Availability, Scalability and Security with Drupal

FOSDEM / DrupalCon 2005

david.monosov@futureinquestion.net

Availability, Scalability, and Security with Drupal

- Drupal is a web application written in PHP, its infrastructure is composed out of a web server, a SQL database, and a PHP interpreter
- For all examples, we will assume your infrastructure of choice is the Apache web server with a modularized PHP interpreter and the MySQL database, running on either Linux or *BSD, this is believed to currently be the most common setup for running Drupal
- Some of these examples will hold valid with a similar setup based on Microsoft® Windows™ as the operating system of choice, modifications required for this environment will be noted where possible
- A marginal number of these examples will also probably apply if your web server of choice is Microsoft® IIS™, but don't take my word on it
- This presentation is not really about Drupal at all ;-)

Why this presentation isn't really about Drupal?

- Drupal is a web application - special, yet in many ways, similar to others before it
- Availability, scalability and security of web applications is a widely documented problem with many widely documented solutions
- Drupal does absolutely nothing to prevent these solutions from being applied, in fact, it has some facilities which makes doing so trivially simple in comparison to other web applications

Availability and Scalability – Introducing Horizontal Scaling

- The most common and affordable form of availability is achieved through redundancy
- The basic assumption is that if you throw more hardware at it, not all of it will fail simultaneously
- This is called horizontal scaling (as opposed to vertical scaling, where instead of throwing more hardware at a problem, we throw in bigger and better hardware with intrinsic availability - e.g. hot swappable components, RAID, etc.)
- These scalability forms are complimentary rather than exclusive - apply both to achieve Zen and sleep better at night

Horizontal Scaling for Redundancy in Practice

- n+1 servers containing the entire Drupal environment, configured for fail-over. If one stops functioning, the other picks up the workload
- In the most basic configuration, the servers operation is mutually exclusive - thus, we achieve availability, but not really any scalability at all

Horizontal Scaling for Redundancy in Practice (Linux)

- The Linux-HA project provides a tool, “heartbeatd”
- This tool allows for n+1 servers to check each other’s “heartbeat”, each server has one or more real IP addresses, and one or more virtual IP addresses are shared between the servers on which the actual Drupal site (and all its infrastructure) is provided
- The site’s hostname (e.g. www.drupalsite.com) points to the virtual address
- If the active server fails, another takes its virtual IP address and service continues nearly uninterrupted

Horizontal Scaling for Redundancy in Practice (Linux) - Caveats

- Only two:
  - All servers have to be aware of active sessions, making the failover transparent for users
  - The database needs to be shared and kept up to date amongst all servers
- Unlike many web applications, Drupal stores it’s session data in the SQL database, thus, there is really only one caveat

Horizontal Scaling for Redundancy in Practice (Linux) – Caveats

- One solution is to hold the database files on a networked, or otherwise shared by means of SCSI/FC storage array, and mount them to the active server during the failover process
- This is really rather simple, but not entirely fool-proof and requires quite significant investment in appropriate storage technologies

Horizontal Scaling for Redundancy in Practice (Microsoft® Windows™)

- Microsoft® makes the Microsoft Clustering Services available since Windows™ NT
- It works properly since Windows™ 2000
- It is only available in the ‘Advanced Server’, ‘Datacenter’ and ‘Enterprise’ editions, which incur additional licensing costs
- It offers all the same functionality of the “heartbeatd” solution, and requires little more than a two-button mouse to configure
- Getting it to work with Apache and MySQL rather than Microsoft® IIS™ and Microsoft® SQL Server however might require losing some hair, and is left as an exercise to adventurous members of the audience

Introducing Scalability

- n+1 servers can be better utilized if all of them are available to users simultaneously, however, this requires a mechanism to distribute the load
- The database then needs to be available to all the servers simultaniously

Scalability in Practice (Linux)

- To achieve scalability we need to share the client load between servers, this is called load balancing
- There is a handy tool for this too, called Ultramonkey
- Ultramonkey is a combination of “heartbeatd” and IPVS
- In this case, a virtual address is assigned to the n+1 Ultramonkey cluster, which, in turn, forwards the requests to individual IP addresses of the n+1 real web servers
- These web servers, in turn, either access an external MySQL server, or also act as a MySQL server and replicate data between themselves

Scalability in Practice (Linux) – Handling MySQL

- Since all servers can both SELECT and INSERT, UPDATE, or DELETE data from the database, having only one external MySQL server which is accessed through the network is the simplest solution
- It is not, however, very reliable
- Thus, an ideal solution would also include n+1 MySQL servers, which would replicate each operation amongst themselves, running either as an additional database cluster, or on each of the web servers

Scalability in Practice (Linux) – Handling MySQL

- MySQL offers master/master replication
- It is known to work fairly well in MySQL 4.x and up
- Still not entirely fool-proof, and may cause collisions if operations on the same data are preformed by different servers
- Sufficient for all but extremely mission-critical setups
- Database servers are easier to scale vertically than horizontally

Scalability in Practice (Microsoft® Windows™)

- Microsoft® makes the Network Load Balancer (NLB) available since Windows™ 2000
- It is also available only in the 'Advanced', 'Datacenter', and 'Enterprise' editions of various Microsoft® server products
- In combination with Microsoft® Cluster Services, it serves as a complete solution with much the same functionality as Ultramonkey

Scalability in Practice (Other Solutions)

- There are commercial load balancers, available as stand alone devices from many vendors, including F5, Nortel, and many others, some network switches also offer load balancing functionality (e.g. those of Extreme Networks)
- For true high availability and scalability, using a n+1 configuration of load balancers is still advised

Security (Look Ma’, My Page Says “w00t”)

- There are no quick and simple solutions for end-to-end security which can be detailed over a few pages of a presentation – unless you’re trying to sell one
- There are, however, a few best practices which, if followed, would make a setup a lot more secure

Security

- Always grant only the minimal set of privileges required to perform the task
- Firewall everything, everywhere
- Log everything, everywhere, and audit

Security (Drupal Specific)

- The Drupal core is under constant review by competent people, and releases can be considered fairly secure
- This situation changes drastically when looking at the contributions repository
- Do not use contributed code which hasn’t been reviewed unless you are willing to review it yourself, or have a good reason to trust its author

Thank your kindly,

for allowing me to waste some of your time.

I hope you found it informative! ;-)