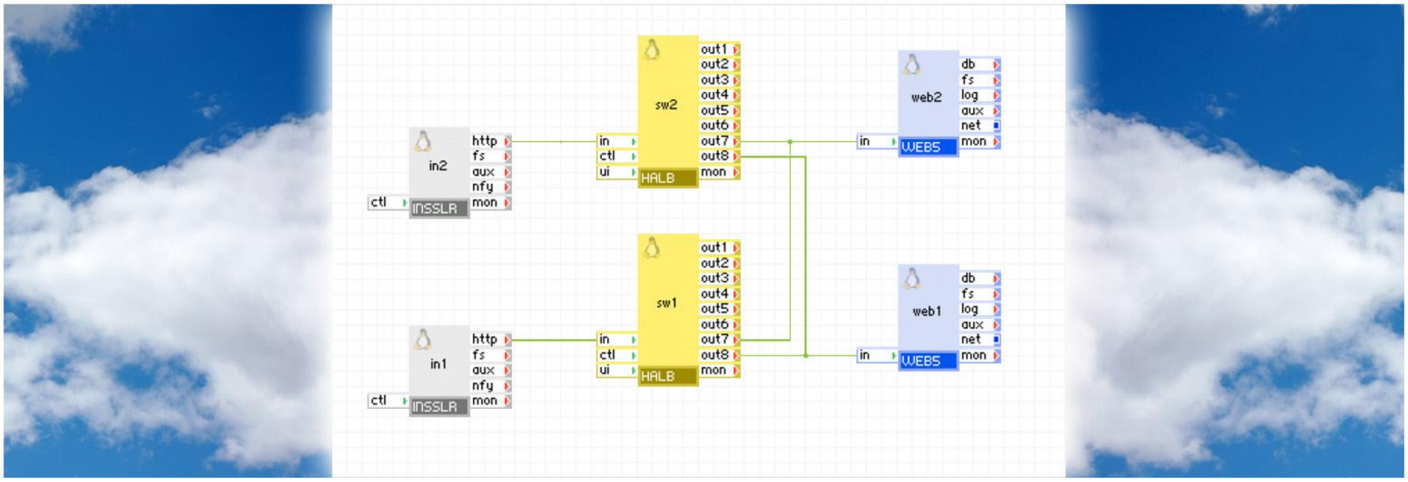
CA AppLogic® turnkey cloud computing platform greatly simplified network management



OVERVIEW

Cloud computing, one of the most innovative recent information technology developments, benefits enterprises by simplifying the administration and optimal use of hardware resources. Clouds also help better align IT and business with higher agility and responsiveness to changing business needs and environments.

CA AppLogic is an effective turnkey cloud solution for deploying, managing, and maintaining network-based distributed applications and services.

In Principled Technologies tests in our labs, we found that CA AppLogic greatly simplified network management while significantly reducing the time it takes to deploy an application or service, thanks to the unique networking capabilities embedded in its modeling and run-time environments and in its cloud fabric.

The CA AppLogic cloud technology abstracts the physical network connecting the cloud hardware resources from the operation of the applications that use the cloud, which is central to achieving the full potential of cloud computing. This paper focuses on the networking aspects of CA AppLogic.

JANUARY 2011
A PRINCIPLED TECHNOLOGIES WHITE PAPER
Commissioned by CA Technologies

We tested the CA 3AppLogic networking features for version 2.9.9 on three dual-processor Dell™ PowerEdge™ 2950 servers. We configured these servers with directly attached storage drives, and connected them with dual front-end and backend switches and an Internet accessible input/output security layer.

We examined several aspects of the CA AppLogic cloud network infrastructure concerning application network topology and bandwidth requirement definition, scalability, resiliency, and security. Specifically, we evaluated the following: (1) the ease of configuring the physical network infrastructure to use in the cloud, (2) the ease of defining application network topologies and components, (3) the ease of scaling up or down the physical network and switch infrastructure assigned to an application in response to changing application requirements, and (4) the resiliency and robustness of cloud application deployments when faced with network failures or reconfigurations.

INTRODUCTION

Clouds and modern applications

Clouds are all about applications. The agility and responsiveness a cloud can bring for provisioning, deploying, and scaling applications is what makes it such an attractive technological solution to align business needs with IT infrastructure.

Modern application architectures differ significantly from the monolithic architecture of early computing applications. The advent of the Internet and its associated Web computing models has introduced complex, multi-layered, and distributed application architectures, where each layer consists of many components deployed and executed on network fabrics and interconnects linking potentially dissimilar servers and data storage facilities. The sheer intricacy of configuring, testing, and deploying such component layers can challenge even highly skilled users. The need to deploy applications manually further compounds these issues, increasing both operational costs and time-to-market for new applications.

Network infrastructure and cloud applications

Modern applications consist of interconnections of components that communicate through standard transports and protocols. Links between components are implemented from a wide selection of physical substrates, ranging from memory-to-memory interconnects through low-speed Ethernet links to high-performance GigE, 10GigE, or Infiniband® fabrics, or even actual Internet links for external connections.

The provisioning and deployment of network infrastructure becomes critical as client bases increase and application requirements change from client to client. Using traditional technologies to provision and deploy many instances of the same application with varying processing, storage, and network bandwidth requirements on potentially dissimilar hardware is a highly error-prone and tedious task. In addition, the fabric and application interconnects must ensure that adequate bandwidth is always available to satisfy their performance requirements. While multi-tenant applications can help address some of these provisioning problems, they introduce other issues, such as the need for appropriate measures to ensure correct data partitioning, data security, and non-interference among application and data instances being executed by different customers.

CLOUD NETWORKING IN CA APPLOGIC

Networking in CA AppLogic

CA AppLogic uses modeling as its primary cloud interface. To this effect, it provides a diagrammatic, Microsoft® Visio®-like interface to model applications as hierarchical compositions of virtualized appliances (i.e., application software components) and their network interconnections. Applications are visually represented as diagrams, where diagram boxes denote virtualized appliances and links between boxes represent network links between appliances (see, for example, the application diagram at the beginning of this paper). In CA AppLogic, a cloud is configured from server, storage, and networking components in a manner opaque to the end user. A cloud controller module takes care of provisioning and managing all networking and hardware infrastructure on behalf of cloud applications. The end user need not know the specifics of the cloud hardware and networking infrastructure to define, deploy, and redeploy cloud applications.

From a networking perspective, each appliance's visual representation includes a set of distinctive input and output pads that implement different possible networking uses for incoming or outgoing connections from the appliance. Functions designate inputs and outputs. Some functions implemented in CA AppLogic include *http* for http links, *db* for database links, *fs* for file system links, and *nfs* and *cifs* for NFS and CIFS file system protocols, among others.

Connections in these application diagrams are logical. The CA AppLogic cloud controller implements these connections on the cloud's physical network infrastructure. By ensuring that all component connections are logical, CA AppLogic has control over all application connections, and can ensure that each connection is locally firewalled, a feat that is nearly impossible on traditional deployments. This has important implications for network security of CA AppLogic clouds (see CLOUD SECURITY IN CA APPLOGIC[1] white paper).

Cloud networking advantages of CA AppLogic

CA AppLogic not only makes it easy to model cloud applications and their networking infrastructure, simplifying reuse and sharing, but also greatly simplifies network management and application deployment and execution directly from these visual models. CA AppLogic can also automatically discover the physical network infrastructure's topology and cabling layout and maintain network redundancy and recovery capabilities in case of network component failures. CA AppLogic displays the state of the network infrastructure—the network paths in use—at all times. It automatically detects and compensates for system changes (such as wiring changes), detects network failures, issues alarms to the cloud console, and performs automated recovery from such failures. CA AppLogic manages all network path optimizations, ensuring maximum bandwidth availability for application network-infrastructure components.

CA AppLogic also ensures high availability of networks and other application subsystems, and recovers from single-element failures without human intervention. CA AppLogic Release 2.9.9 provides redundant network switch support, including fault detection, failed component isolation, and recovery of failed applications. Upon restoring an application to full operation, CA AppLogic rebuilds network redundancy. CA AppLogic is a unique solution in this respect, because redundancy is handled centrally by a single cloud product, simplifying user work and increasing overall system reliability and failure recovery capabilities.

WHAT WE FOUND

To test the cloud networking features of CA AppLogic, we set up a test-bed cloud and defined on it a simple network-intensive online application. In the process, we verified how easy it was to get a cloud and application running. Then, to determine the network stability of the CA AppLogic program, we evaluated both the physical and virtual functionalities of the network and bandwidth provisioning for this application.

Initial cloud setup

We began our testing of CA AppLogic by setting up an operational cloud. The most challenging part of the process was to ensure that our physical hardware setup satisfied all constraints imposed by the CA AppLogic cloud controller, which involved our studying in depth the available documentation and performing several experiments. Once we achieved this, deploying the cloud was simple and involved only a minimal

[1] http://principledtechnologies.com/clients/reports/CA/AppLogic_security.pdf

amount of work. We expect that for most users, a successful first CA AppLogic cloud deployment will make subsequent deployments much simpler as users will better understand the physical hardware requirements of their clouds.

We decided to build our cloud using commodity Linux servers. We selected one of these as our deployment server, called the *Aldo Distribution Server* in the CA AppLogic documentation. We downloaded the CA AppLogic software onto it and set up the software. Next, we defined a cloud configuration file, which included a number of required attributes at the cloud boundary, including the name of the cloud, the IP address range for external IPs, a list of IP addresses of all servers to use for deployment of the cloud, an SSH certificate path, and a user name associated with the certificate. Defining the configuration file was straightforward. The SSH certificate needed to be set up from the outset as it serves the role of a license key—it is authenticated by the CA AppLogic servers, verifies that an AppLogic cloud deployment is valid, and is required to download updates from the CA AppLogic Web site.

Before you can install CA AppLogic on a cloud server, it must be running a Linux distribution. CA has tested that AppLogic works with Red Hat®, CentOS, and Fedora™ distributions. We had CA AppLogic install 32- and 64-bit versions of CentOS initially, and later we had it add a Red Hat Enterprise Linux® 5.5 server to our cloud.

Once we had defined the configuration file and our deployment and cloud servers were in place, we ran a CA AppLogic command to create the cloud from the configuration file. This installed a bare-minimum Linux configuration on each of the cloud servers, and created two partitions in each cloud server with the CA AppLogic cloud software, one active and one backup. This feature is handy for rolling back to a previous version of CA AppLogic if a software upgrade does not work as expected.

At this point, CA AppLogic built a VSAN accessible from all directly attached storage in the cloud servers. During this process, CA AppLogic selected one of the cloud servers as the primary server for the cloud controller and one or more secondary servers as backups should the primary server fail. Once CA AppLogic had finished deploying the cloud, we verified that the cloud was operational and that we could log in and start defining our test application.

Application definition, deployment, and execution

Next, we defined an application to run while we tested the resiliency of CA AppLogic cloud networking. To examine the effects of network failures on cloud application performance, we elected to define a simple, minimal-fail-point Web application. We created a load-balanced WebBench application executing in redundant servers against a shared 1GB data volume.

We found application definition to be extremely simple, especially given our understanding of the types of appliances CA AppLogic provides. We selected redundant SSL switches connected to a dual load-balancer running against two instances of a Web server. When considering what kind of backend storage to use, we decided to forego using a NAS or similar appliance and went with the much simpler solution of defining a data volume on the AppLogic VSAN and sharing the volume amongst both Web server instances.

The application template is created using a Microsoft Visio-style diagrammatic user interface that includes a catalog of appliance icons. The user drags and drops these onto a drawing frame, and connects the appliances by dragging links from output pads to input pads. Because all appliances are running on their own virtual machines, by drawing the application graph, the user is effectively drawing the network topology. Figure 1 shows the cloud test environment we built. We chose high availability load balancing (HALB) components to experiment with the HA features introduced in the CA AppLogic 2.9.9 release we were testing.

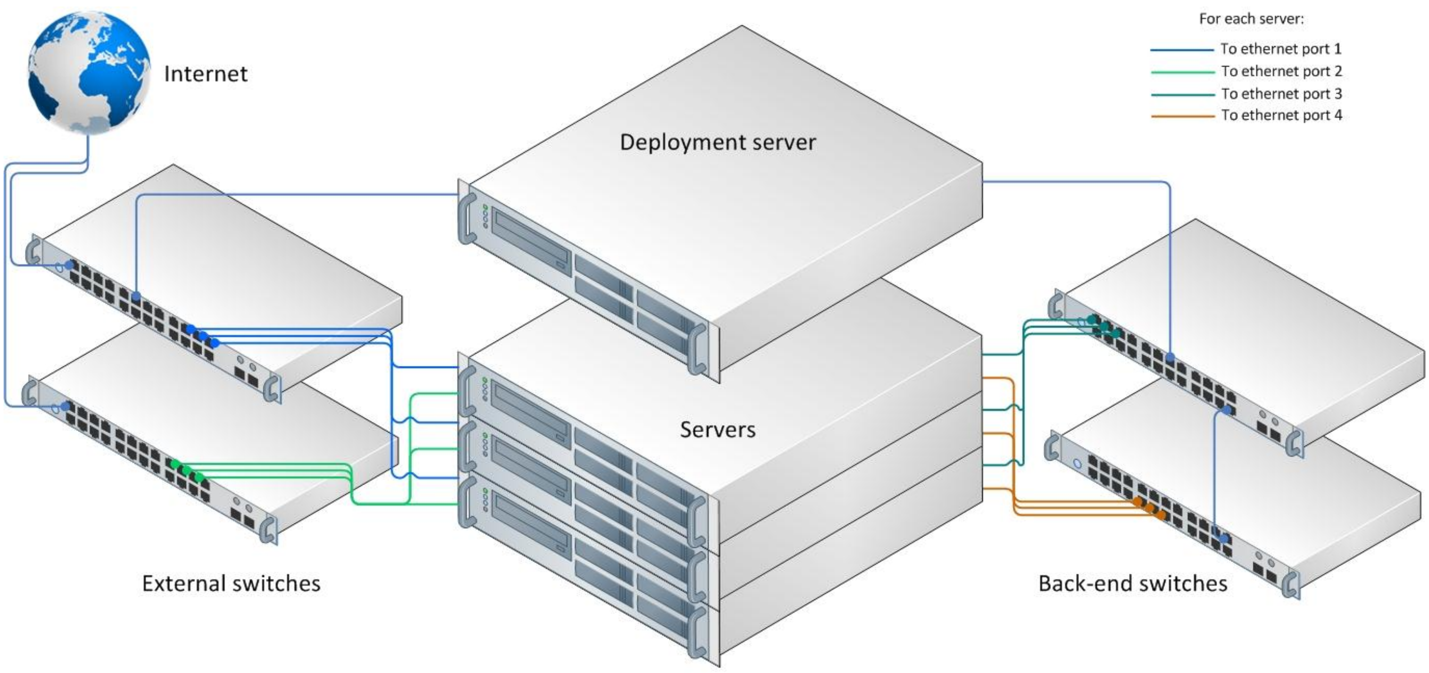


Figure 1: Our test environment.

Aside from the simplicity of the visual interface for defining an application, the CA AppLogic feature we found most useful for this stage was the self-documenting nature of each AppLogic component. By right-clicking a component and selecting Class Documentation, the user goes directly to detailed documentation of the component. This documentation includes a description of the component, its properties, its attributes, and—most valuably—many examples of actual application deployment situations the user can alter or use as desired. CA AppLogic self-documentation is invaluable for application definition.

High availability functionality

High availability, the ability to recover from all types of failures with little to no downtime, is crucial for modern networks. Cloud computing addresses this by implementing full network and system redundancy.

We configured a redundant Web server application in CA AppLogic as we mentioned above, to verify the fault tolerance. We performed several tests to evaluate how well the application recovered from failures on the external (public) and backbone (private) networks. The goal was to show that when a network failure occurs at any point, little or no interruption in network traffic occurs.

We used WebBench 5.0, an industry-standard benchmark for Web server software and hardware, to simulate network traffic in our test network. WebBench generates HTTP 1.0 GET requests to the Web server and shows a results metric of requests per second. We simulated system failures with WebBench running and monitored the requests per second as well as network traffic to see how well the system recovered from the failure.

External network failure

We unplugged the power to one of the network switches to initiate an external (public) network failure and see how well the system recovered. To do this, we ran a 15-minute WebBench test, which ran in three separate 5-minute mixes, to simulate network traffic to our Web application. We unplugged the external switch during the second mix to demonstrate normal network traffic followed by network traffic after external switch failure. Downtime was minimal, as network traffic dropped for 3 seconds before failing over to the other cloud switch. Figure 2 illustrates the network traffic pattern we saw in our test.

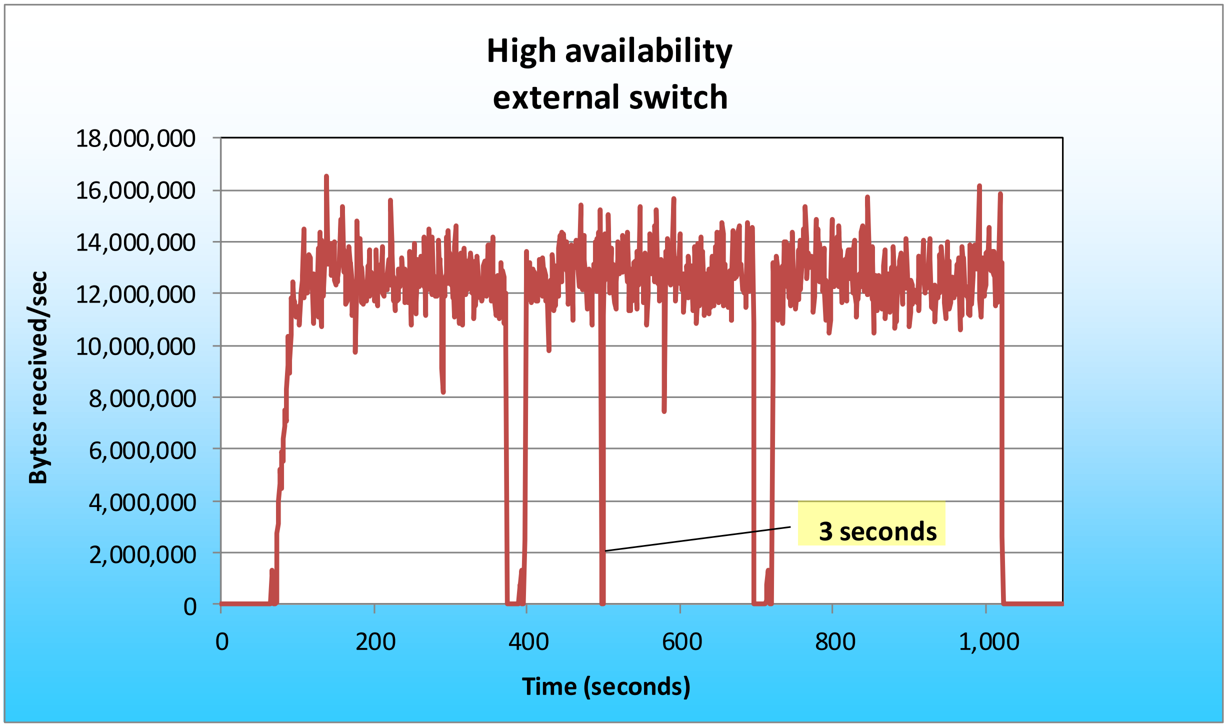


Figure 2: Our external switch high-availability scenario encountered only 3 seconds of downtime.

Backbone network failure

We unplugged the power to one of the network switches to initiate a backbone (private) network failure and see how well the system recovered. To do this, we ran a 15-minute WebBench test, which ran in three separate 5-minute mixes, to simulate network traffic to our Web application. We unplugged the backbone switch during the second mix to demonstrate normal network traffic followed by network traffic after backbone switch failure. Downtime was minimal, as network traffic dropped for 2 seconds before failing over to the other cloud switch. Figure 3 illustrates the network traffic pattern that we saw in our test.

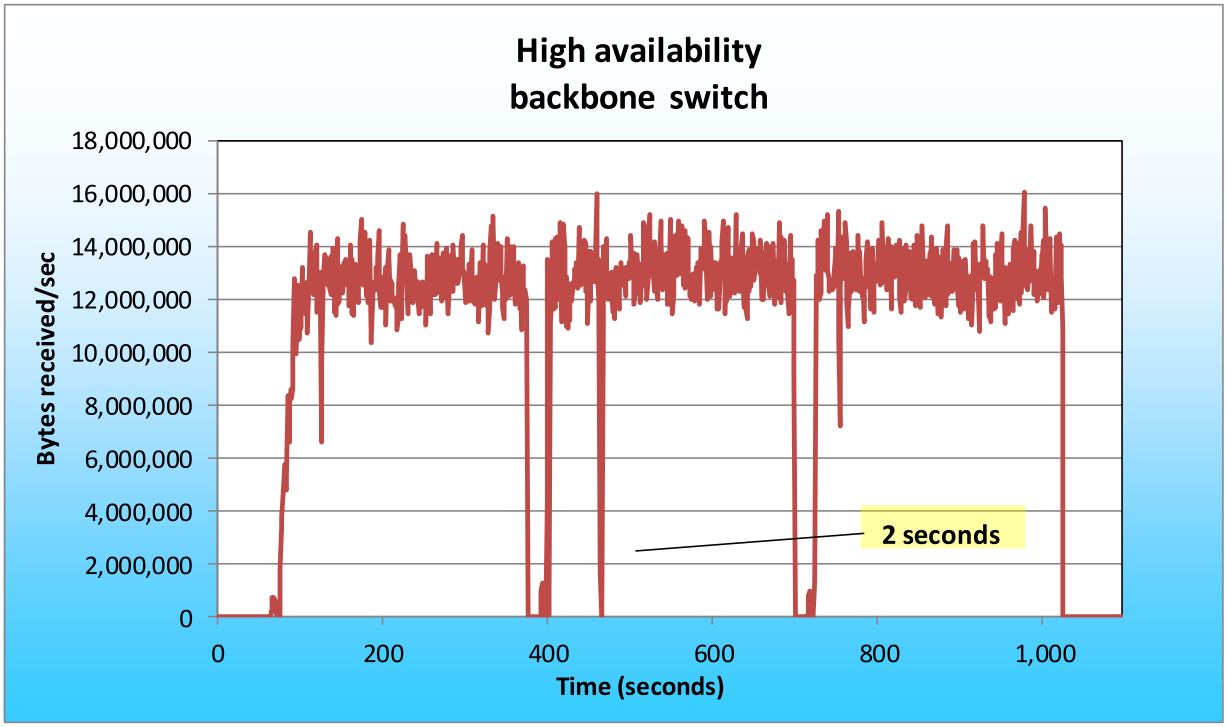


Figure 3: Our backbone switch high-availability scenario encountered only 2 seconds of downtime.

Bandwidth provisioning

CA AppLogic allows the user to set bandwidth for each appliance inside the application. For our evaluation, we configured yet another application, in this case a simple Web server application so we could easily modify and monitor the network bandwidth utilization. We provide more details on this Web server application in the How We Tested section.

We used WebBench to simulate network traffic with the Web server set to two different bandwidth provisions. We recorded the network bandwidth with the Web server bandwidth set to 250 Mb and 500 Mb. We monitored the network bandwidth from inside the application and from the Web clients to ensure we never over-utilized the network in both configurations.

To show the capabilities of bandwidth provisioning, we show two scenarios where the Web clients do not surpass their bandwidth provisioning. To do this, we made sure all other pieces of our application were powerful enough so as not to be bottlenecks that would influence our observed results. The graphs in Figures 4 and 5 illustrate how doubling the network bandwidth effectively doubles the bandwidth observed at the Web server appliance.

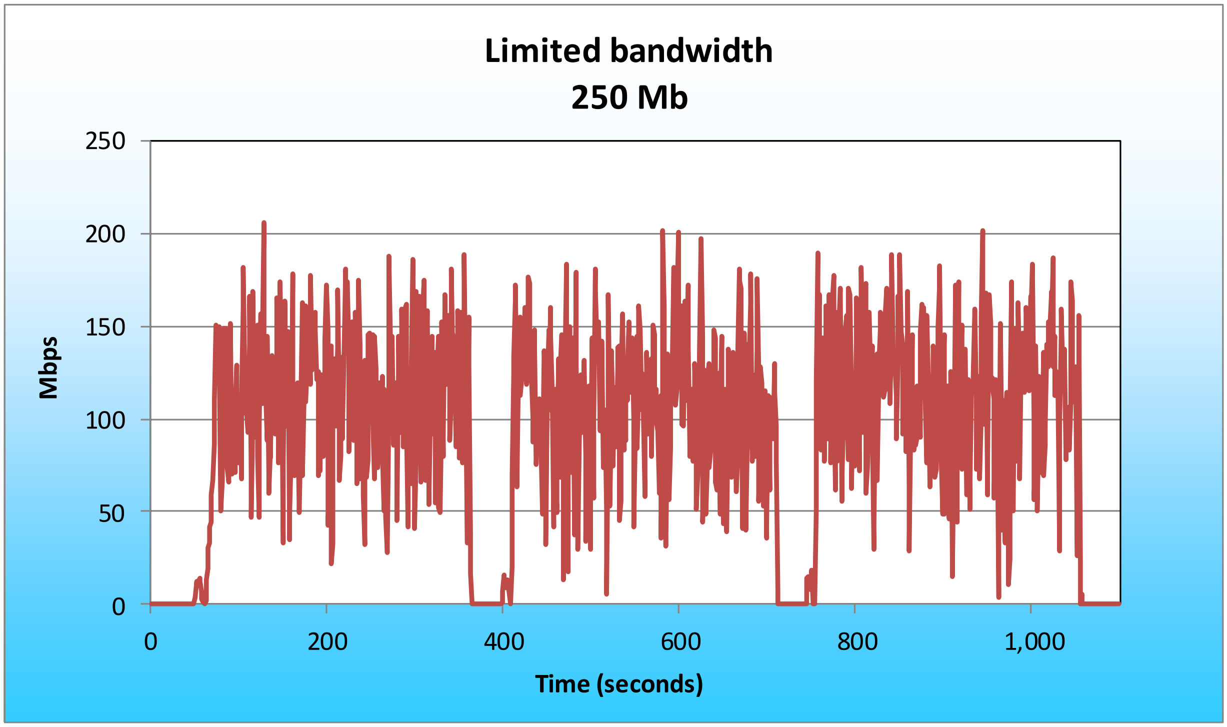


Figure 4: Network bandwidth with a limitation of 250 Mb.

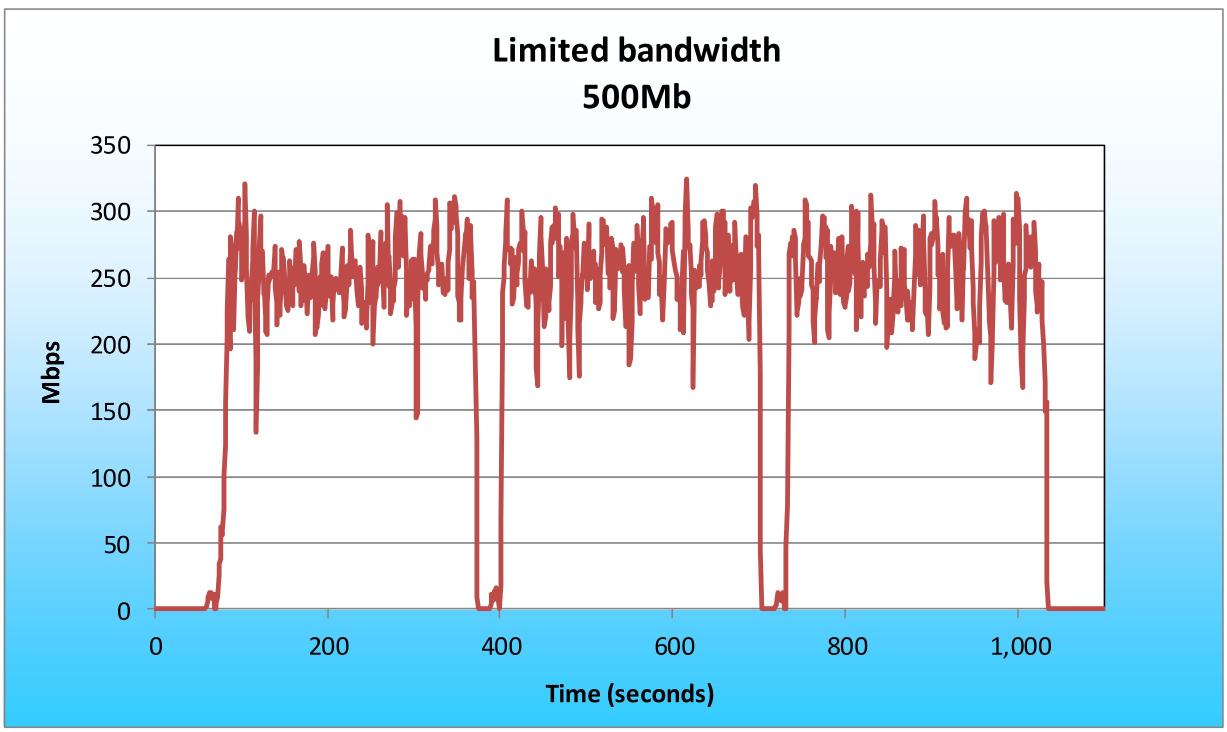


Figure 5: Network bandwidth with a limitation of 500 Mb.

HOW WE TESTED

One advantage of using CA AppLogic is that the user does not need identical server configurations for his or her cloud infrastructure. For testing, we used three Dell PowerEdge 2950 servers with different hardware configurations to configure the cloud. We used a fourth Dell Power Edge 2950 as the deployment server. Figure 1 in the What We Found section illustrates our test environment. See Appendix A for detailed configuration information for the test servers.

Configuring high availability functionality

Figure 6 shows the redundant Web server application we used for high availability functionality testing. We dragged all components from the left pane windows to the application editor main window. We connected all components together as we outlined. We added monitoring to view data behavior during testing.

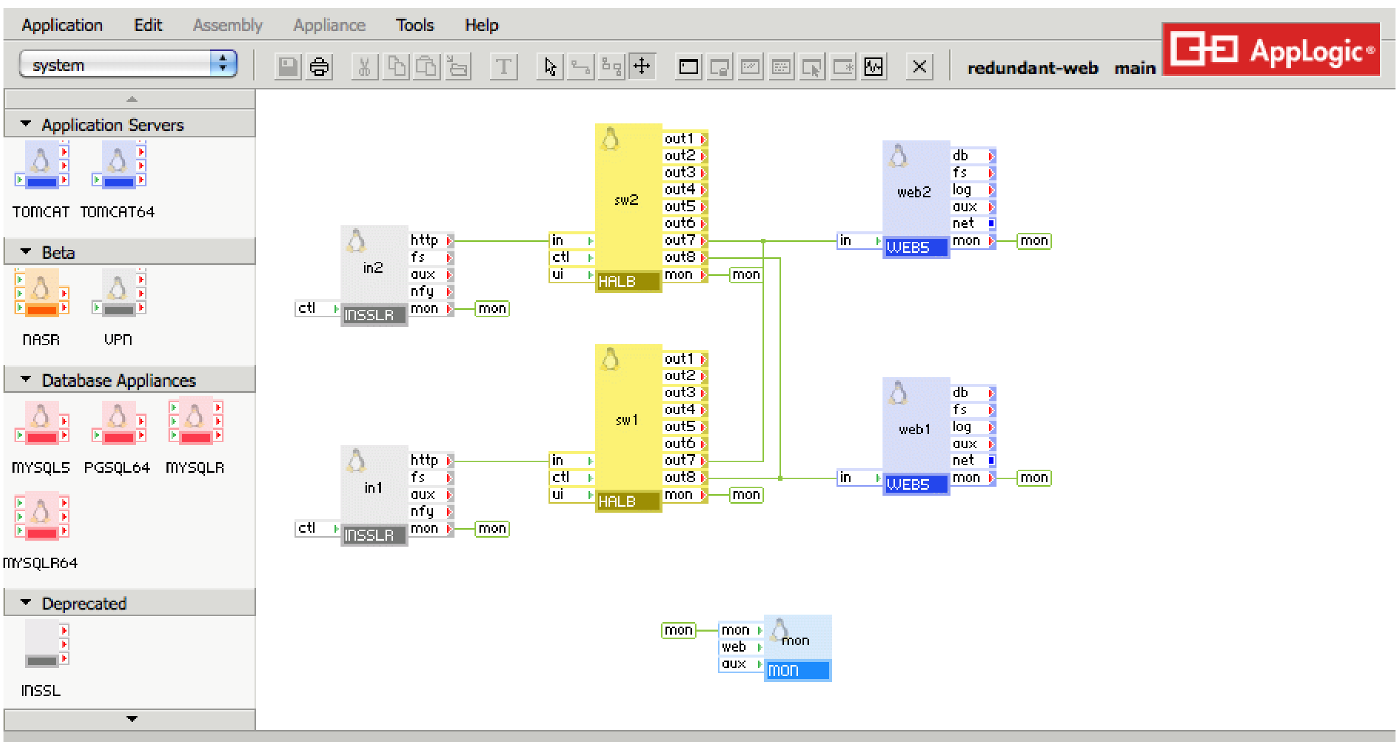


Figure 6: The redundant Web server application we used for our evaluation.

Configuring the gateways

We made the following configuration options to the *in1* gateway and *in2* gateway:

- ip_addr: `172.16.84.102`
- netmask: `255.255.255.0`
- gateway: `172.16.84.1`

- l7_accept: `http`
- fover_mode: `symmetric`
- fover_local_ip: `192.168.100.1 (in1); 192.168.100.2 (in2)`
- fover_remote_ip: `192.168.100.2 (in1); 192.168.100.1 (in2)`
- fover_netmask: `255.255.255.0`

Configuring bandwidth provisioning

Figure 7 shows the Web server application we used for bandwidth provisioning testing. We used a simple, non-redundant application configuration to simplify testing. We modified the resources of the Web server to restrict bandwidth. We added monitoring to view data behavior during testing.

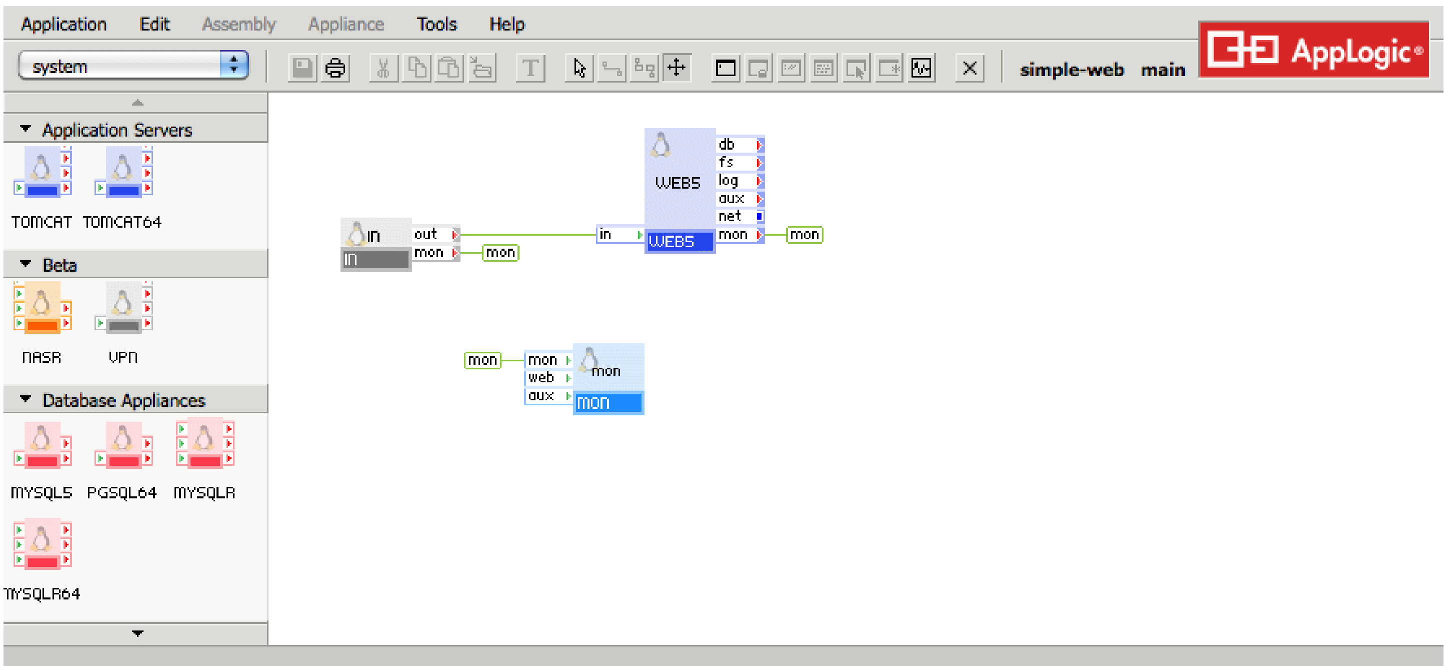


Figure 7: The simple Web server application we used for our evaluation.

Configuring the gateway

We made the following configuration options to the *in* gateway:

- ip_addr: `172.16.84.103`
- netmask: `255.255.255.0`
- gateway: `172.16.84.1`
- iface2_protocol: `tcp`
- iface2_port: `443`

APPENDIX A – SERVER CONFIGURATION INFORMATION

Figure 8 provides detailed configuration information about the test servers.

System	Dell PowerEdge 2950 deployment server	Dell PowerEdge 2950 server 1	Dell PowerEdge 2950 server 2	Dell PowerEdge 2950 server 3
Power supplies				
Total number	1	1	1	1
Vendor and model number	Dell 7001072-Y000	Dell 7001072-Y000	Dell 7001072-Y000	Dell 7001072-Y000
Wattage of each (W)	750	750	750	750
Cooling fans				
Total number	4	4	4	4
Vendor and model number	NMB-MAT 2415KL-04W-B96	NMB-MAT 2415KL-04W-B96	NMB-MAT 2415KL-04W-B96	NMB-MAT 2415KL-04W-B96
Dimensions (h x w) of each	2” x 2”	2” x 2”	2” x 2”	2” x 2”
Volts (V)	12	12	12	12
Amps (A)	2.10	2.10	2.10	2.10
General				
Number of processor packages	2	2	2	2
Number of cores per processor	4	4	4	4
Number of hardware threads per core	1	1	1	1
CPU				
Vendor	Intel®	Intel	Intel	Intel
Name	Xeon®	Xeon	Xeon	Xeon
Model number	E5440	E5405	E5405	E5405
Socket type	LGA771	LGA771	LGA771	LGA771
Core frequency (GHz)	2.83	2.00	2.00	2.00
Bus frequency (MHz)	1,333	1,333	1,333	1,333
L1 cache (KB)	128	128	128	128
L2 cache (MB)	12	12	12	12
Platform				
Vendor and model number	PowerEdge 2950	PowerEdge 2950	PowerEdge 2950	PowerEdge 2950
Motherboard model number	0J250G	0M332H	0M332H	0M332H
Motherboard chipset	Intel 5000X	Intel 5000X	Intel 5000X	Intel 5000X
BIOS name and version	Dell 2.6.1	Dell 2.6.1	Dell 2.6.1	Dell 2.6.1
BIOS settings	Default	Default	Default	Default
Memory module(s)				

System	Dell PowerEdge 2950 deployment server	Dell PowerEdge 2950 server 1	Dell PowerEdge 2950 server 2	Dell PowerEdge 2950 server 3
Total RAM in system (GB)	32	16	16	16
Vendor and model number	Hynix HYMP151F72CP 4N3-Y5	Samsung M395T5750EZ4 -CE66	Samsung M395T5750EZ4 -CE66	Samsung M395T5750EZ4-CE66
Type	PC2-5300F	PC2-5300F	PC2-5300F	PC2-5300F
Speed (MHz)	667	667	667	667
Speed running in the system (MHz)	667	667	667	667
Timing/latency (tCL-tRCD-tRP-tRASmin)	5-5-5-15	5-5-5-15	5-5-5-15	5-5-5-15
Size (GB)	4	2	2	2
Number of RAM module(s)	8	8	8	8
Chip organization	Double-sided	Double-sided	Double-sided	Double-sided
Rank	Dual	Dual	Dual	Dual
Hard disk				
Vendor and model number	Western Digital WD1602ABKS	Western Digital WD1602ABKS	Seagate ST973402SS	Western Digital WD1602ABKS
Number of disks in system	2	2	3	3
Size (GB)	160	160	73	160
Buffer size (MB)	16	16	16	16
RPM	7,200	7,200	10,000	7,200
Type	SATA	SATA	SAS	SATA
Disk controller	Dell SAS 6/IR	Dell SAS 6/IR	Dell PERC 6/I	Dell SAS 6/IR
Operating system				
Name	CentOS Release 5.5	CentOS Release 5.3	CentOS Release 5.3	CentOS Release 5.3
Kernel release	2.6.18-194.el5	2.6.18.8-xen0	2.6.18.8-xen0	2.6.18.8-xen0
Kernel version	SMP Fri Apr 2 14:58:14	#1 Thu May 13 11:28:43	#1 Thu May 13 11:28:43	#1 Thu May 13 11:28:43
File system	ext3	ext3	ext3	ext3
Language	English	English	English	English
Ethernet				
NIC Type 1				
Vendor and model number	Broadcom® 5708 NetXtreme® II 5708 Ethernet Adapter	Broadcom 5708 NetXtreme II 5708 Ethernet Adapter	Broadcom 5708 NetXtreme II 5708 Ethernet Adapter	Broadcom 5708 NetXtreme II 5708 Ethernet Adapter
Type	Onboard	Onboard	Onboard	Onboard

System	Dell PowerEdge 2950 deployment server	Dell PowerEdge 2950 server 1	Dell PowerEdge 2950 server 2	Dell PowerEdge 2950 server 3
NIC Type 2				
Vendor and model number	N/A	Intel 82571EB Quad Port Low Profile Adapter	Intel 82571EB Quad Port Low Profile Adapter	Broadcom 5709 NetXtreme II Ethernet Adapter
Type	N/A	PCI-E	PCI-E	PCI-E
Optical drive(s)				
Vendor and model number	TEAC CD-ROM CD-224E-N	TEAC CD-ROM CD-224E-N	TEAC CD-ROM CD-224E-N	TEAC CD-ROM CD-224E-N
USB ports				
Number	4	4	4	4
Type	2.0	2.0	2.0	2.0

Figure 8: Configuration details for the test servers.

ABOUT PRINCIPLED TECHNOLOGIES

Principled Technologies, Inc.
1007 Slater Road, Suite 300
Durham, NC, 27703
www.principledtechnologies.com

We provide industry-leading technology assessment and fact-based marketing services. We bring to every assignment extensive experience with and expertise in all aspects of technology testing and analysis, from researching new technologies, to developing new methodologies, to testing with existing and new tools.

When the assessment is complete, we know how to present the results to a broad range of target audiences. We provide our clients with the materials they need, from market-focused data to use in their own collateral to custom sales aids, such as test reports, performance assessments, and white papers. Every document reflects the results of our trusted independent analysis.

We provide customized services that focus on our clients' individual requirements. Whether the technology involves hardware, software, Web sites, or services, we offer the experience, expertise, and tools to help our clients assess how it will fare against its competition, its performance, its market readiness, and its quality and reliability.

Our founders, Mark L. Van Name and Bill Catchings, have worked together in technology assessment for over 20 years. As journalists, they published over a thousand articles on a wide array of technology subjects. They created and led the Ziff-Davis Benchmark Operation, which developed such industry-standard benchmarks as Ziff Davis Media's Winstone and WebBench. They founded and led eTesting Labs, and after the acquisition of that company by Lionbridge Technologies were the head and CTO of VeriTest.