

# Are You Ready for the Holiday Rush?

## Five Survival Tips

Written by [Joseph Palumbo](#), Cloud Usability Team Leader

---

## Table of Contents

1. Vertical and Horizontal Scaling
2. Caching In
3. Web Server Tuning
4. Load Balancing
5. CDN Hosting

## Introduction

While consumers are thinking about which gifts to buy their friends and family, businesses are crossing their fingers hoping that their web platform will survive the holiday rush. Did you know that 40 percent of customers abandon an e-commerce website that takes more than three seconds to load?

It's no wonder that any business hosting a retail site would want to know how they can best prepare their web applications and environments for scaling traffic. In fact, this could be something you might have been tasked to figure out.

Our recommendation is to think about scaling traffic during the process of building your configuration, not after. You want to build today as if you're going to land on a famous talk show tomorrow.

# 1. To Scale Vertically or Horizontally?

When developing your application from the ground up, keep scalability in mind. Your application may run perfectly on your desktop or even after you've launched on the cloud, but how will it run when you receive a swarm of visitors trying to access your application at the same time? The cloud allows you to upgrade on the fly, but there will always be a cap. It's important to think about a strategy for growth before you launch into production.

## 1.1 Vertical Scaling

The architecture of Performance Cloud Servers does not support vertical scaling. For standard cloud servers, vertical scaling is the easiest. It essentially resizes your Cloud Server powered by OpenStack® with a click of a button and with no change to code. For example, you can go from a 1GB server instance to a 4GB instance by scaling vertically. There are limitations to vertical scaling because you can only get as big as the size of one server.

## 1.2 Horizontal Scaling

Horizontal scaling affords the ability to scale "wider" to deal with the traffic. Essentially, you could run your application on multiple servers, with the ability to add more servers that can help handle the traffic. However, horizontal scaling can't be done on a moment's notice; it requires some advance planning. You have to prep for horizontal scaling, which includes: (1) having a load balancer in front of your configuration (more on this later), (2) splitting out the layers of your configuration and (3) developing your application to take advantage of horizontal scaling.

If you have more questions on how to prepare your application for horizontal scaling, be sure to check out the Code to Scale in the Cloud blog or contact a Rackspace representative.

## 2. Caching In

Caching is an easy way to speed up your application or website (which can help with your bounce rate, saving you from potential lost revenue). Figure out what data you access frequently, and cache it in memory for repeated high-speed access to it. Whether your application is generating static content for web pages or storing sessions in caches, you have to decide how to store those caches. You can store the caches on your local file system or utilize distributed memory caches like memcached clusters.

If you use a file system to cache, you're going to want a shared file system, especially for session data. Keep in mind that disk I/O is expensive compared to memory. A memcached server receives better performance.

Depending on what platform you're running on, there are multiple ways you can implement caching. Here are a few at a glance:

- **Squid** — <http://www.squid-cache.org/>
- **Varnish** — <http://www.varnish-cache.org/>
- **PressFlow** (for Drupal websites) — <http://pressflow.org/>
- **W3 Total Cache** (WordPress websites) — <https://cloudtools.rackspace.com/apps/193?2030227621>

For static content, another simple and beneficial solution is to publish to a CDN (Content Delivery Network) instead of the web server. You can do this easily with services like Rackspace Cloud Files, built with Akamai's CDN technology. W3 Total Cache and PressFlow have built-in CDN technology as well. The files you elect to be public are shared over a content distribution network with locations all over the world.

The first time your content is served to a user, a copy of the content is stored in cache on the edge servers closest to that user's location. The next time the content is requested, it's pulled directly from the cached copy on the edge servers, reducing delivery time. For example, all the images published on our blog are stored on Rackspace Cloud Files.

As always, make sure to take backups. You can use Cloud Files to back up your instance – just set up automatic cron jobs to make backups of your MySQL data. It is recommended to store your source somewhere like github.

## 3. Web Server Tuning

Before launching into production, you'll want to do some web server tuning. You need to tune it for your application and for the instance size. For example, if your app takes up 30MB of RAM, tune Apache to limit the number of processes that a 1GB box would run.

Let's say you have six Apache processes for a 1GB instance. If you receive 32 concurrent visitors, your web server will fall over. It'll take some math skills to figure out how to prevent this from happening. Look at what your codes does and make sure Apache doesn't have modules you don't need.

If you're running PHP, for example, don't load the PHP GD module if you're not going to be loading images. If you're not using authentication for your web application, take the authentication module out. Just remember, if you get 16GB out of the box, you don't necessarily get 16GB — some memory is allocated for default apps.

As web traffic increases, your database may become slow at some point. You can look at making additional server-side fixes, like splitting reads and writes. Eight-five percent to 95 percent of apps will be read heavy which means you'll need more slaves than masters, such as MySQL multi-master replication or even MySQL slave-slave replication. This lets your app take advantage of the reads and writes. When you split the reads to a group of slaves, you can also split the writes to a master server.

## 4. Load Balancing

Load balancing as a way to scale horizontally. A Cloud Load Balancer is a virtual networking device that distributes traffic for maximum site availability. When using a Cloud Load Balancer with a public-facing IP address, this address essentially becomes the IP address of your website. As traffic is received, it's routed to your web servers using the pre-defined, load-balancing algorithm that you configure. The Cloud Load Balancer then acts as a crossing guard for web traffic, directing traffic to the path of least congestion so that your site is available whether you have 200 or 200,000 concurrent visitors. Rackspace offers load-balancing solutions that can be implemented as a dedicated service, or as a cloud-based service launched on-demand.

You'll know it's time to implement or reassess your load balancing solution when you're reviewing site traffic reports and seeing an increase in the number of connections refused during peak traffic times. That's the first sign that your serving capacity is too small for the amount of traffic that your site receives. The next sign will probably be visitors calling or emailing to complain that they can't access or transact on your site.

The number of load balancers you deploy is determined by your traffic and performance goals; there is no magic formula. It's important that you take the time to evaluate and stress test your system to ensure the load balancers work properly before traffic begins to spike. In the middle of a spike, it may be too late to mitigate the impact of increased traffic by adding a load balancer, and you'll be forced to tap more expensive solutions, such as dynamically resizing your server, or risk site outages and a diminished customer experience.

Some other things to keep in mind about load balancers:

- There is a feature that can be configured on the Cloud Load Balancer to allow for session persistence. This will direct incoming traffic from a given address to the same webserver node behind the Cloud Load Balancer. This prevents the customer from having to re-authenticate on your site if their traffic would have been sent to a different web server in the pool. For SSL encrypted websites employing a Cloud Load Balancer, the Session Persistence feature will not work. In this instance, it is advisable to use a database to store your session information.
- Balancing multiple SSL sites behind a single load balancer is not recommended. There are ways around it using Server Name Identification (SNI) or creating separate pools with load balancers for each site. Please consult with one of our Cloud Support specialists for more details on this particular solution.
- For optimal performance, it's not recommended to exceed more than 24 nodes behind a single Cloud Load Balancer.

## 5. Hosting Your Landing Page on the CDN

When you want to make sure that your web presence doesn't go down, you can host a static landing page from a Rackspace Cloud Files container with the Content Delivery Network (CDN) enabled. While you might lose dynamic functionality, this strategy will ensure that your visitors will not be hit with a timeout error. The key here is that as more people visit your site, your landing page will be cached across the globe for faster load times. The rest of your fully functional site can live on your scaled out web service layer. The technical information for how to host from a Cloud Containers can be found [here](#).

## SUMMARY

Preparation is the key. By working with a trusted hosting provider like Rackspace you can rest assured knowing the open sign for your online store hangs on your site all day, every day. Our infrastructure is built upon secure and scalable solutions that are not only customized to meet your needs, but are also backed 24x7x365 with our award-winning **Fanatical Support**<sup>®</sup>. We treat your site as if it was our own — our support team becomes an extension of yours.

## DISCLAIMER

© 2012 Rackspace US, Inc. All rights reserved.

This White Paper is for informational purposes only. The information contained in this White Paper is selected from public sources which we believe is reasonable in relation to subject matter and are believed to be accurate. RACKSPACE MAKES NO WARRANTIES OF ANY KIND, EXPRESS OR IMPLIED, AS TO THE INFORMATION OR SPECIFIC PRODUCTS. Users must take full responsibility for application of any product mentioned herein.

Rackspace and Fanatical Support are service marks of Rackspace US, Inc. registered in the United States and other countries. OpenStack is a trademark or registered trademark of OpenStack, LLC in the United States and/or other countries. All other product names and trademarks used in this document are for identification purposes only to refer to either the entities claiming the marks and names or their products, and are property of their respective owners. We do not intend our use or display of other companies' tradenames, trademarks, or service marks to imply a relationship with, or endorsement or sponsorship of us by, these other companies.