Wikipedia

A very cheap story

Me

- Edited Wikipedia in ~2003

- Started development work in 2004

- Database administration, performance engineering, capacity management since 2005

- Support Engineer at MySQL AB (now Sun) since 2005

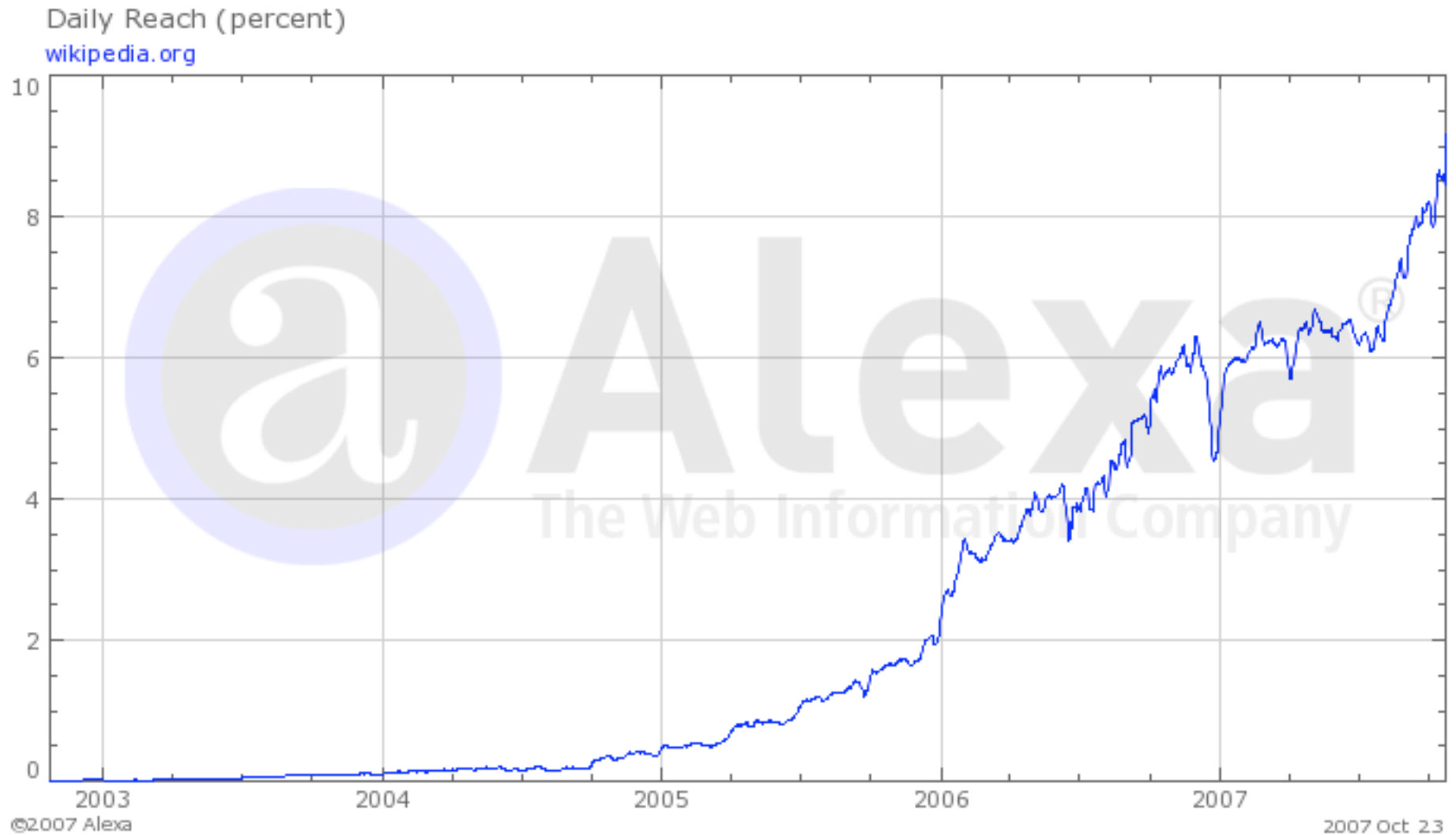- Filling board member seat at Wikimedia

The wiki experience
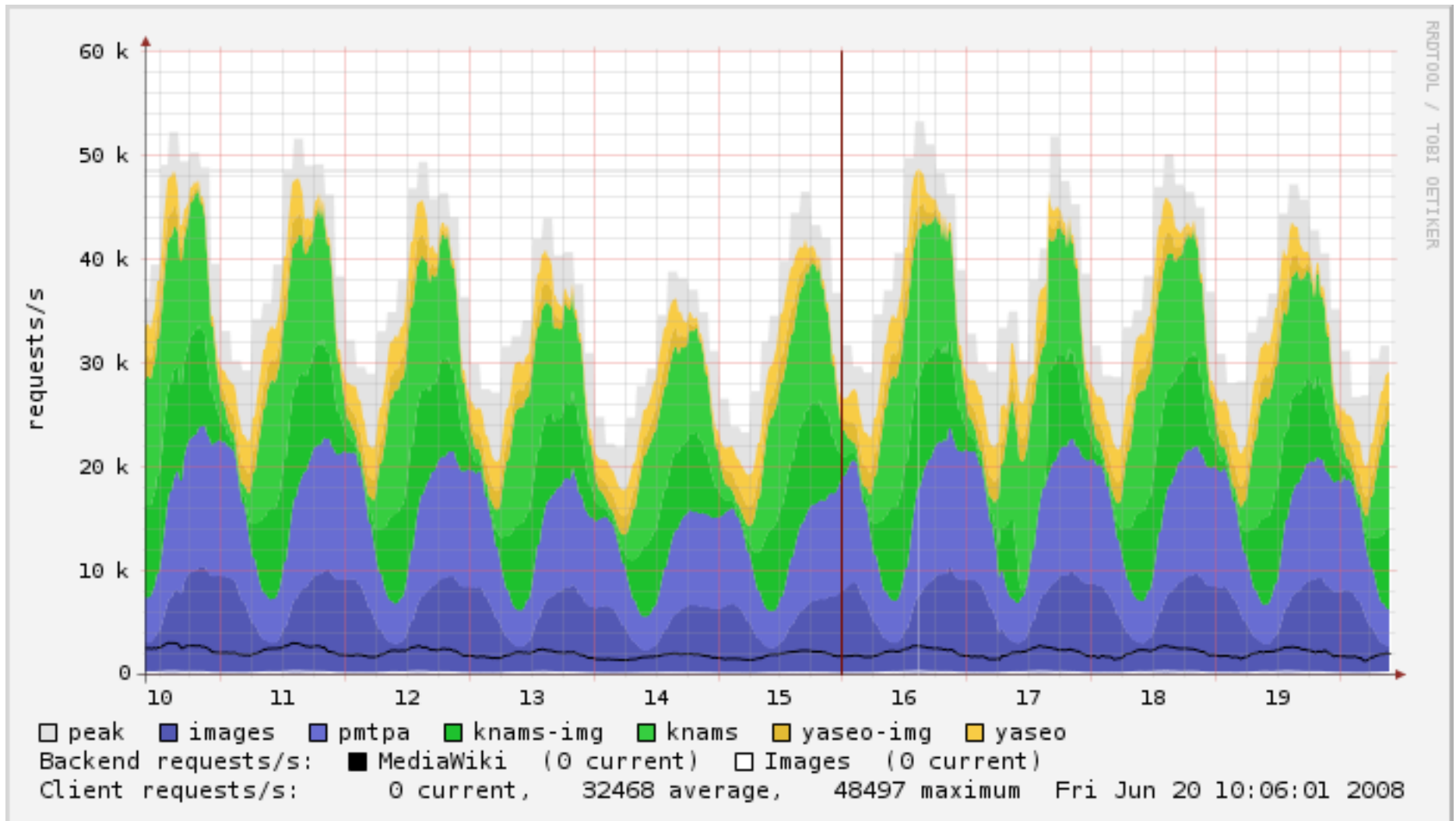
- Open
- Entertaining
- Educating
- Free
- Relaxing

The project

- It is community project

- Technology is just a side issue

- Though important one too

- MMORPG. Not a web site.

Growth

Daily Reach (percent)
wikipedia.org

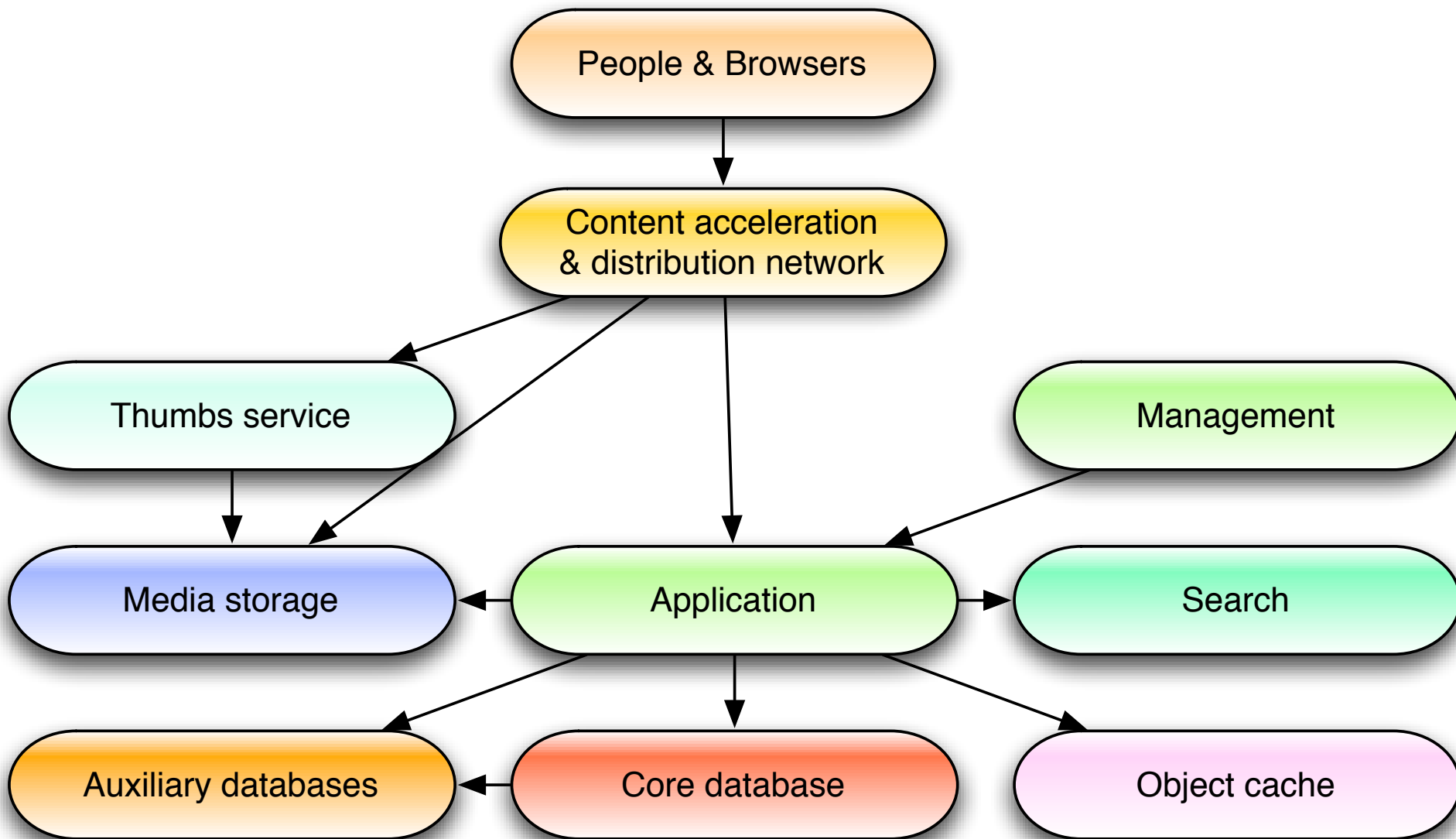©2007 Alexa

2007 Oct 23

Scale

MediaWiki

- Standard (Linux/Apache)/MySQL/PHP software

- Used by thousands of 3rd party sites

- Used by us

- Fits both

Though we...

- Built high-performance caching infrastructure in front

- Built high-performance data storage environment at the back end

- Went beyond Apache, PHP and MySQL

Big picture is big

Operation guidelines

- General availability (instead of high)

- Maximum efficiency (donations based)

- Be ready (drastically increasing loads)

- Understand the process and costs

- Community "owns" us

Availability

- Hardware does crash, but not very often (multiple year server uptime likely)

- As long as crash doesn't become a disaster, it is tolerable (nobody loses jobs)

- Losing few seconds of changes doesn't destroy business

- Downtime is (used to be) most profitable product

Efficiency

- All resources get used

- Solutions are simple

- Spending is limited

- Community would bash us otherwise

- Peer to peer Wikipedia would NOT be efficient. Really.

DON'T OVERLOAD

- Don't overload servers
- They have limited capacity
- We used to overload
- That was stressful
- Tuning for 1% doesn't work
- Go for 50%, 100%, 1000%
- millisecond at a time

Profile

- gprof
- oprofile
- xdebug
- KCacheGrind
- Profiler.php
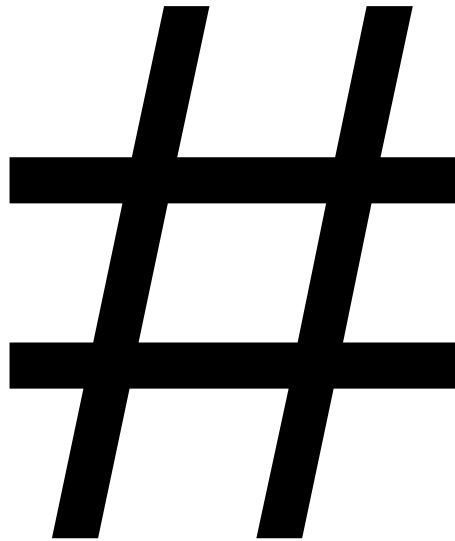- udpstats
- Google perftools
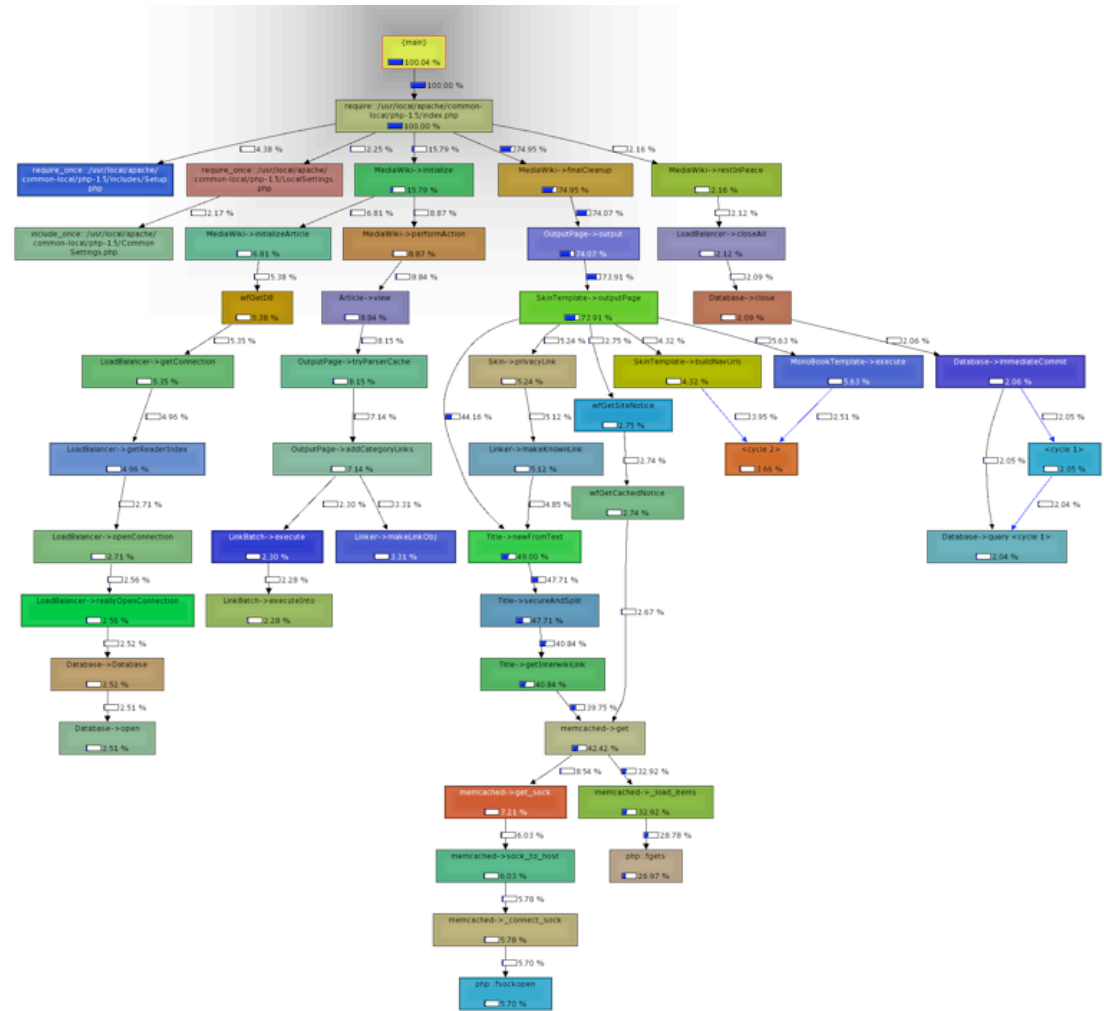
Optimization operator

//

For Perl/Python/Ruby

\#

Heavy tasks get

- Eliminated (see previous slide)

- Rewritten

- Delayed

- Simplified

- Reduced

Our data gets

- Split

- Reduced

- Compressed   •Used

- Aggregated

- Replicated

- Duplicated

Database splits

- Master got slaves
- Slaves got specialized by language
- Slaves forgot other languages
- Text storage offloaded to other instances
- Masters were split by language
- Special tasks got special slaves

MySQL!!!

- InnoDB stores, ~200-300GB instances

- 16GB boxes, upgrading to 32GB

- 6-8 disk RAID0, upgrading to 16 disk RAID10

- Replication (3-7 slaves per master)

- App-level load balancing & concurrency control

- Forked. Google patches and our minor improvements help

Lucene

- Handles full text search

- Is designed for that

- Flexible

- Java

Caching

- Features must allow caching

- Database is a cache too

- Evaluate the need for cache

- Invalidate when needed

Object cache

- Memcached - the networked cache

- Difficult to compute data

  - Parsed texts

  - Page differences

  - Compiled localization

- Temporary data - users, sessions, environment states

HTTP Cache

- Squid 2.6

- COSS

- Ruled by application, purge instead of evict

- Distributed: US, Europe, Asia

- Geo-backend for PowerDNS

- Multiple-layer (URL-hash routing)

Hardware

- Anyone needs old P4s?

- Quadcore chips are nice.

- Shutting down old hardware increases response times and saves hosting expenses

- RAID0 causes too much of maintenance work, high disk (2.5" SAS) density boxes to the rescue

Numbers

- >50000 HTTP requests per second

- >80000 SQL queries per second

- Somewhere in Alexa 10

- ~200 application servers, ~20 database servers, ~70 squid servers

- 80 2GB memcached instances, ~60 "stores"

- Up to ~5Gbps traffic

Data sizes

- For English Wikipedia:
  - 18m page objects
  - 250m pagelinks
  - 7m registered users
  - 220m revisions of text
  - 1.5TB of compressed revision storage
- Over a terabyte of media

Other...

- Image storage - two boxes with bunch of SATA disks and lighttpd

- LVS - efficient load balancing

- Lots of small components, most of them tweaked

People

- Difficult to scale volunteers

- System becomes more scary

- Some volunteers are now on foundation's payroll

- ~7 site engineers, from US, Australia, Netherlands, UK, Germany and Lithuania

Future

- All of you are the most important part of it

Questions and contacts

- Ask now, please!

- domas at sun dot com

- domas at #wikimedia-tech at freenode

- http://dammit.lt/

- http://dammit.lt/uc/ - detailed description