

Website Performance Best Practices

1. **Plan a testing strategy to test page load time and site capacity.** Page load testing indicates how fast pages load; site capacity determines the number of customers or concurrent visitors your site can handle during peak times. Both must be considered together as they each affect each other.
2. **Monitor mobile and desktop page load times separately.** Mobile users have an even lower tolerance for slow page loads than desktop users. Mobile testing must encompass multiple device types and operating systems.
3. **Optimize slow, highest-impact pages first – especially on mobile.** If you have a page that is rarely used (i.e., less than 0.5% of the traffic), then it may not make sense to optimize it. However, if it is your homepage or your product pages where each may contribute 20% of the traffic, then optimize them first. On a responsive site, optimization work on mobile load times will carry over to the desktop experience.
4. **Use actual web traffic as a basis for your load test plan.** Web logs are more accurate than analytics to determine total visitors, pages, and the percentage of load each page type receives. Once you know the total traffic and percentages of each page type request (e.g., the Homepage 10%, Level 1 category pages 4%, Product pages 12%), you can create realistic test cases to accurately test against the current load and peak load times (+50%).
5. **Test page load times under low load and while running high-capacity testing.** Servers can respond differently when under low or high load. Simply testing page load times under minimal load will miss the impact that traffic causes. In addition to low traffic situations, test each page while under heavy load (maybe 80% capacity).
6. **Test all possible page scenarios.** Although every page does not need to be tested, each page type should be tested to make sure that they all load properly and do not cause major server or load issues.
7. **Define your acceptability criteria for page load times.** Speeding up pages can be expensive, determine an acceptable speed and work toward that, make it reasonable.
8. **Separate page load times into component load times.** If 99% of a page loads in 1 second, but the other 1% loads in 20 seconds, then your page load time will never be better than 20 seconds. Optimize the slowest components on each page.
9. **Use logs to look at all page, application and database requests.** Use web and server logs to determine what happens in your server when pages are requested. There may be one simple request that runs slowly and can be fixed easily.
10. **Consider using a Content Delivery Network (CDN) for larger assets.** CDNs can effectively be used to reduce server load and page download times by offloading larger images, video, etc., to a geographically dispersed CDN that is closer to your customers than your servers.
11. **Identify where your customers are; investigate latency to their location.** If your servers are in California and many of your customers are in New York, look at how long it takes your pages to load in California and compare them to New York. If you find transmission time is a problem, CDNs can help.
12. **Plan future capacity based on current conversion rates and planned order volume.** To calculate concurrent user: $\#orders / conversion\ rate \times 20\%$ (or the percentage of orders in your busiest hour). Multiply this by your growth and increase during peak to make sure it can handle peak.
13. **Online monitoring services can continually monitor your site.** Hiring an online monitoring service can help you react to site slowdowns and spikes and alleviate potential issues as they are occur.
14. **Use the same monitoring service over time.** The point is to ensure measurement and reporting process consistency and allow for clean trend analysis (how things are changing over time).

**For 1000s of Best Practices and Advice across 100s of topics:
Join eCommerceKnow-How.com!**

eCommerce Diligence™ Checklist

Website Performance Key Questions to Ask Providers

Company

1. How long have you been in this business?
2. How many clients have you sold? How many clients have you lost, and why?
3. What clients fall into your “sweet spot”?
4. What peripheral services do you offer (e.g., CDN, frontend optimization, etc.)?
5. How many hosting location centers (POPs) do you have? Domestically? Globally?

Products/Services

1. Is your solution offered as perpetual licensed software or on-demand multi-tenant solution?
 - a. What are the client support requirements?
 - b. What is the process to add features you currently don't have?
 - c. What redundancy and disaster recovery do you have in place?
2. What skills does my organization need to use your tools? What training do you provide? Can I make my own changes to the software?
3. Describe the implementation process in detail.
4. What technical support services are available?
 - a. Are there human beings I can reach during reasonable hours (chat or phone) or do I have to wait over 24 hours for an email response or callback? Is your support team located in the US or abroad?
 - b. What is your SLA for support issues?
5. Do you have a support knowledge base, community forum, or applications that are shared by customers?
6. Can you work with our current CDN?

Features

1. Do you provide capacity testing solutions?
 - a. How many steps can be in each scenario?
 - b. Can scenarios navigate from non-secure to secure pages?
 - c. Do separate concurrent users maintain separate cookies?
 - d. Can different users add different products to the cart to prevent inventory run through? From Excel? How many scenarios can you run and support simultaneously?
 - e. What is the maximum number of concurrent users you can model in your capacity test?
 - f. What metrics or page load component times do you track under capacity tests?
 - g. What reports are available during and after a capacity test?
 - h. Do you provide network and server component test solutions? Which can you track and report?
2. What tools does your solution provide for managing and monitoring tests? Real-time?
3. Does your solution provide visual reporting or a dashboard view of reports in progress?
4. Does your solution provide integration (imports/exports) capabilities? Can you provide a list of partners or solutions (web analytics, server, QOS logs) you have integrated with in the past?
5. Does your solution provide for monitoring site performance?
 - a. What timeframes are possible?
 - b. Can it run load test, single page load tests?
 - c. What reporting or escalation processes are available?
- i. Can your solution test point-to-point response times? For example, can they imitate a user in Des Moines, IA accessing a server in California?
- j. Can your solution test network-hub-to-server response times?
- k. What other software, systems and web applications can you track and report?

Pricing

1. Please describe your pricing model.
 - a. Do you charge for number of page views, users, assets, etc.? If not, what metrics do you base your charges on? How are these metrics calculated and defined?
 - b. Different levels? Comparison matrix?
 - c. Why do I want to pay for upgrades?
2. How do you charge for annual support? What about maintenance and professional services?
3. Are there any additional fees?
 - a. Initial implementation? Ongoing changes? New feature developments? How do you handle overages and campaign bursts? Transaction costs?

**For 1000s of Best Practices and Advice across 100s of topics:
Join eCommerceKnow-How.com!**