The Unique Challenges of Enterprise WordPress

Jeff Matson
About Me

- Minister of Propaganda (the lord of all things technical content) @ Pagely
- Community, documentation, open source, and more @ NorthStack
- Tinkerer
- Arcade collector and restoration hobbyist
- Lover of creative solutions to complex problems
How Big Is Big?
TopClassActions.com

- 1.25M+ pageviews/month
- Weekly email blast to over 700k email subscribers
- 60 seconds of downtime = thousands of dollars in revenue lost
- #1 SEO rank for the “class action settlements” keyword
For TopClassActions.com, a single failed form submission can cause $8,000-10,000 in lost revenue
Scaling
Mistakes are fine when you’re small.

But how do you handle growth?
Expert Knowledge Is Critical
Knowledge Considerations

Do you have all of your bases covered?

- Server performance
- Code performance
- Security trends
- Event monitoring
Some issues don’t appear at the beginning

1. User visits form built with Gravity Forms with partial entries enabled.
2. Database query to get the form.
3. Database query to update the view count.
4. Request and database query when each field changes.
5. Database query on submission.
6. Database query on confirmation page.

(And we’re not even considering PHP processes)

Would you notice when you’re small? Probably not.
But it’s HUGE when running a website at scale.
Performance
Performance Impacts More Than Speed

When running at scale, even the little things can cripple you.

Poor performance impacts your overall site health and most importantly, your bank account.
How Poor Performance Impacts Sites

- Slow page load times (the obvious one)
- SEO impacts
- Increased server costs
- Unexpected downtime

A site that performs well is a happy site. Not only does it make it faster, but it also gets you more traffic, costs less, and overall is less headache-prone.
Things to Consider

- What's being served from cache?
- What's the a typical request look like from the server side?
- Are you offloading assets properly?
- How are your pages being generated?
- Are you actively monitoring your site's performance and overall health?
- Do you know what abnormal performance looks like?

Throwing more money at a problem should never be the solution.

There are times when upgrades are needed, but only after everything else is tuned properly.

If your site is performing poorly, a good host/sysadmin will be able to tell you where it’s coming from, how you can fix the problem, and (most of the time) offer server-side tweaks that are lessen the impact.
It’s 3:00 AM on a Tuesday.

Do you know what your website is doing?
Issue Prevention

Understanding the scenarios that impact your site will help you prevent them from becoming a real problem.
Things To Consider

- What’s a typical request look like?
- Am I caching things that don’t need to be dynamic on every request?
- Are there any trends that can be predicted and used in the future?
- Does anything vary from the previous trends I’ve witnessed?
- Is there anything that I can optimize?
- If something happens, how quickly can I react?
- If I need to react, how long will it take to resolve?
When the 🎆 Hits the Fan

It doesn’t matter how well you prepare for it. Something is going to go wrong.

The important part is how you deal with it.
Things To Consider During Chaos

- What’s the problem?
- Who’s being impacted?
- What’s causing the problem?
- What’s the short-term fix?
- What’s the long-term fix?
- Who’s fixing it?
- Could this happen again?
Security Hardening
The Bigger They Are, The Harder They Fall

Security breaches for a small site aren’t fun.

Security breaches on a large site can be monumental.
Something We All Know: Brute Force Attacks

It’s more than just an attempt to guess a password.

- The request still has to be processed.
- SQL queries still have to be executed.
- The attack still has to be recognized.

Can an unsuccessful attack still cost me money?

Can it be stopped before it’s even attempted?
Consider The Scary Stuff

- Do you know if your site is vulnerable?
- If someone were to hack your site today, would you know about it?
- Could you identify how the site was hacked?
- How would you assess the damage?
- How would you know if the threat is gone or still lurking?
- What would you need to do after it’s taken care of?
## Bringing It Together

<table>
<thead>
<tr>
<th>Problems</th>
<th>Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance is critical</td>
<td>Better code = better performance</td>
</tr>
<tr>
<td>High security risk</td>
<td>Stop attacks before they’re a problem</td>
</tr>
<tr>
<td>Downtime really hurts</td>
<td>Know the site well and watch for abnormalities</td>
</tr>
<tr>
<td>Things are bound to happen</td>
<td>Have someone who can react quickly</td>
</tr>
<tr>
<td>Big sites are expensive</td>
<td>Better performing code is significantly cheaper in the long run</td>
</tr>
</tbody>
</table>
Thank You!

Follow me on Twitter: @TheJeffMatson