Architecture for Scaling a Three-Tier Application

Scaling to meet the load demand of customers is a critical issue for users of web-based applications. The three-tier approach presents an opportunity to provide a more robust environment for configuring, testing, deploying, maintaining, and scaling Web applications that are deployed across a server farm.

Component-based load balancing is based on the fact that there is one or more computers receive a request for a page, then forwards that requests to servers within the application layer. The application layer is one or more servers that are configured to service requests from a web server; only requests for dynamic content is passed to the application layer. Static content is served by the web server or servers.

The web server routing requests to the application layer must get information from the application monitor. Then it is able route requests to the appropriate application server and uses the application monitor to poll the application servers for information regarding the current load on the server. Timing data is maintained on each component configured to be load balanced from the web server. The application monitor uses the timing data to rank the servers and control which server will receive the next request.

Multiple or large numbers of concurrent users

Our applications, architected in the three-tier model, support scaling several different ways. First, we are able to add multiple web servers that handle http/HTTPS traffic. Static data is cached on the web servers, such as images, text that doesn't change, style sheets, etc. These servers have connections distributed among them via DNS and

router load balancing. This facility also helps address failover – a URL can be mapped to one or more hosts, and host names can be aliased among available web server.

Further, these web servers may reside in different data centers. For example, we manage a web application for an international think tank that provides collaboration to various users. The web servers for the application are in data centers in suburban Washington DC and Toronto, Canada. In the unlikely event of catastrophic failure at the Internet point-of-presence for either city, HTTP traffic is seamlessly (and indivisible to the user) re routed to the other web servers. This configuration also helps balance traffic from European points-of-presence to another environment, and time-zone based load balancing ensures capacity is available to the east coast of the U.S. or to western Europe depending on business hours.

Besides multiple web servers, load balancing and failover, we are also able to offer higher traffic throughput to the web applications by adding more instances of the application on the application server, by adding more application servers, or both. This means that in a transaction-intensive environment (such as financial services), we can ensure the application capacity to handle processing of session data.

Finally, at the database level, we can use clustering or mirroring to provide substantial database reliability and capacity. In the case above, the actual web server-app server-database server combinations are duplicated in both data centers. Database synchronization occurs to mirror changes made on both systems; either system could support users in the event the other becomes unavailable.

Apache, the web server environment we use most, provides more than adequate capacity for large numbers of concurrent users. We craft components of our applications to work sessionless whenever possible, leveraging the application resources available.

Exhibit 1 shows the basic request-response loop when a user accesses a web application.

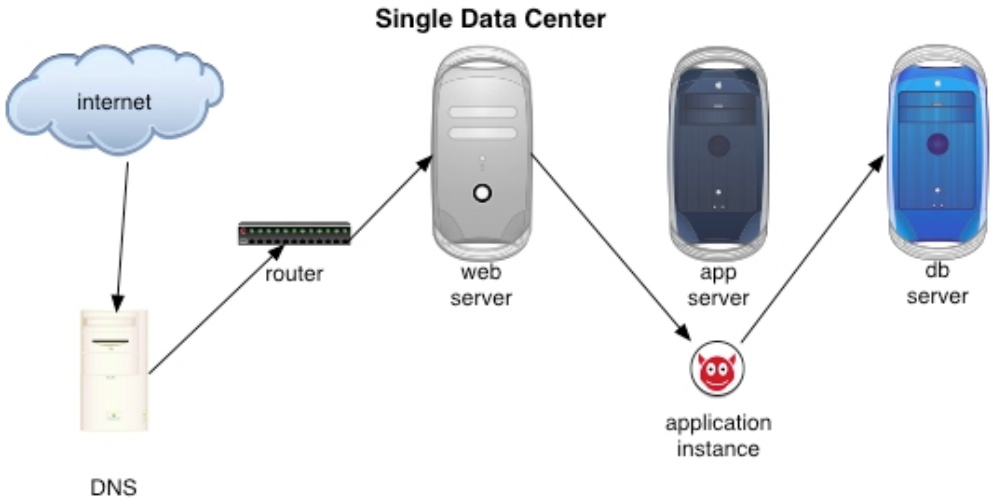

Exhibit 1. Basic Request – Response Loop

Exhibit 2 shows how this approach scales by adding more web servers.

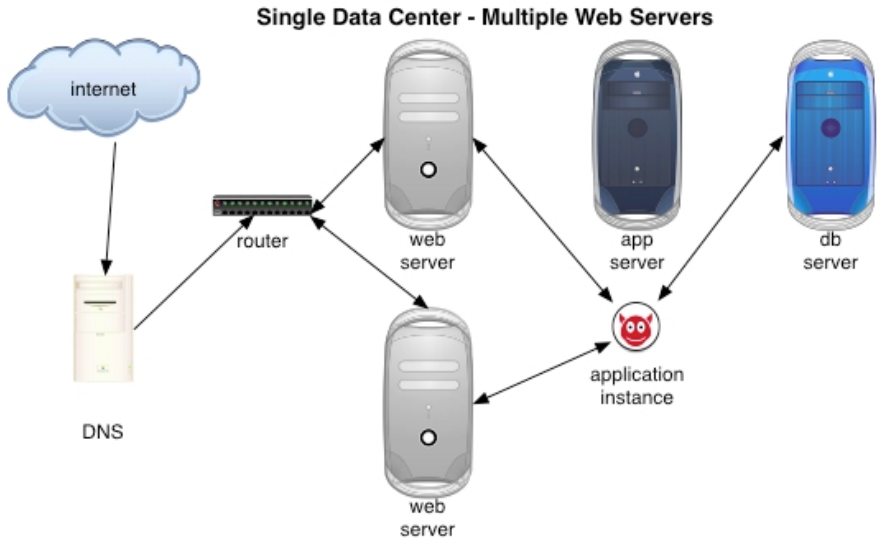

Exhibit 2. Adding Web Servers

Exhibit 3 shows how adding more application instances further increases load capacity.

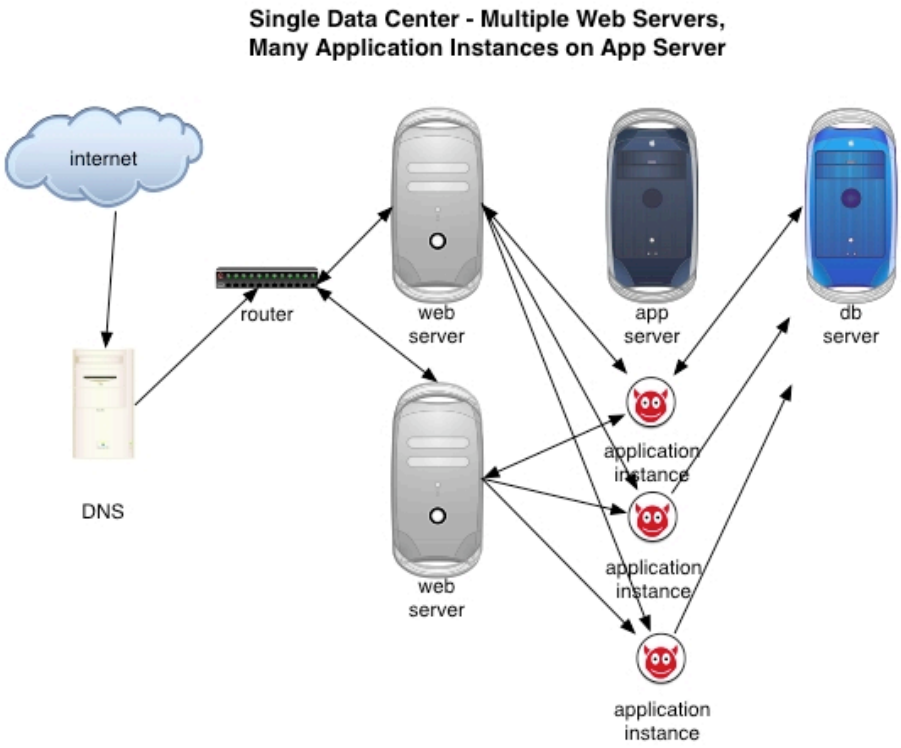

Exhibit 3. Adding Application Instances Increases Transaction Processing Capacity

Exhibit 4 shows adding more database servers to increase storage capacity and data management.

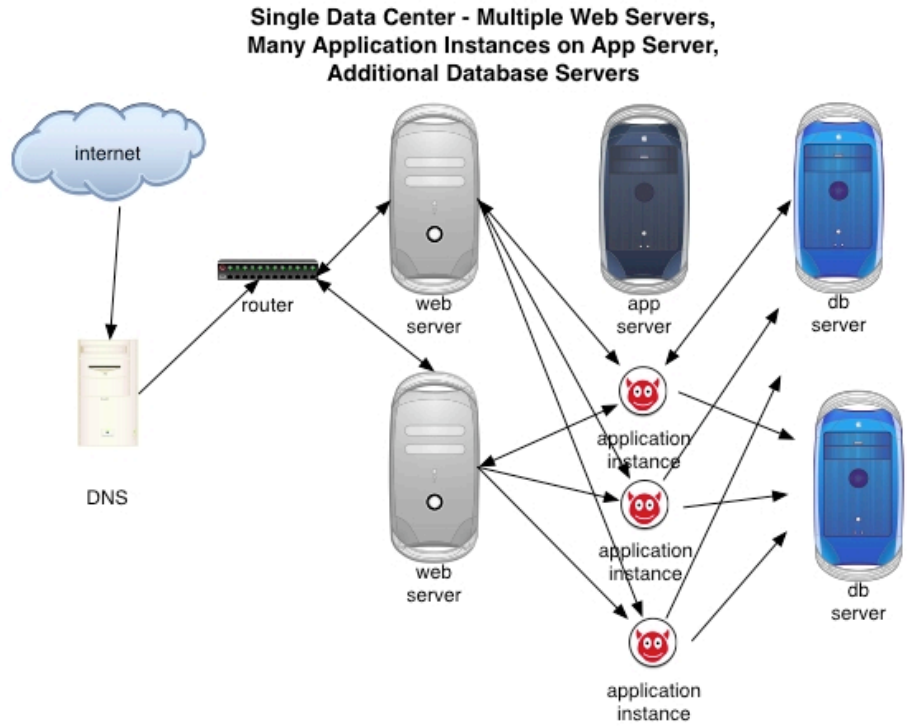

Exhibit 4. Adding Database servers increases data management capability

Exhibit 5 shows how adding more application servers increases throughput, responsiveness, and overall system capacity.

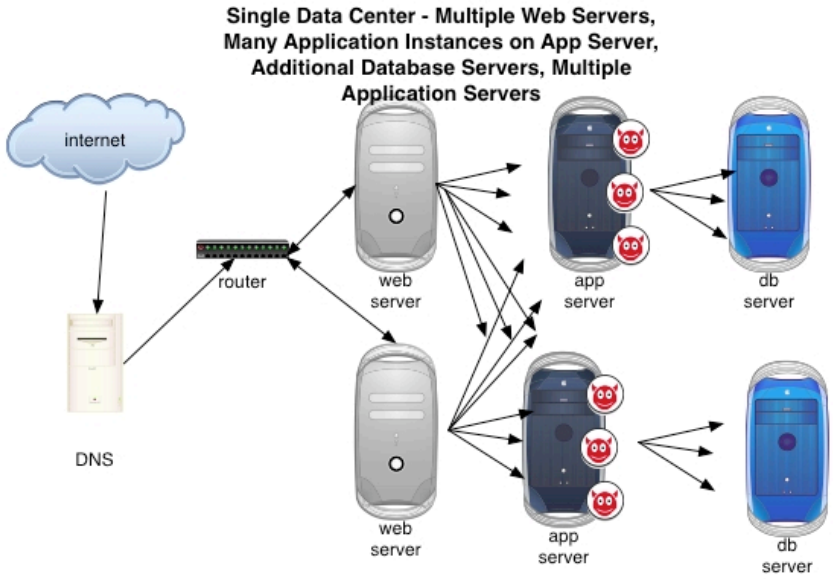

Exhibit 5. Adding Application Servers increase throughput

Exhibit 6 shows how multiple data centers supports scaling (as well as failover).

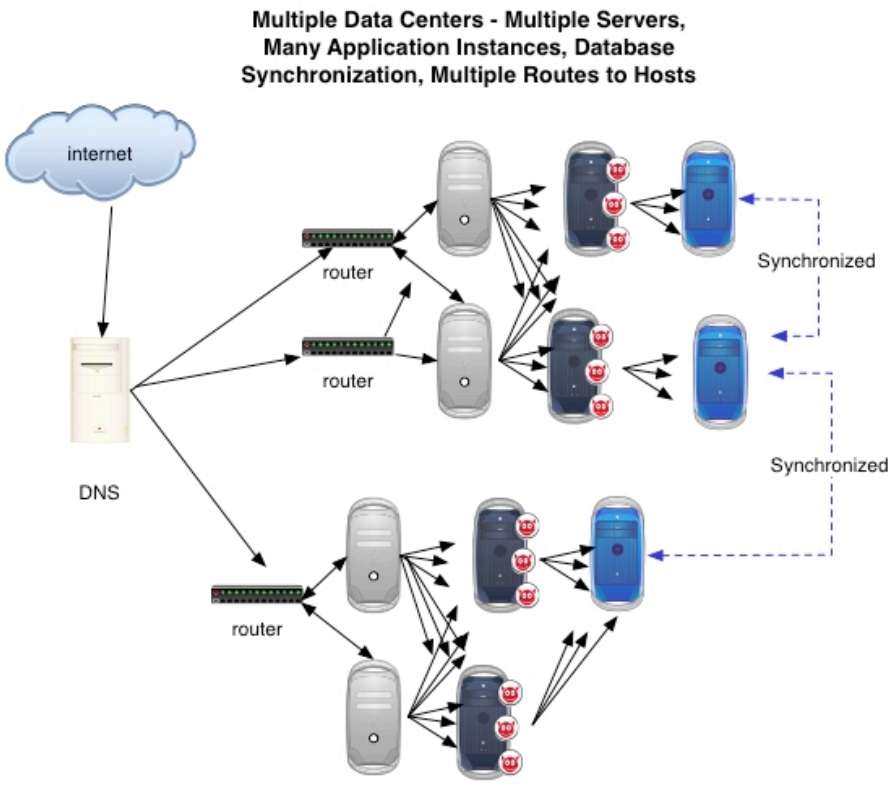

Exhibit 6. Multiple Data Centers Increase Capacity, Provides Failover and Redundancy, and Supports Intelligent Traffic Management

Page download speed

The architecture that supports scaling also improves the user experience in terms of page loading and responsiveness. Cached assets (on the web server, pushed down to the client's local cache, presented by the application through session caching, and even data query caches) ensures data is available and served up as quickly as possible. Pushing static or infrequently refreshed data and images to the edge of the network offloads server work, and keeps assets closest to the user.

Design of the GUI supports page download speeds – use of CSS means formatting is less dependent on slow-loading code and runs quickly as layout and format are processed in the browser engine. Distributed image assets, even from multiple web servers, increases page loading.

Compliance with Section 508 accessibility standards and offering a low-bandwidth option for legacy browsers or non-broadband connections matches the user environment to the user's facilities. We make best use of the separation of the presentation layer made possible in a three-tier architecture. Alternate interfaces (Palm OS or WAP, for example) are easily supported.

Maintaining the reliability and redundancy of systems and facilities.

We have a great deal of experience with network operations, including manning of the NOC, providing 24/7 customer support, and designing and maintaining enterprise network infrastructure. Our architecture (see above) supports redundant and highly reliable installations. We have government-approved security, backup and failover planning capability, and have devised high-availability environments for Homeland Security, Department of Defense, and the financial industries.

We offer a complement of network administration, infrastructure support, and end-user support. We will devise SOPs for backup and archiving of data, capacity planning, failover, and hot site operations.