

Multiple Servers (Load Balancing and Proxy)

The standard WebObjects deployment method using wotaskd has built-in support for Load Balancing between multiple servers or Proxying requests through to a different server.

Assume you have two servers:

- Server1 (www.server.com) will be the primary server and will be the gateway for all requests (the proxy).
- Server2 (www2.server.com) will be a secondary server that also runs WO application instances.

In the standard configuration requests are processed the following way.

Web Browser -> Apache on Server1 -> mod_WebObjects (the WO Adaptor) -> WOApplication (local or remote)

So the requests first go the apache, then apache passes it to mod_WebObjects (the WO Adaptor). The adaptor decides where to send the request next based on the information it has received from the wotaskd instances running on each server. wotaskd is configured via JavaMonitor and it tells the adaptor which servers are running this application and what the load balancing strategy is (typically Round Robin). If the next WO application instance to be used is running on the same server as Apache then the request is passed to the local app and processed directly. If the next app instance is running on a different server then the request is passed directly to the remote application instance on the port it is running on (like 2001) and the response is returned. Note that the request doesn't go through Apache again on the remote server - it is received directly by the WO application through its port, so you need to ensure that these ports are open for Server1 to access.

Two Servers

These are the components needed on each server:

Server1 (web server+WO application server). Serves ALL WebServerResources (static content) and processes 1/2 (or whatever you want) of WO requests proxies requests to Server2.

- Apache and mod_WebObjects
- wotaskd
- JavaMonitor
- WOApplication

Server2 (WO application server)

- wotaskd
- WOApplication
- (Optional) Apache & mod_WebObjects (possibly useful for debugging)

Dedicated Web Server (3 Servers)

Alternatively, if you wanted to use 3 servers with one being dedicated for Apache, you would need these things:

Server1 (web server). Serves ALL WebServerResources (static content) and proxies requests to the WO apps.

- Apache and mod_WebObjects

Server2 (WO application server)

- wotaskd
- WOApplication

Server3 (WO application server)

- wotaskd
- WOApplication

Setting it up

Adaptor (mod_WebObjects)

In the "3 server" example, notice that the web server (Server1) doesn't run wotaskd. How does it know where the WO application servers are and how to contact them? This information is contained in the configuration file for the WOAdaptor (typically /System/Library/WebObjects/Adaptors/Apache2.2/apache.conf) on the line:

WebObjectsConfig <http://www.server.com:1085>,<http://www2.server.com:1085> 10

Here you need to specify the addresses for each of your WO application servers. This is necessary in either a 2 server or 3+ server situation - it is how the adaptor communicates with wotaskd.

JavaMonitor

On the "Site" tab in JavaMonitor you can select a default load balancing algorithm. The default is "random". The "load average" option does not function correctly, so avoid using it. "Round robin" is the best choice.

Add the addresses for the WO application servers on the Hosts page. Then on the Application Detail page add instances to each server by choosing the right host name from the drop down box. Since "Round Robin" is the only useful load balancing algorithm that works currently you probably want to alternate servers as you add instances so you end up with something like this:

- Instance 1: server 1
- Instance 2: server 2
- Instance 3: server 1
- Instance 4: server 2
- Instance 5: server 1
- Instance 6: server 2

This way the Round Robin balancer will alternate requests between the two servers. You could also choose to target one server with more load than another by adding more instances to that server; so you don't have to split the traffic evenly.

Ports

You'll need to open up port 1085 on your application servers for local traffic (192.168) to wotaskd. You'll also need to open up the WOPort(s) your applications are running on (2001, 2002, etc) for local traffic (192.168) to the WO application servers.

Troubleshooting

To see if it's working you can just send requests to your web server - this is either a standalone web server or the combined web server +application server. All requests go to the web server address initially since it is the proxy for the WO applications. Then in JavaMonitor on the application detail page you can click "Show Details" and examine the "Transaction Count" column. Each instance should have an increasing number of transactions as you send requests.

If you are having problems with a separate application server, you can try send requests directly to that server.

First try sending them through the web server by appending a specific instance number to the URL for your app, like: <http://www.server.com/cgi-bin/WebObjects/App.woa/2>

The instance number should correspond to an instance that is running on the server you are trying to check.

Check to make sure the ports are open. You can check this using telnet. On your web server run these commands:

```
telnet www2.server.com 1085 # check wotaskd
telnet www2.server.com 2001 # check the WO application's port
telnet www2.server.com 2002 # check the WO application's port, etc
```

Use the port number that your WO application is running on. If you get the message "Connected to www.server.com" then the port is open.

If you have installed and configured Apache and mod_WebObjects on your secondary / standalone WO application server, then you can send requests to it through apache to see if the app is functioning correctly - like if it established a database connection, etc, with a url pointed at that server: <http://www2.server.com/cgi-bin/WebObjects/App.woa>

Apache is only used for troubleshooting here; it plays no part in the regular operation of your setup.