

Open Source 101: Performance of Open Source Portal Software

Published: May, 2002

I have been a small part of the Open Source community since 1996 and I've been a regular Unix user since 1986. These technologies, which grew up on the Internet, offer compelling benefits for most organizations. A recent experience with an Open Source portal application, Drupal, pointed out to me just how good the performance of Open Source applications can be – when it is done correctly.

What Is Open Source?

At its very heart, Open Source is a philosophy that basically says "People should have access to the source code of their software and not be controlled by a vendor". While Open Source software is usually free, this definition says nothing about money - the Open Source movement is about freedom. It's the freedom to make changes as needed and freedom from being locked in by vendors. What organization, in today's technology world, hasn't been harmed by one vendor or another? Horror stories abound about bad vendors and with good reason. Open Source solves these issues once and for all by giving your organization full control.

NOTE: The term Open Source is itself controversial. Depending on the "license" agreement used, the term "Free Software" or "Free Open Source Software" is technically more accurate. In this document Open Source, Free Software and Free Open Source Software are synonymous.

Open Source Performance Issues Discussed

In software good performance is a function of good engineering. And good engineering is found in both commercial and open source applications. The interesting aspect of open source software (OSS) with respect to performance is that often OSS applications dramatically outperform commercial software by orders of magnitude – with tiny engineering teams and outdated hardware to boot. You should not, however, get the idea that all OSS has good performance. OSS can be as bad or worse as commercial software in certain areas.

Open Source Performance Illustrated: KernelTrap

Performance is always one of those things that computer people debate at great length. For myself, performance is best illustrated by example, not by statistic. On May 28, 2002, the web site www.kerneltrap.org was featured on Slashdot. Slashdot, for those unfamiliar with it, is a website which is extremely popular with computer people. It's a virtual certainty, that whenever a web site is featured on Slashdot, tens of thousands of people will view it within a very short time – generally

Open Source Software Performance

Why Can It Be Fast?

- It's personal – the engineering teams take speed seriously
- Projects are cash poor and time rich – it's hard to buy big servers when you are doing it for free
- Simple protocols and elegant architectures instead of complex, bloated design by committee
- Generally built by brilliant, passionate people

Why Can It Be Slow?

- Small teams that focus on features instead of architecture
- Inexperienced programmers
- Poor design choices

a few hours. This is now common enough that the phrase “my site was slashdotted” is actually common in web development circles. More than a few websites have been brought to their knees by the Slashdot effect.

The KernelTrap site runs an Open Source portal / content management platform known as Drupal, available via the GPL. Drupal, developed primarily in Europe, is built with PHP and MySQL. The Drupal team has taken the product up to version 3.0.2 and now is close to a release of their 4.0 platform. For more info about Drupal, please see <http://www.drupal.org>.

Actual Performance Statistics

Here are the actual statistics for the KernelTrap site for a portion of the month of May. The day of the Slashdot reference is near the bottom, in the wider, red row. It’s important to note that the overall traffic load increased **six fold** due to the Slashdot effect – yet Drupal kept running.

Daily Statistics for May 2002												
Day	Hits		Files		Pages		Visits		Sites		KBytes	
1	9203	1.86%	7126	1.76%	3777	2.02%	1959	2.08%	1360	2.90%	92448	1.35%
...												
21	18613	3.77%	14418	3.56%	6481	3.47%	2929	3.11%	2226	4.75%	180626	2.65%
22	16242	3.29%	12260	3.03%	6010	3.21%	2687	2.85%	1992	4.25%	163631	2.40%
23	14318	2.90%	10798	2.67%	4872	2.61%	2507	2.66%	1858	3.96%	139306	2.04%
24	14561	2.95%	11191	2.77%	5300	2.84%	2505	2.66%	1812	3.87%	144529	2.12%
25	10669	2.16%	8216	2.03%	3730	2.00%	1940	2.06%	1370	2.92%	106995	1.57%
26	9597	1.94%	7537	1.86%	3511	1.88%	1788	1.90%	1264	2.70%	91114	1.33%
27	12180	2.47%	9128	2.26%	4670	2.50%	2317	2.46%	1625	3.47%	117781	1.73%
28	75396	15.27%	68356	16.89%	30812	16.48%	19039	20.21%	15888	33.90%	1542238	22.59%
29	13790	2.79%	12280	3.03%	5344	2.86%	3267	3.47%	2989	6.38%	329425	4.83%

Real Time Performance Metrics

I am in fairly regular contact via Instant Messaging with one of the Drupal Team Leads and I learned about this in real time as it occurred. By auditing the Apache log files, as the Slashdot traffic was first arriving (i.e. when the reference to the site was “above the fold”), Kjartan gave me these statistics:

- ❑ Almost 16,000 hits from Slashdot.org and all it’s sub-domains in a 4 hour period
- ❑ Another 3,000 hits from other sites linking to the story
- ❑ 50 hits per minute on average, all served from a MySQL database through the Drupal code
- ❑ 100 hits per minute at times, again all served from a MySQL database through the Drupal PHP code

Hardware

Performance is relative to lots of different factors but, by illustrating it with specific examples, a good feel for it can be arrived at. Here's their current hardware and OS setup:

- ❑ A PII 300 mhz Intel server, single processor (that's about a \$300 box in today's dollars, if that much). This box runs the web server, database server and Drupal itself.
- ❑ 512 megs of RAM and 512 megs of Cache
- ❑ 100 megabit connection to the UUNet backbone
- ❑ RedHat Linux 7.2 with the 2.4.9 kernel
- ❑ Apache 1.3.22 & MySQL
- ❑ Drupal 3.0.2 (with caching)

Optimizations

The KernelTrap website took advantage of the standard Drupal caching option which reduces dynamic page generation to 2 database calls, 1 to fetch the user id and 1 to get the cached page from the database itself. An additional optimization, turning off unused modules, was not used at first. Once the site was getting >60 hits per minute, this became necessary. In real time, as Drupal was serving pages, the unused modules were turned off and performance increased. An additional tuning step of increasing the cached setting from 30 seconds to 10 minutes was also used. The total tuning time, admittedly by an experienced Drupal administrator, was **3 minutes**.

NOTE: This is not recommended. It is far better to optimize your site before this happens.

A Side Note: Support

Just as a side note, people often complain about support in the OSS world. Just as performance can be good or bad in the OSS world, so too can support be good or bad. In this case the quality of support in the OSS world was illustrated – as soon as one of the team leads for Drupal was alerted to the situation, he pitched in and helped tune the software. What happened here was that when the owner of the Kernel Trap site noticed that the Drupal page was maxing out on performance due to his own poor tuning (his term, not mine), he sent a request for help to the Drupal support mailing list. “Within minutes, I had support from Kjartan, Dries and Marco and through their help the tuning was quickly accomplished and I was back in business.” (source – email correspondence).

Summary

Clearly these performance levels are dramatically above those of competing applications be they open source or commercial software. These numbers speak volumes about the scalability of Drupal. If you have any concerns about performance of your web applications, you need to take a look at Drupal.

References

Download Drupal from <http://www.drupal.org/> and view a full community portal built with it at <http://www.drop.org/>. See http://kerneltrap.org/stats/usage_200205.html for the full performance analysis. KernelTrap is at <http://kerneltrap.org/>.

Disclaimer

The author of this document is also the author of a Drupal FAQ and is helping out on the project documentation.

Credits

Thanks go out to Jeremy Andrews and Guy Haas for assisting in the production of this document.