



Scaling the drupal.org Infrastructure

Keeping up with exponential growth

Gerhard Killesreiter and Kieran Lal

`gerhard@killesreiter.de`

Drupal vzw

History of the drupal.org infra-structure



- Until autumn 2005: single server for Apache and MySQL, managed by Kjartan Mannes.

History of the drupal.org infra-structure



- Until autumn 2005: single server for Apache and MySQL, managed by Kjartan Mannes.
- Since then: SUN Fire v20z as MySQL-Server, two Dell as Apache server, hosted at Open Source Lab of the Oregon State University (OSUOSL).

History of the drupal.org infra-structure



- Until autumn 2005: single server for Apache and MySQL, managed by Kjartan Mannes.
- Since then: SUN Fire v20z as MySQL-Server, two Dell as Apache server, hosted at Open Source Lab of the Oregon State University (OSUOSL).
- Since early summer 2007: two load balancers on loan from OSUOSL

History of the drupal.org infra-structure



- Until autumn 2005: single server for Apache and MySQL, managed by Kjartan Mannes.
- Since then: SUN Fire v20z as MySQL-Server, two Dell as Apache server, hosted at Open Source Lab of the Oregon State University (OSUOSL).
- Since early summer 2007: two load balancers on loan from OSUOSL
- Since summer 2007: a MySQL-Slave on loan from OSUOSL.

History of the drupal.org infra-structure

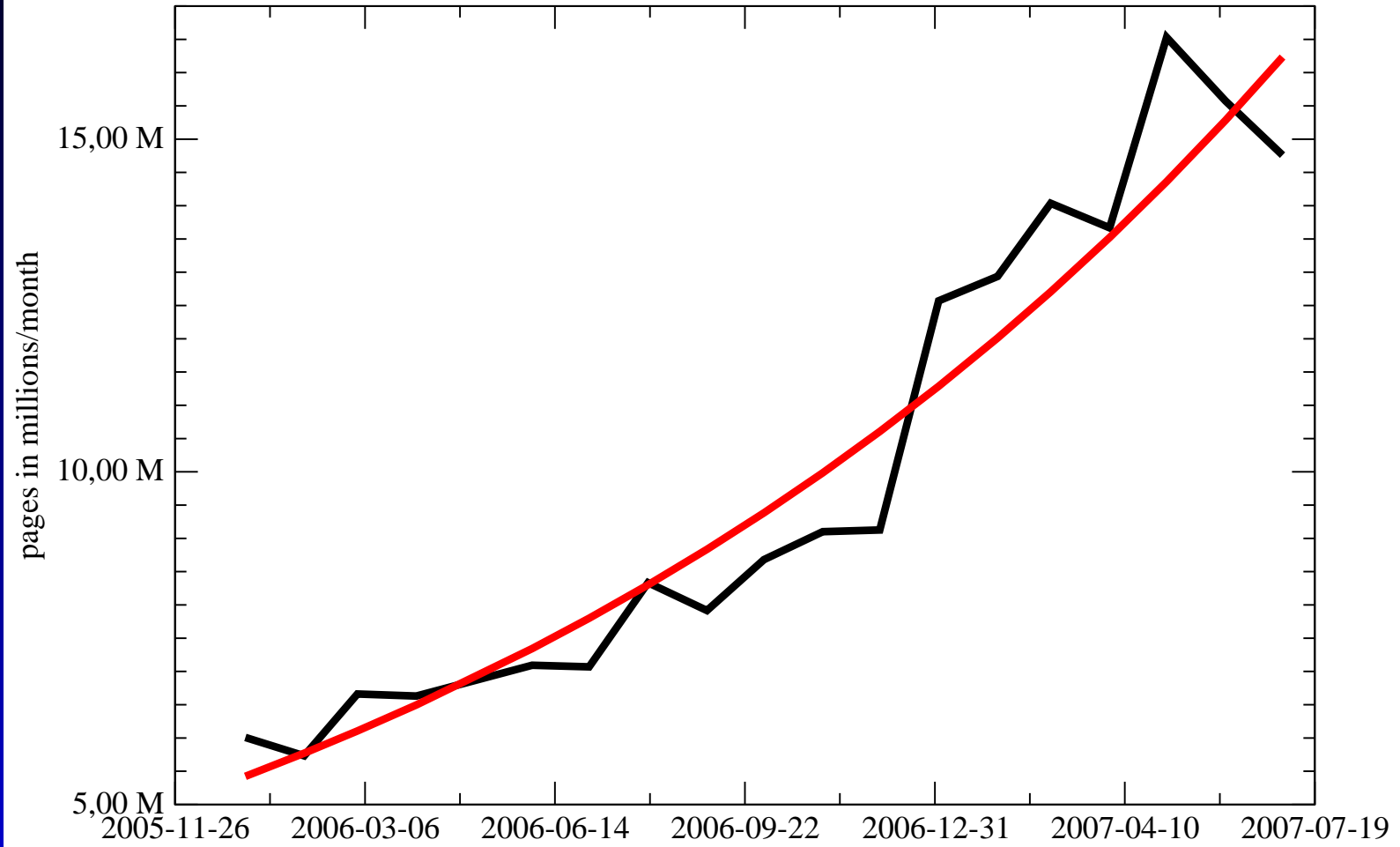


- Until autumn 2005: single server for Apache and MySQL, managed by Kjartan Mannes.
- Since then: SUN Fire v20z as MySQL-Server, two Dell as Apache server, hosted at Open Source Lab of the Oregon State University (OSUOSL).
- Since early summer 2007: two load balancers on loan from OSUOSL
- Since summer 2007: a MySQL-Slave on loan from OSUOSL.
- Since autumn 2007: new MySQL-Master bought by the Drupal association (not yet deployed).



Why? Exponential growth

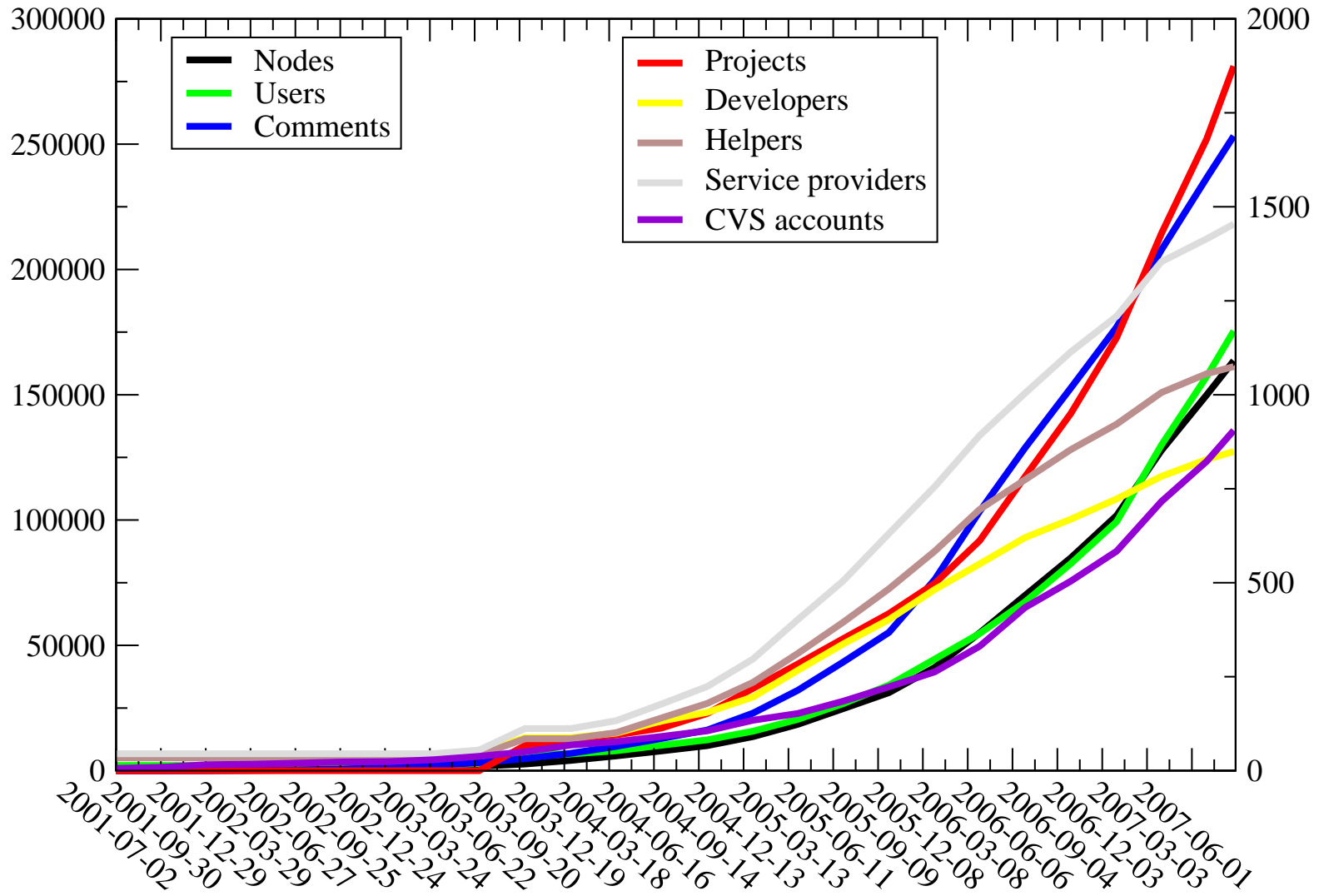
- Number of pages viewed by people grown exponentially from 6 to 16 Million per month.





Why? Exponential growth

- Number of pages viewed by people grown exponentially from 6 to 16 Million per month.
- Other indicators also grow exponentially.





Main problem

As soon as you improve the efficiency of the web server, the DB server will get all the extra load from the extra pages you are able to serve.



What did we try?

- PHP Opcode Cache (APC)



What did we try?

- PHP Opcode Cache (APC)
- Drupal improvements (more caching, less locking)



What did we try?

- PHP Opcode Cache (APC)
- Drupal improvements (more caching, less locking)
- Switched off statistics module.



What did we try?

- PHP Opcode Cache (APC)
- Drupal improvements (more caching, less locking)
- Switched off statistics module.
- change from MyISAM to InnoDB (and back for the search tables)



What did we try?

- PHP Opcode Cache (APC)
- Drupal improvements (more caching, less locking)
- Switched off statistics module.
- change from MyISAM to InnoDB (and back for the search tables)
- Evaluated „INSERT DELAYED“RR and friends



What did we try?

- PHP Opcode Cache (APC)
- Drupal improvements (more caching, less locking)
- Switched off statistics module.
- change from MyISAM to InnoDB (and back for the search tables)
- Evaluated „INSERT DELAYED“RR and friends
- Squid as reverse proxy



What did we try?

- PHP Opcode Cache (APC)
- Drupal improvements (more caching, less locking)
- Switched off statistics module.
- change from MyISAM to InnoDB (and back for the search tables)
- Evaluated „INSERT DELAYED“RR and friends
- Squid as reverse proxy
- More RAM: from 2 to 4 GB (webserver), 4 to 8 GB (DB server)



What did we try? II

- memcached



What did we try? II

- memcached
- Master/Slave DB-setup



What did we try? II

- memcached
- Master/Slave DB-setup
- Use http:bl blacklist to lock out illegitimate harvesters



What did we try? II

- memcached
- Master/Slave DB-setup
- Use http:bl blacklist to lock out illegitimate harvesters
- Extra caching for forum block and download pages



What did we try? II

- memcached
- Master/Slave DB-setup
- Use http:bl blacklist to lock out illegitimate harvesters
- Extra caching for forum block and download pages
- tune google down (100-150k pages/day vs. 10k pages/day)



What did we try? II

- memcached
- Master/Slave DB-setup
- Use http:bl blacklist to lock out illegitimate harvesters
- Extra caching for forum block and download pages
- tune google down (100-150k pages/day vs. 10k pages/day)
- Buy better server (done, is being burned in).



Things that didn't work so well

- Optimizing search failed (auto-fixed by upgrading to D5 at Jan 7th).



Things that didn't work so well

- Optimizing search failed (auto-fixed by upgrading to D5 at Jan 7th).
- Use a Sun T2000 server from OSUOSL (processor too slow).



Things that didn't work so well

- Optimizing search failed (auto-fixed by upgrading to D5 at Jan 7th).
- Use a Sun T2000 server from OSUOSL (processor too slow).
- memcached (page cache).



Things that didn't work so well

- Optimizing search failed (auto-fixed by upgrading to D5 at Jan 7th).
- Use a Sun T2000 server from OSUOSL (processor too slow).
- memcached (page cache).
- Fixing groups.drupal.org (much less traffic than drupal.org).



Things that didn't work so well

- Optimizing search failed (auto-fixed by upgrading to D5 at Jan 7th).
- Use a Sun T2000 server from OSUOSL (processor too slow).
- memcached (page cache).
- Fixing groups.drupal.org (much less traffic than drupal.org).
- http:bl blacklist only blocks about 2500 page requests/day.



Things that didn't work so well

- Optimizing search failed (auto-fixed by upgrading to D5 at Jan 7th).
- Use a Sun T2000 server from OSUOSL (processor too slow).
- memcached (page cache).
- Fixing groups.drupal.org (much less traffic than drupal.org).
- http:bl blacklist only blocks about 2500 page requests/day.
- Using fast-cgi (unexpected crashes, moved back to mod_php starting June 2007)



Things that worked well

- APC



Things that worked well

- APC
- Master/Slave setup (after initial problems)



Things that worked well

- APC
- Master/Slave setup (after initial problems)
- tuning google down



Things that worked well

- APC
- Master/Slave setup (after initial problems)
- tuning google down



Things that OSUOSL helped with



- Fixing PHP/Apache/APC setup

Things that OSUOSL helped with



- Fixing PHP/Apache/APC setup
- Setting up memcached

Things that OSUOSL helped with



- Fixing PHP/Apache/APC setup
- Setting up memcached
- Tweaking MySQL (including sending patches to MySQL AB)

Things that OSUOSL helped with



- Fixing PHP/Apache/APC setup
- Setting up memcached
- Tweaking MySQL (including sending patches to MySQL AB)
- Setting up Master/Slave

Things that OSUOSL helped with



- Fixing PHP/Apache/APC setup
- Setting up memcached
- Tweaking MySQL (including sending patches to MySQL AB)
- Setting up Master/Slave
- Moved servers to better datacentre.

Things that OSUOSL helped with



- Fixing PHP/Apache/APC setup
- Setting up memcached
- Tweaking MySQL (including sending patches to MySQL AB)
- Setting up Master/Slave
- Moved servers to better datacentre.
- Setting up Cacti to allow us monitor the servers

Things that OSUOSL helped with



- Fixing PHP/Apache/APC setup
- Setting up memcached
- Tweaking MySQL (including sending patches to MySQL AB)
- Setting up Master/Slave
- Moved servers to better datacentre.
- Setting up Cacti to allow us monitor the servers
- Put servers into cfengine to let them easier run them

Things that OSUOSL helped with



- Fixing PHP/Apache/APC setup
- Setting up memcached
- Tweaking MySQL (including sending patches to MySQL AB)
- Setting up Master/Slave
- Moved servers to better datacentre.
- Setting up Cacti to allow us monitor the servers
- Put servers into cfengine to let them easier run them
- Provide backup facilities

Things that OSUOSL helped with



- Fixing PHP/Apache/APC setup
- Setting up memcached
- Tweaking MySQL (including sending patches to MySQL AB)
- Setting up Master/Slave
- Moved servers to better datacentre.
- Setting up Cacti to allow us monitor the servers
- Put servers into cfengine to let them easier run them
- Provide backup facilities
- Provide FTP-Server for faster downloads

Things that OSUOSL helped with



- Fixing PHP/Apache/APC setup
- Setting up memcached
- Tweaking MySQL (including sending patches to MySQL AB)
- Setting up Master/Slave
- Moved servers to better datacentre.
- Setting up Cacti to allow us monitor the servers
- Put servers into cfengine to let them easier run them
- Provide backup facilities
- Provide FTP-Server for faster downloads
- Lots of other stuff I forgot



Plans for the future

- Get more webnodes



Plans for the future

- Get more webnodes
- If neccessary, buy more slave server



Plans for the future

- Get more webnodes
- If necessary, buy more slave server
- Fix Drupal to be able to run master/slave setup without hacks.



Plans for the future

- Get more webnodes
- If necessary, buy more slave server
- Fix Drupal to be able to run master/slave setup without hacks.
- Monitor system resource usage to get early hints on possible bottlenecks.