JavaEye网站架构解密

Robbin Fan

内容提要

- JavaEye网站的架构进化
- JavaEye网站的缓存
- JavaEye网站的全文检索
- 几个实战经验总结

JavaEye的网站流量?

JavaEye的硬件配置

- 两台IU的服务器
- 1台服务器是Web Server, 1台是DB Server
- 购买总价格是2.6万人民币
- 已经稳定运行将近3年

Web Server

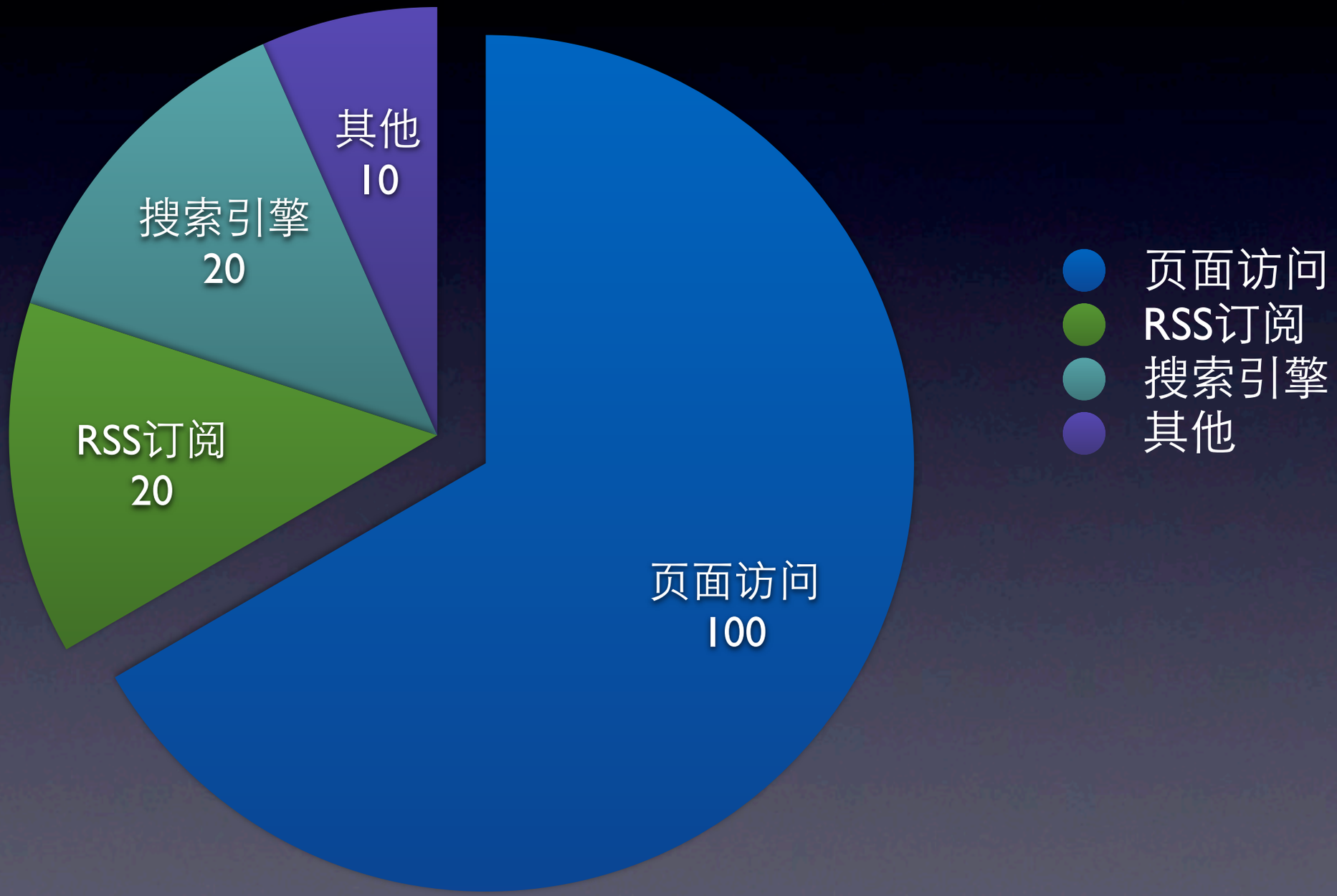
- AMD Opteron 2.4GHz单核 * 2 颗
- 8G内存
- 146G SCSI硬盘

DB Server

- AMD Opteron 2.0GHz单核 * 2 颗
- 4G内存
- 73G SCSI硬盘

150万动态请求/天

JavaEye网站访问来源比例



封杀列表

```
  $HTTP["useragent"] =~ "^PHP|^Java|sogou|galbot|RedBot|Shelob|AideRSS|Elanso Check Url|msnbot|WebCopier|OOZBO
T|HTTPRetriever|proximic|Steeler|heritrix|PageNest|Gaisbot|DotBot|LiveCategory|Teleport|psbot|Spinn3r|Indy|Yan
dex|A1 Website Download|Twiceler|Yahoo|yahoo|WebZIP|Nettraffic Sensor|SiteStatusChecker|VoilaBot|Ask Jeeves|Fr
eeWebMonitoring|Plonebot|Ginxbot|Cogentbot|EuripBot|PycURL|BigBellyBot|HttpComponents|Vidibot|Ebus-RSN|ichiro|
FeedTools|Funnel Web Profiler|Inar_spider|Commons-HttpClient|Wget|wget|larbin|speedy_spider|Yeti|voyager|HTMLP
arser|Yanga|CCBot|ApacheBench|CazoodleBot|Collapsar|KaloogaBot|qihoobot|HTTrack|Gigabot|perl|Ruby|ruby|python|
Python|365BlogLink|Blogbus|Nutch|MJ12bot|Tasapspider|Webdup|NextGenSearchBot|runnk\.com|LargeSmall Crawler|hl_
ftien_spider|R6_CommentReader|BlogBackupr|BBReaderFeedSpider|Isidorus|R6_FeedFetcher|RSSMicro\.com|Zwl\.Rss\.S
pider|JSpider|SEOENGBot|Gaisboti|PKU Student Spider|ia_archiver|robotgenius|hitcrawler" {
    url.rewrite = ( "^/(.*)" => "/crawler.html" )
  }
```

```
iptables -A INPUT -i eth0 -j DROP -p tcp --dport 80 -s 60.29.123.0/24   # spam
iptables -A INPUT -i eth0 -j DROP -p tcp --dport 80 -s 60.191.222.0/24  # spam
iptables -A INPUT -i eth0 -j DROP -p tcp --dport 80 -s 61.129.70.0/24   # 51jb.net
iptables -A INPUT -i eth0 -j DROP -p tcp --dport 80 -s 61.135.155.0/24  # it168
iptables -A INPUT -i eth0 -j DROP -p tcp --dport 80 -s 61.135.168.0/24  # it168
iptables -A INPUT -i eth0 -j DROP -p tcp --dport 80 -s 61.135.220.0/24  # it168
iptables -A INPUT -i eth0 -j DROP -p tcp --dport 80 -s 61.135.221.0/24  # it168
iptables -A INPUT -i eth0 -j DROP -p tcp --dport 80 -s 61.135.249.0/24  # youdao bot
iptables -A INPUT -i eth0 -j DROP -p tcp --dport 80 -s 61.142.250.0/24  # itpub
iptables -A INPUT -i eth0 -j DROP -p tcp --dport 80 -s 61.188.38.0/24   # spam
iptables -A INPUT -i eth0 -j DROP -p tcp --dport 80 -s 61.237.237.0/24  # spam
iptables -A INPUT -i eth0 -j DROP -p tcp --dport 80 -s 66.171.252.0/24  # spam
iptables -A INPUT -i eth0 -j DROP -p tcp --dport 80 -s 91.207.4.0/24    # spam
iptables -A INPUT -i eth0 -j DROP -p tcp --dport 80 -s 118.102.29.0/24  # spam
iptables -A INPUT -i eth0 -j DROP -p tcp --dport 80 -s 118.249.114.0/24 # spam
iptables -A INPUT -i eth0 -j DROP -p tcp --dport 80 -s 119.120.94.0/24  # spam
iptables -A INPUT -i eth0 -j DROP -p tcp --dport 80 -s 122.78.236.0/24  # FDM spam
iptables -A INPUT -i eth0 -j DROP -p tcp --dport 80 -s 122.96.159.0/24  # jiangsu netcom spam
iptables -A INPUT -i eth0 -j DROP -p tcp --dport 80 -s 124.115.0.0/24   # Xi'an IDC
iptables -A INPUT -i eth0 -j DROP -p tcp --dport 80 -s 124.115.4.0/24   # Xi'an IDC
iptables -A INPUT -i eth0 -j DROP -p tcp --dport 80 -s 124.238.254.0/24 # Heibei IDC
iptables -A INPUT -i eth0 -j DROP -p tcp --dport 80 -s 159.226.20.0/24  # zhongkeyuan spam
iptables -A INPUT -i eth0 -j DROP -p tcp --dport 80 -s 159.226.58.0/24  # zhongkeyuan spam
iptables -A INPUT -i eth0 -j DROP -p tcp --dport 80 -s 202.106.88.0/24  # Beijing Netcom spam
```

听说ruby很慢.....

Google Adplanner Data for JavaEye

Site profile javaeye.com

Thumbnail

Categories

Computers & Electronics > Programming
Computers & Electronics > Software

Advertising accepted

✓ Yes

Description

No description was found for javaeye.com

Publishers - click here to edit your site info.

View data for: China

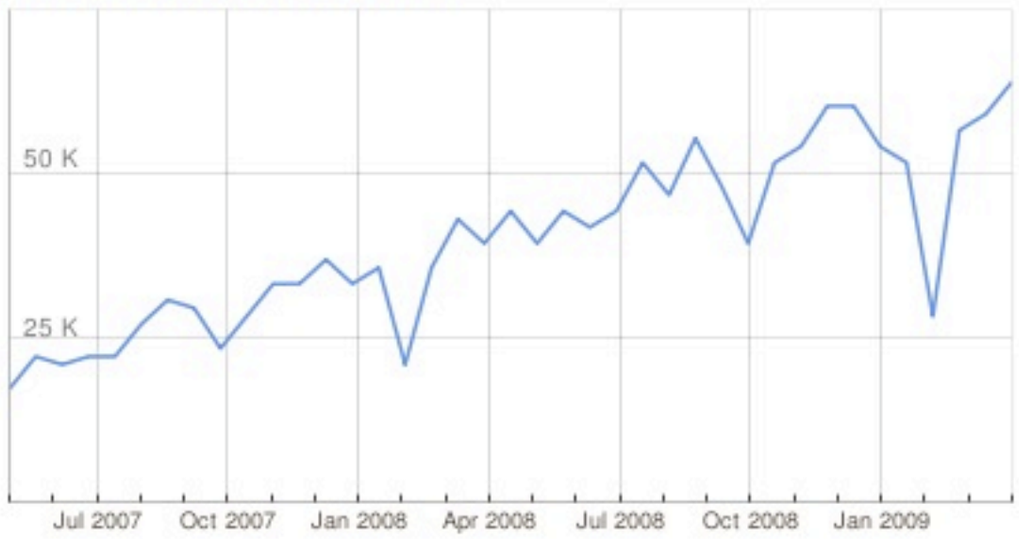
Data: Mar 2009

Traffic statistics

All traffic statistics are estimates.

	Country	Worldwide
Unique visitors (estimated cookies)	3.8 M	3.8 M
Unique visitors (users)	760 K	910 K
Reach	0.4%	0.1%
Page views	67 M	74 M
Total visits	13 M	13 M
Avg visits per visitor	17	14
Avg time on site	10:30	11:00

Daily Unique Visitors (users)



Google Adplanner Data for JavaEye

Site profile javaeye.com

Thumbnail

Categories

Computers & Electronics > Programming
Computers & Electronics > Software

Advertising accepted

✓ Yes

Description

No description was found for javaeye.com

Publishers - click here to edit your site info.

View data for: China

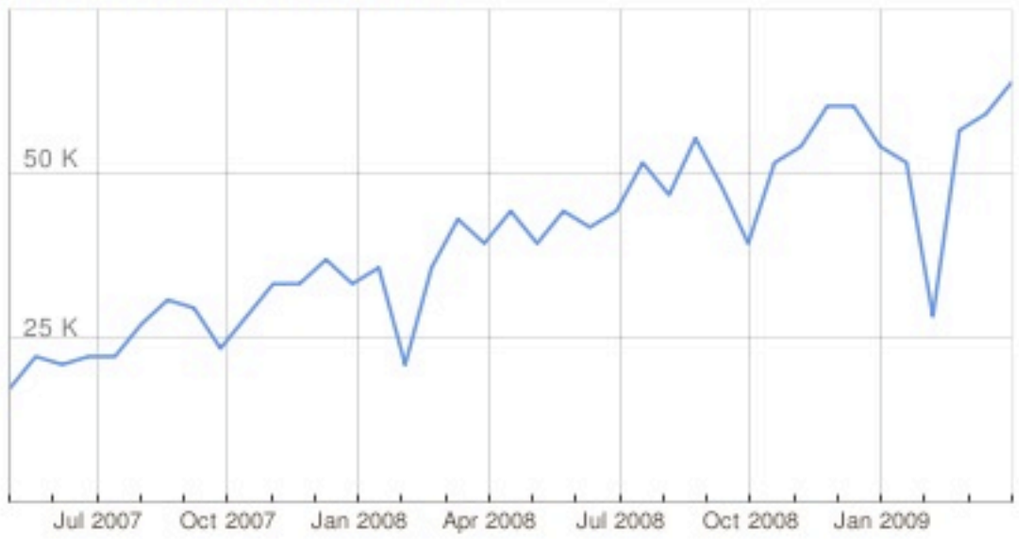
Data: Mar 2009

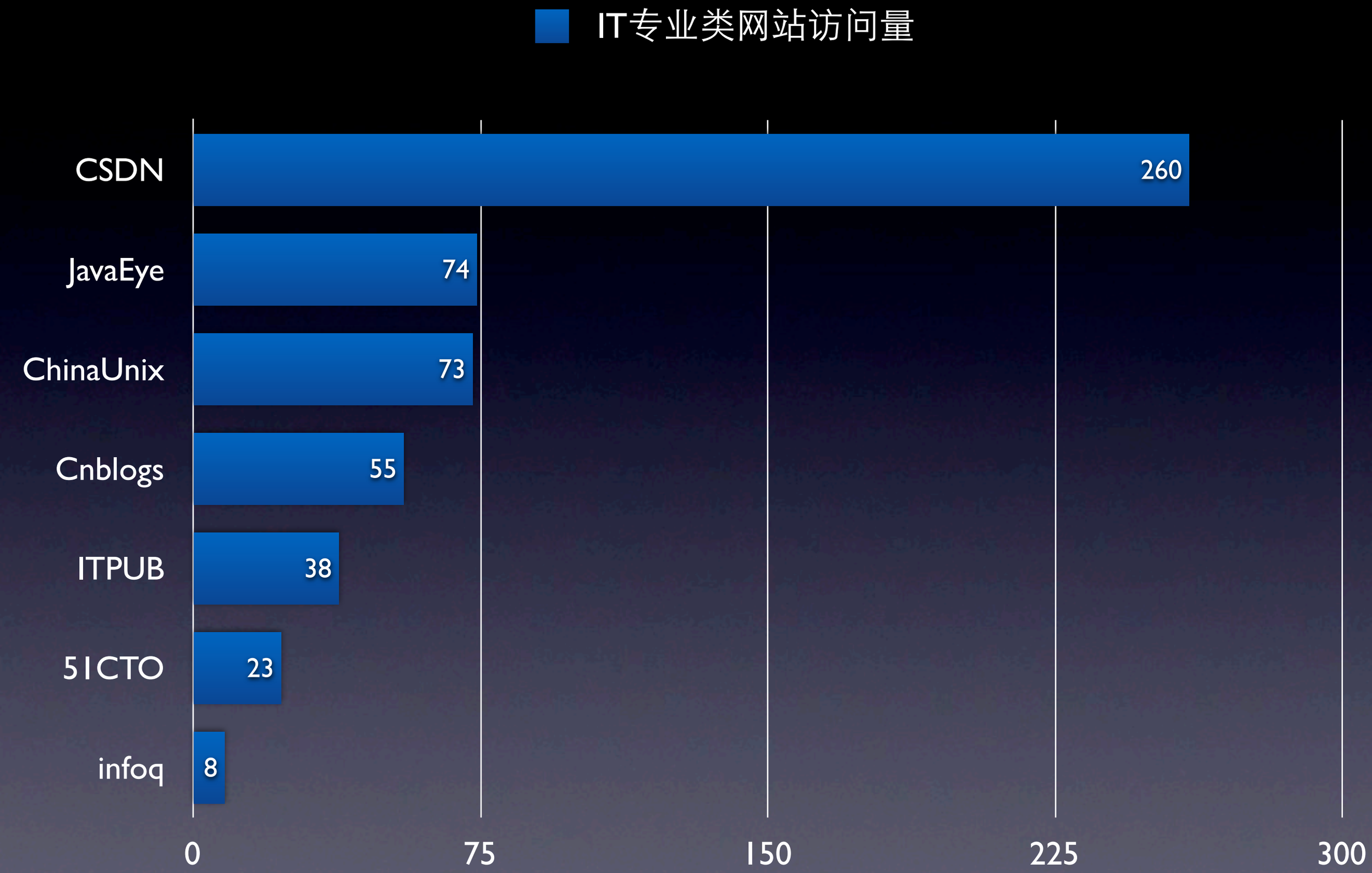
Traffic statistics

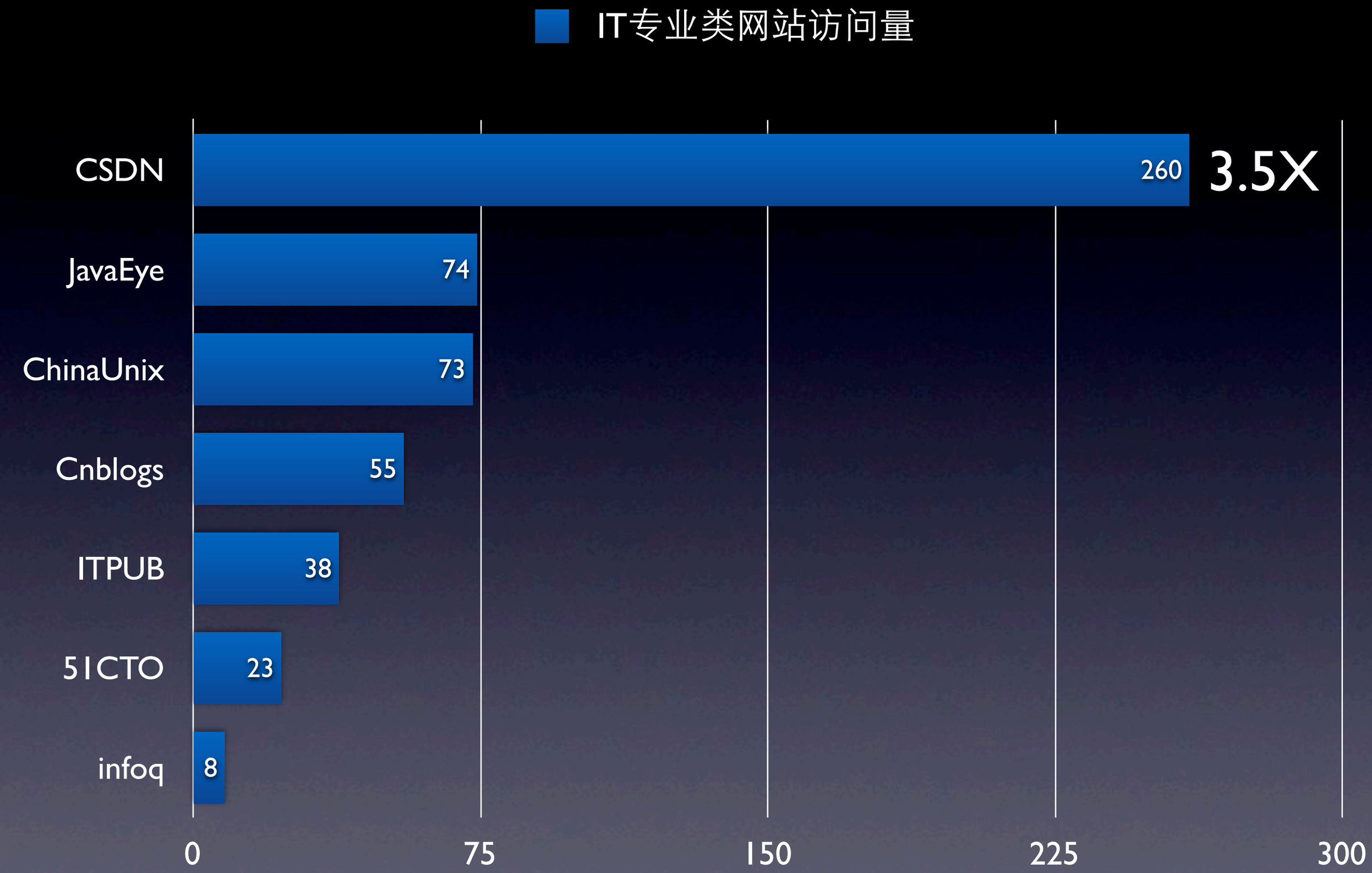
All traffic statistics are estimates.

	Country	Worldwide
Unique visitors (estimated cookies)	3.8 M	3.8 M
Unique visitors (users)	760 K	910 K
Reach	0.4%	0.1%
Page views	67 M	74 M
Total visits	13 M	13 M
Avg visits per visitor	17	14
Avg time on site	10:30	11:00

Daily Unique Visitors (users)




IT专业类网站访问量
CSDN
260
JavaEye
74
ChinaUnix
73
Cnblogs
55
ITPUB
38
51CTO
23
infoq
8
0
75
150
225
300


IT专业类网站访问量
CSDN
260
3.5X
JavaEye
74
ChinaUnix
73
Cnblogs
55
ITPUB
38
51CTO
23
infoq
8
0
75
150
225
300

IT专业类网站访问量

CSDN 30+ Server 260 3.5X

JavaEye 74

ChinaUnix 73

Cnblogs 55

ITPUB 38

51CTO 23

infoq 8

0 75 150 225 300

IT专业类网站访问量

CSDN 260

30+ Server ASP.net

3.5X

JavaEye 74

ChinaUnix 73

Cnblogs 55

ITPUB 38

51CTO 23

infoq 8

0 75 150 225 300

IT专业类网站访问量

CSDN 260 30+ Server ASP.net 3.5X

JavaEye 74 2 Server

ChinaUnix 73

Cnblogs 55

ITPUB 38

51CTO 23

infoq 8

0 75 150 225 300

IT专业类网站访问量

CSDN 260

JavaEye 74

ChinaUnix 73

Cnblogs 55

ITPUB 38

51CTO 23

infoq 8

0 75 150 225 300

30+ Server ASP.net 3.5X

2 Server ruby

IT专业类网站访问量

CSDN 260 30+ Server ASP.net 3.5X

JavaEye 74 2 Server ruby

ChinaUnix 73 4+ Server

Cnblogs 55

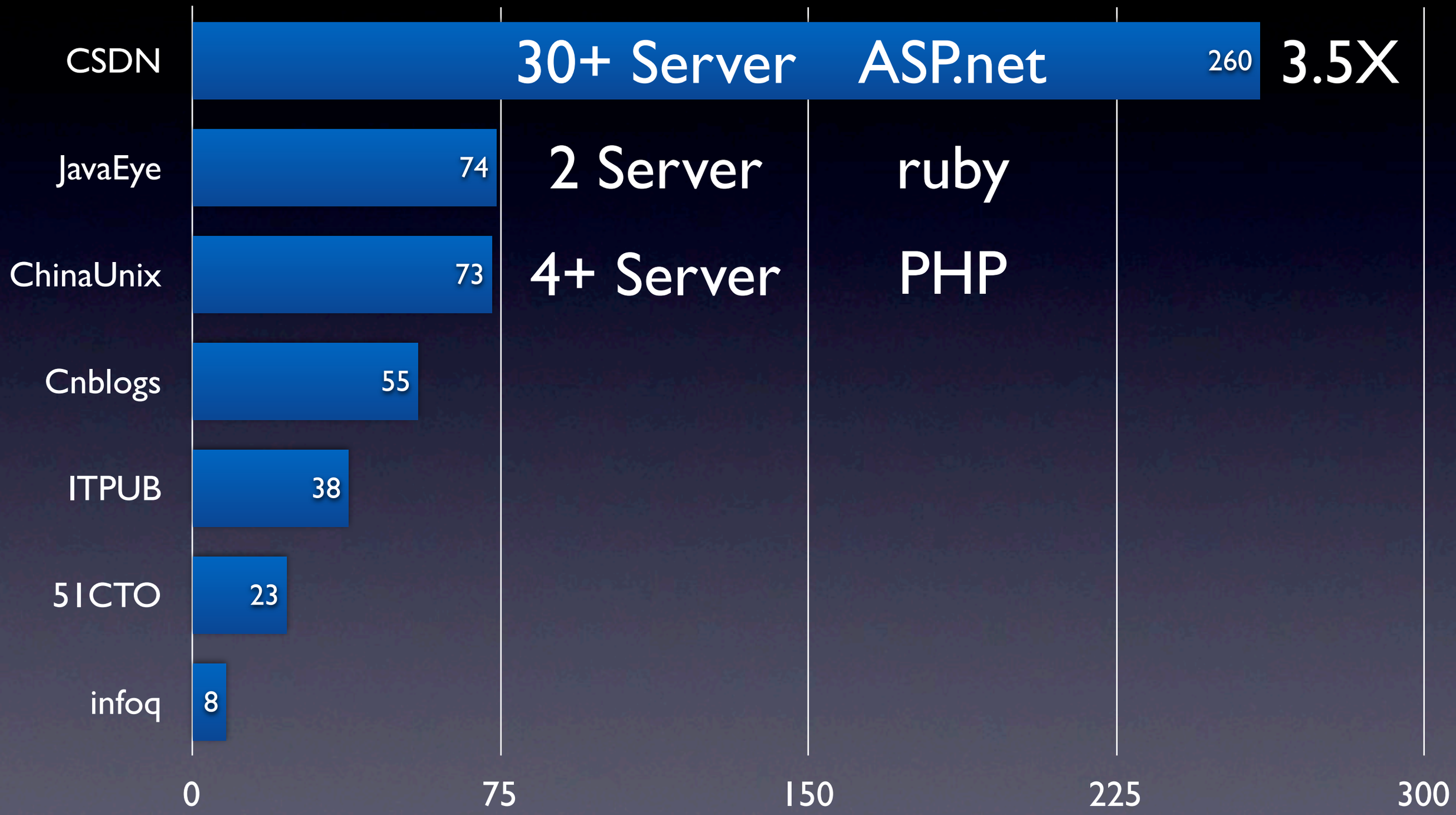
ITPUB 38

51CTO 23

infoq 8

0 75 150 225 300

IT专业类网站访问量


CSDN
30+ Server
ASP.net
260
3.5X
JavaEye
74
2 Server
ruby
ChinaUnix
73
4+ Server
PHP
Cnblogs
55
ITPUB
38
51CTO
23
infoq
8
0
75
150
225
300

IT专业类网站访问量

CSDN	260	30+ Server	ASP.net
JavaEye	74	2 Server	ruby
ChinaUnix	73	4+ Server	PHP
Cnblogs	55	4+ Server	
ITPUB	38		
51CTO	23		
infoq	8		

3.5X

0 75 150 225 300

IT专业类网站访问量

CSDN	260	30+ Server	ASP.net
JavaEye	74	2 Server	ruby
ChinaUnix	73	4+ Server	PHP
Cnblogs	55	4+ Server	ASP.net
ITPUB	38		
51CTO	23		
infoq	8		

3.5X

0 75 150 225 300

IT专业类网站访问量

CSDN	260	30+ Server	ASP.net	3.5X
JavaEye	74	2 Server	ruby	
ChinaUnix	73	4+ Server	PHP	
Cnblogs	55	4+ Server	ASP.net	
ITPUB	38		PHP	
51CTO	23			
infoq	8			

0 75 150 225 300

IT专业类网站访问量

CSDN	260	30+ Server	ASP.net	3.5X
JavaEye	74	2 Server	ruby	
ChinaUnix	73	4+ Server	PHP	
Cnblogs	55	4+ Server	ASP.net	
ITPUB	38		PHP	
51CTO	23			
infoq	8		Java	

0 75 150 225 300

JavaEye网站架构进化

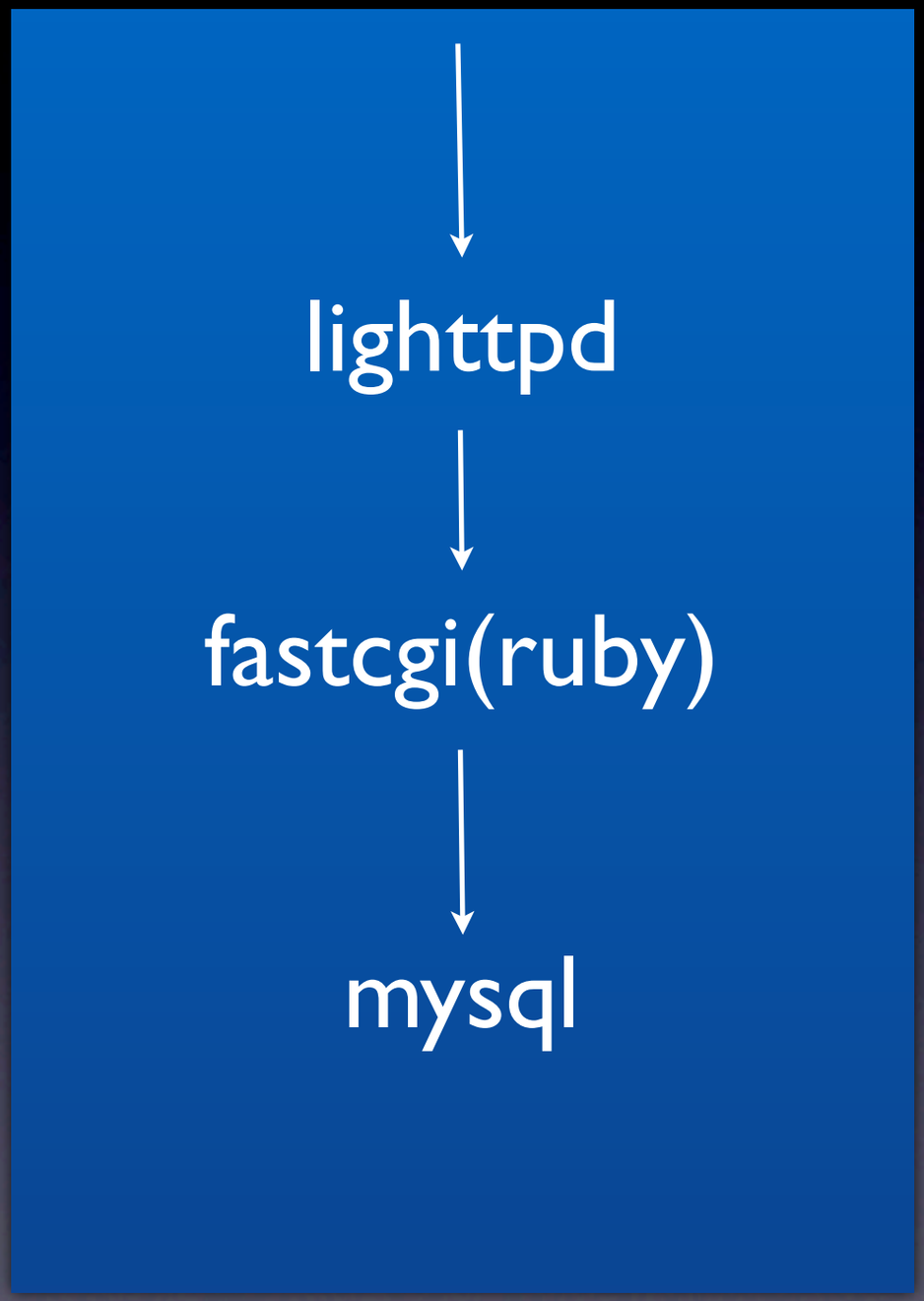
2006.09

- 1台服务器
- 没有缓存
- 没有全文检索

2006.09

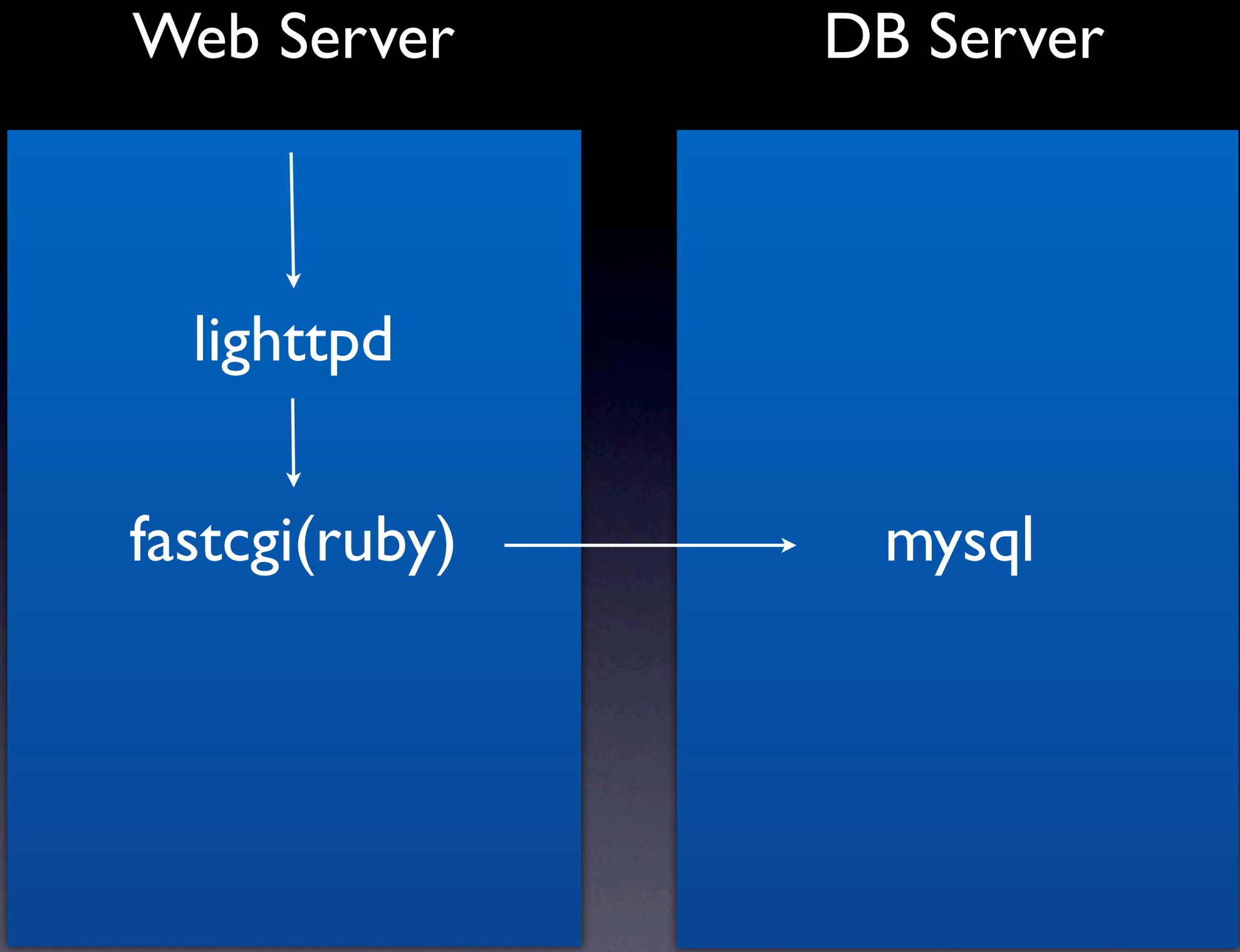
- lighttpd
- ruby 1.8.4, rails 1.1.2, 以fastcgi方式运行
- mysql5.0

Single Server



2007.01

- 添加了第2台服务器
- 把web和DB分开
- 系统瓶颈在数据库IO端


Web Server
DB Server
lighttpd
fastcgi(ruby)
mysql

2007.02

- 把posts表的大字段剥离出来
- posts表的select count操作从30秒减少到0.1秒

posts表

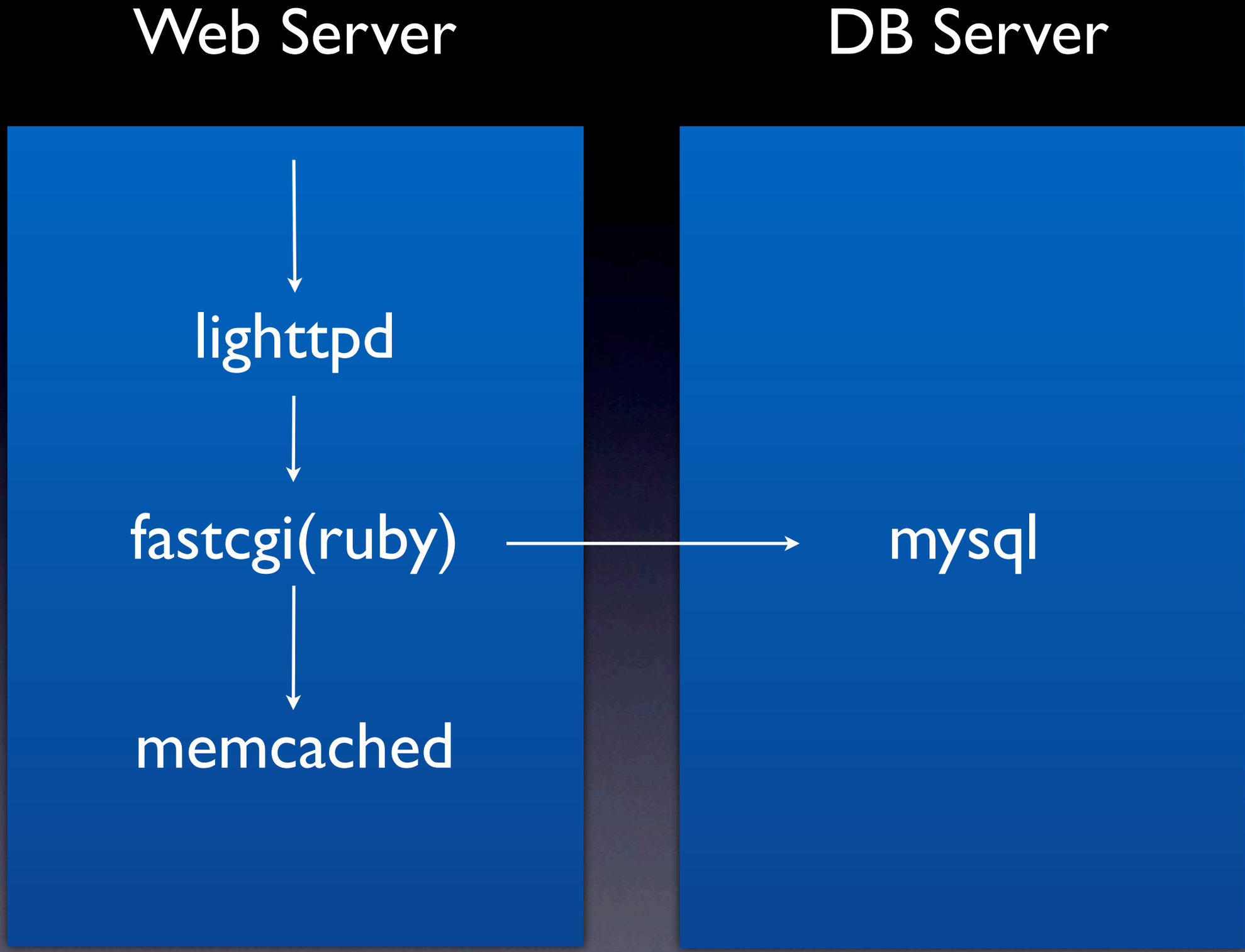
- posts(id, ..., body)
- 磁盘存储空间2GB

剥离后posts表

- posts(id, post_text_id,...) 50MB
- post_texts(id, body) 2GB

2007.03

- 数据库瓶颈仍然存在
- 引入memcached和CachedModel
- 自己编写了简单的查询缓存
- 240 sql query/s 下降到 140 sql query/s
- memcached缓存命中率在75%


Web Server
DB Server
lighttpd
fastcgi(ruby)
memcached
mysql

2007.05

- 数据库调优
- 优化数据库索引

2007.09

- 引入全文检索
- 使用ruby的ferret
- 中文分词使用单字拆分法

2008.01

- JavaEye网站代码重写
- 缓存框架改用cache_fu
- 缓存命中率上升到84%
- sql query下降到 50条/s

2008.05

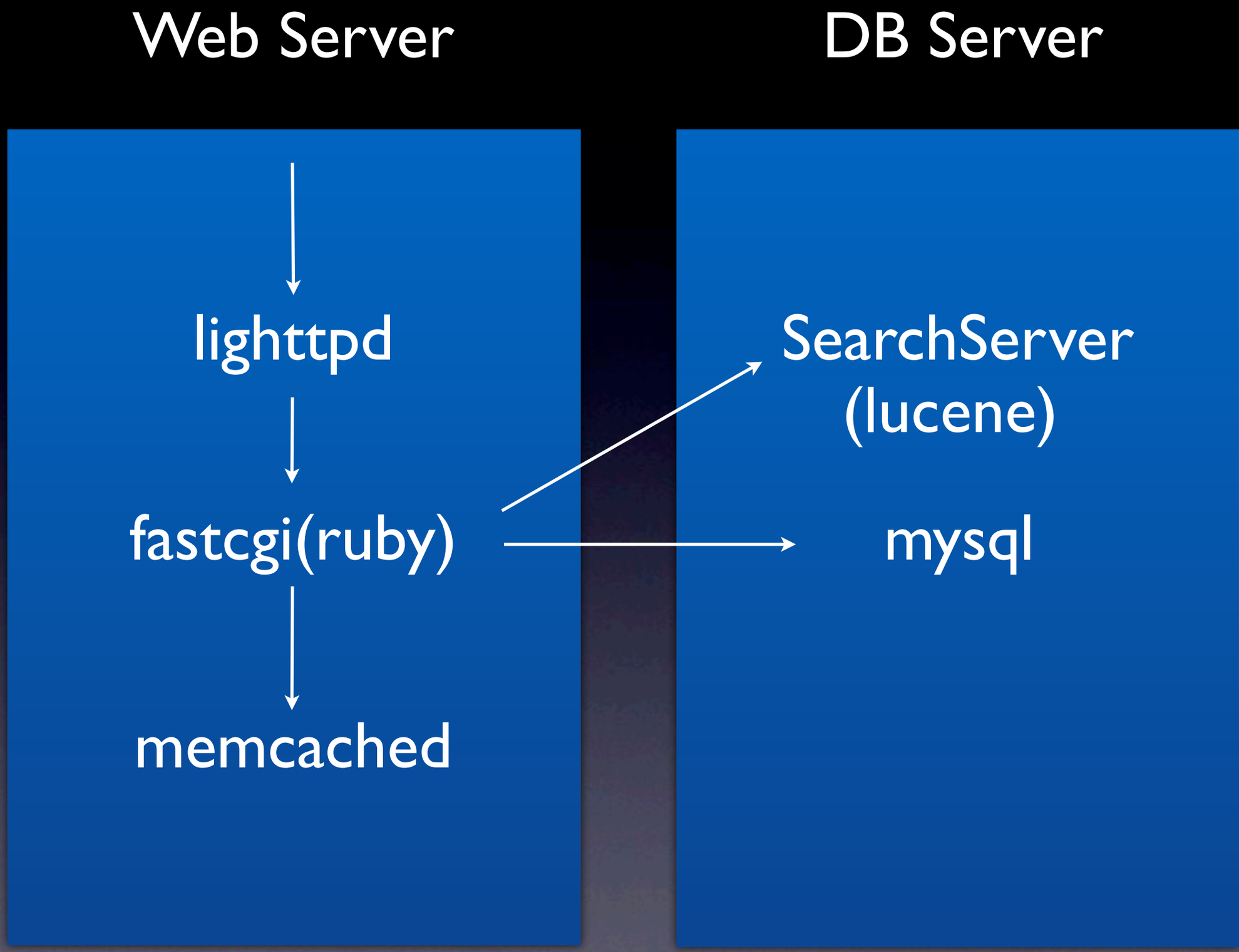
- 中文分词算法改用rmmseg-cpp

2008.10

- 自制山寨cache plugin
- 缓存命中率上升到96%以上

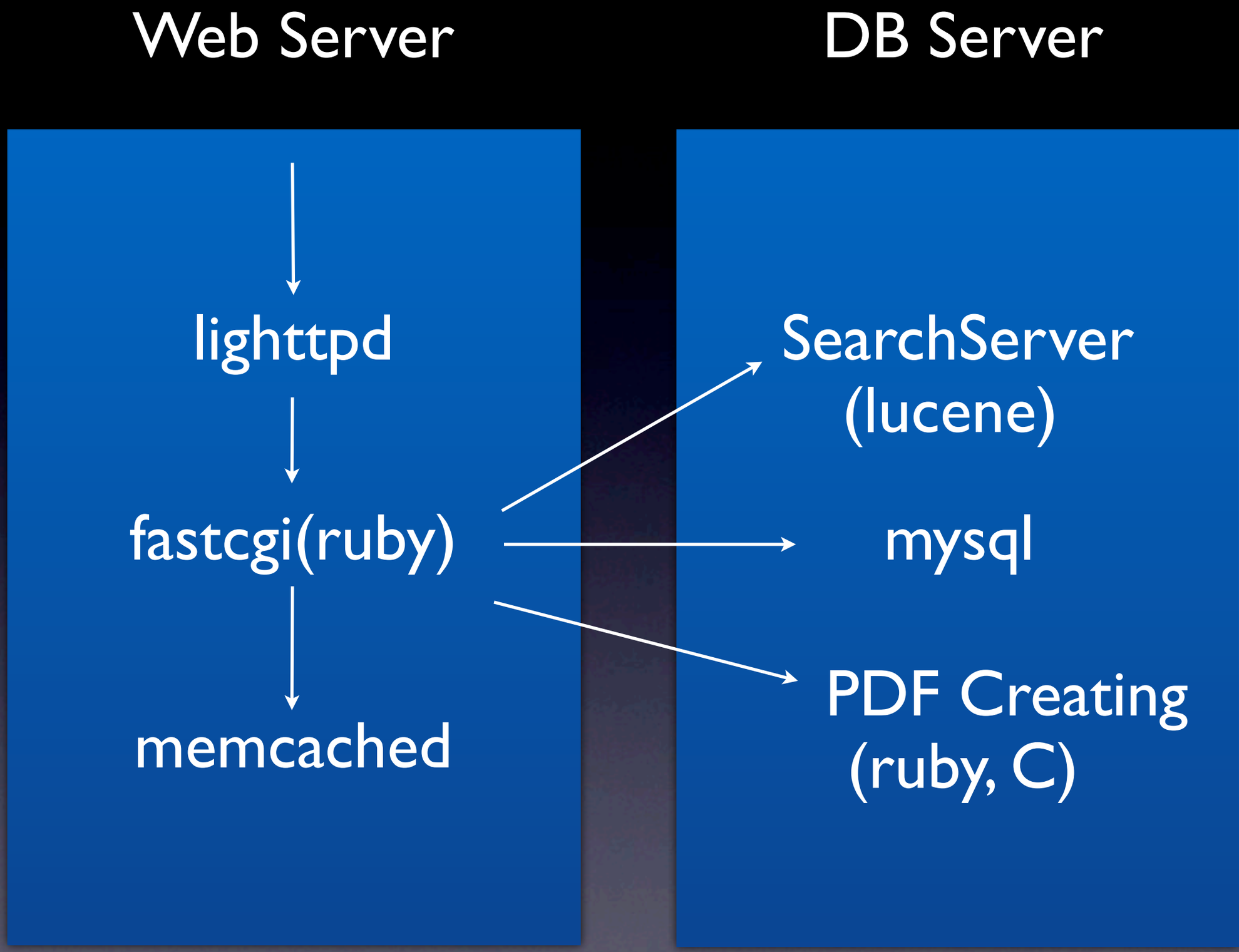
2008.10

- 抛弃ferret，自己编写全文检索服务器
- 使用Java的lucene作为全文检索引擎
- 自己实现C/S架构的内部调用


Web Server
DB Server
lighttpd
fastcgi(ruby)
memcached
SearchServer
(lucene)
mysql

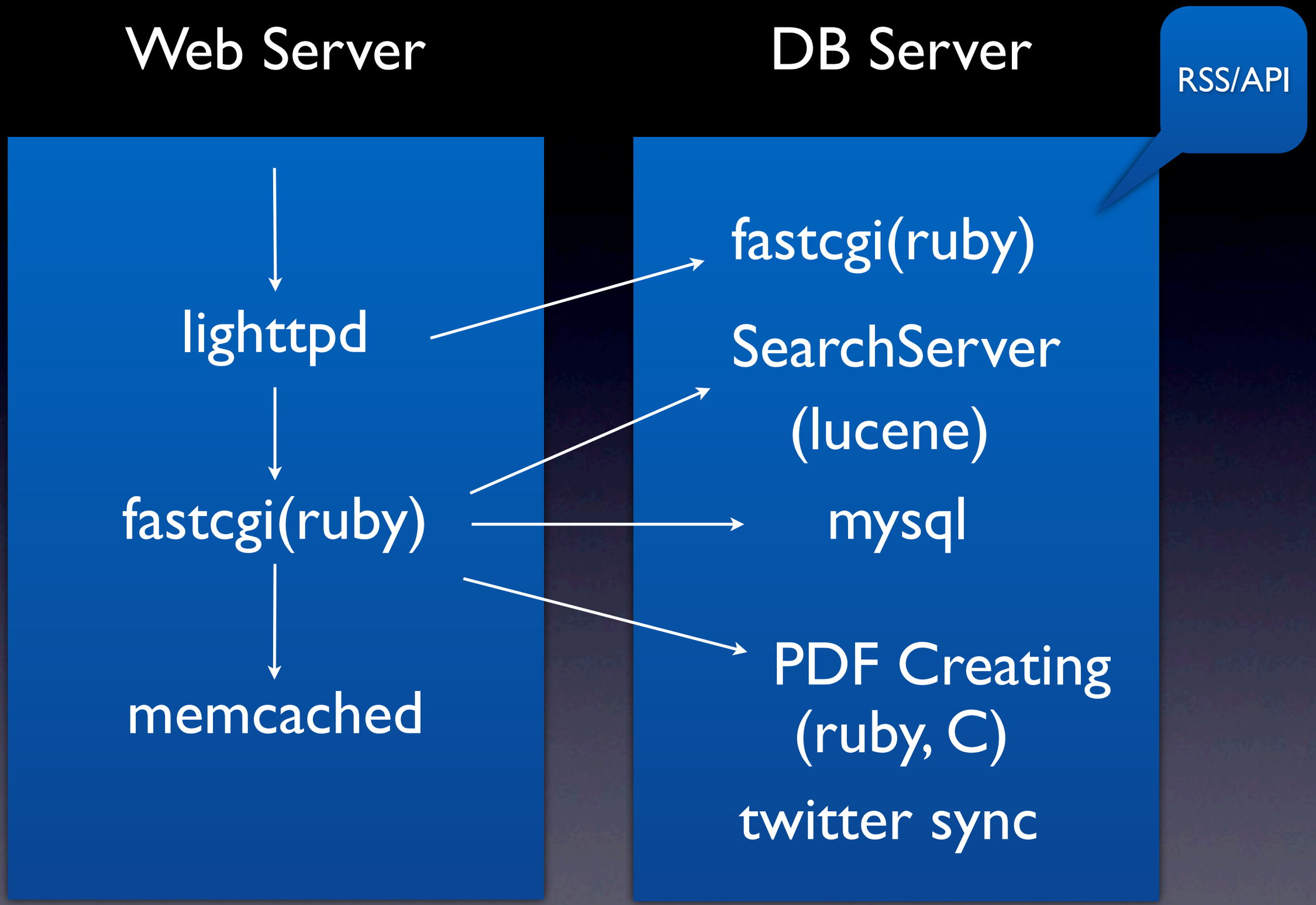
2008.11

- 实现博客，新闻制作PDF功能


Web Server
DB Server
lighttpd
fastcgi(ruby)
memcached
SearchServer
(lucene)
mysql
PDF Creating
(ruby, C)

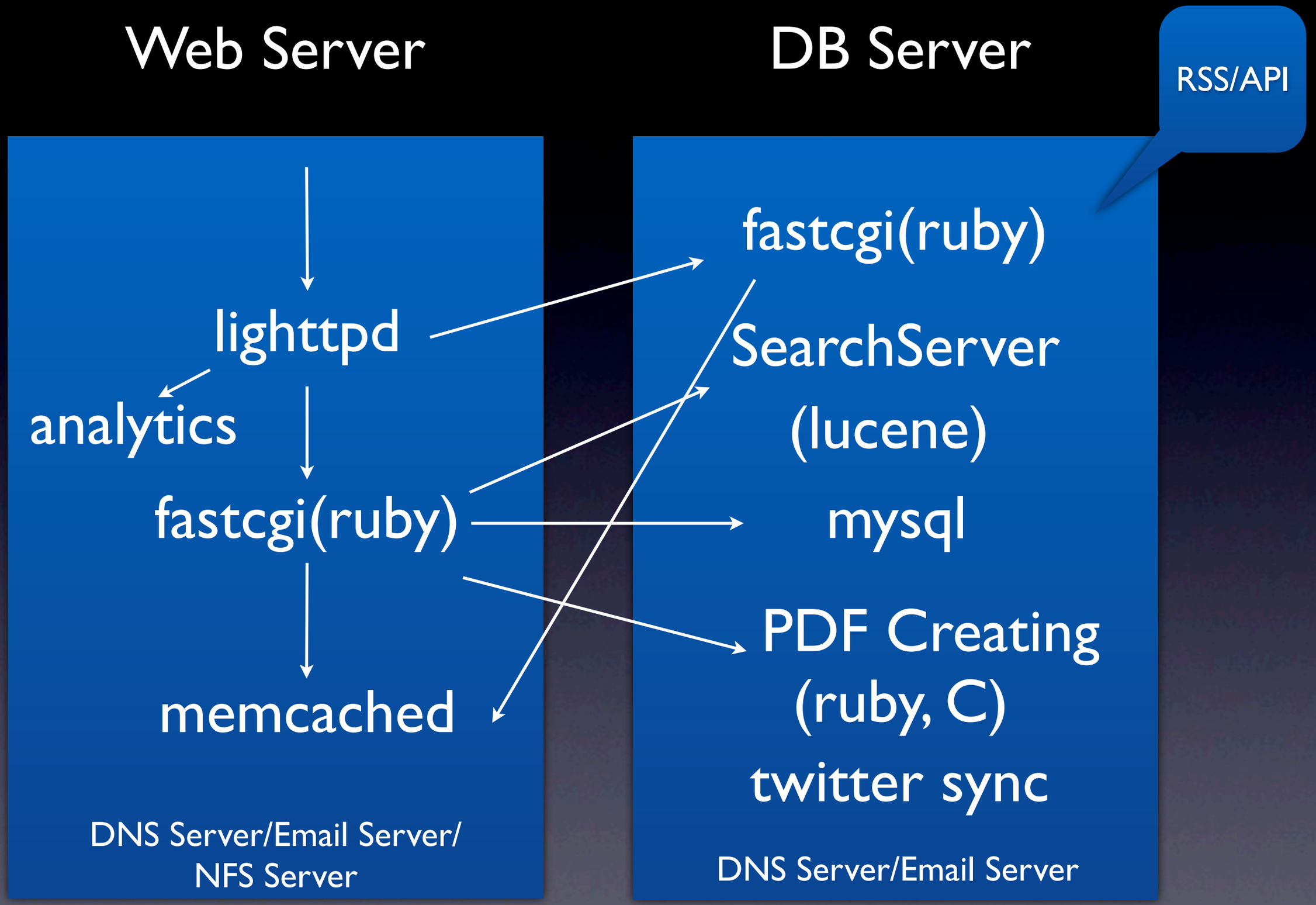
2009.03

- SNS feed功能
- twitter绑定功能
- 开放API


Web Server
DB Server
RSS/API
lighttpd
fastcgi(ruby)
memcached
fastcgi(ruby)
SearchServer
(lucene)
mysql
PDF Creating
(ruby, C)
twitter sync

2009.03

- 废弃Google Analytics
- 自己编写简单的网站流量分析系统


Web Server
DB Server
RSS/API
lighttpd
analytics
fastcgi(ruby)
memcached
DNS Server/Email Server/
NFS Server
fastcgi(ruby)
SearchServer
(lucene)
mysql
PDF Creating
(ruby, C)
twitter sync
DNS Server/Email Server

2009.12

- 添加Web IM
- 添加一台服务器
- 合理规划服务器

Web Server

lighttpd

fastcgi(ruby)

memcached

Backgroud Server

fastcgi(ruby)

IM Server

PDF Creating

SearchServer

DB Server

mysql

JavaEye缓存进化之路

- CachedModel+山寨QueryCache
- cache_fu
- 山寨cache_plugin
- cache_money*

应用缓存概述

- 对象缓存
- 查询缓存
- 页面缓存
  - 动态页面静态化（page cache）
  - 页面片段缓存（frgment cache）
  - 基于REST资源的缓存

对象缓存原则

- 数据库表的设计要细颗粒度
- 把有冗余字段的大表拆分为n个互相外键关联的小表
- ORM的性能瓶颈不在于表关联，而在于大表的全表扫描
- 尽量避免join查询，多制造n+1条SQL

对象缓存的意义

- Web应用很容易通过群集方式实现横向扩展，系统的瓶颈往往出现在数据库
- 数据库的瓶颈往往出现在磁盘IO读写
- 因此要避免数据库的全表扫描和大表的数据扫描操作
- 如何避免：拆表和臭名昭著的n+1条SQL

名称	类型	记录	大小	上次
post_texts	InnoDB	~1,128,719	2,767 MB	???
user_sessions	MEMORY	17,897	276 MB	???
posts	InnoDB	~780,001	117 MB	???
topics	InnoDB	~297,017	96 MB	???
users	InnoDB	~301,521	82 MB	???
user_visits	InnoDB	~635,967	79 MB	???
messages	InnoDB	~243,889	65 MB	???
group_posts	InnoDB	~89,367	50 MB	???
problems	InnoDB	~15,921	44 MB	???
activities	InnoDB	~152,476	29 MB	???
user_profiles	InnoDB	~285,189	28 MB	???
message_texts	InnoDB	~44,890	28 MB	???
user_favorites	InnoDB	~157,897	26 MB	???
attachments	InnoDB	~82,552	24 MB	???
solutions	InnoDB	~37,054	20 MB	???
news	InnoDB	~7,612	20 MB	???
cards	InnoDB	~5,230	20 MB	???
taggings	InnoDB	~141,427	18 MB	???
auto_saves	InnoDB	~2,335	18 MB	???
resumes	InnoDB	~7,214	14 MB	???
article_texts	InnoDB	~3,719	10 MB	???

名称	类型	记录	大小	上次
post_texts	InnoDB	~1,128,719	2,767 MB	???
user_sessions	MEMORY	17,897	276 MB	???
posts	InnoDB	~780,001	117 MB	???
topics	InnoDB	~297,017	96 MB	???
users	InnoDB	~301,521	82 MB	???
user_visits	InnoDB	~635,967	79 MB	???
messages	InnoDB	~243,889	65 MB	???
group_posts	InnoDB	~89,367	50 MB	???
problems	InnoDB	~15,921	44 MB	???
activities	InnoDB	~152,476	29 MB	???
user_profiles	InnoDB	~285,189	28 MB	???
message_texts	InnoDB	~44,890	28 MB	???
user_favorites	InnoDB	~157,897	26 MB	???
attachments	InnoDB	~82,552	24 MB	???
solutions	InnoDB	~37,054	20 MB	???
news	InnoDB	~7,612	20 MB	???
cards	InnoDB	~5,230	20 MB	???
taggings	InnoDB	~141,427	18 MB	???
auto_saves	InnoDB	~2,335	18 MB	???
resumes	InnoDB	~7,214	14 MB	???
article_texts	InnoDB	~3,719	10 MB	???

CachedModel

- 对象缓存,和Hibernate二级缓存类似
- 透明化存取，无需特别的编码
- select * from post_texts where id = ?
- 达到了75%的缓存命中率

```
# == Schema Information
#
# Table name: post_texts
#
#  id   :integer(11)   not null, primary key
#  text :text
#
```

```
class PostText < CachedModel
  has_one :post
  def topic
    self.post.topic
  end
end
```

```
# == Schema Information
#
# Table name: post_texts
#
#  id   :integer(11)   not null, primary key
#  text :text
#

class PostText < CachedModel
  has_one :post
  def topic
    self.post.topic
  end
end
```

CachedModel缺点

- 拦截AR的find_by_sql，兼容性不好
- 没有给用户更多灵活的可控制性
- 作者自己已经很久不维护了

cache_fu

- cache_fu不对AR对象进行任何拦截，全部交给用户编程
- 用户有完全的控制权，但所有的缓存代码要自己手工编写
- 采用cache_fu，memcached缓存命中率达到84%

山寨cache plugin

- 基于Rails Cache的简单封装，仅60行代码
- 可以自动实现对象缓存的管理，n:1关系的缓存，但不支持1:n集合缓存
- memcached缓存命中率超过96%

memcached统计信息

Host Status Diagrams

Cache Usage

Free: 223.8 MBytes (10.9%)

Used: 1.8 GBytes (89.1%)

Hits & Misses

Hits: 1292978479 (95.8%)

Misses: 56835391 (4.2%)

Cache Information	
Current Items(total)	926794 (66927546)
Hits	1292978479
Misses	56835391
Request Rate (hits, misses)	350.52 cache requests/second
Hit Rate	335.76 cache requests/second
Miss Rate	14.76 cache requests/second
Set Rate	17.38 cache requests/second

memcached统计信息

Host Status Diagrams

Cache Usage

Free: 223.8 MBytes (10.9%)

Used: 1.8 GBytes (89.1%)

Hits & Misses

Hits: 1292978479 (95.8%)

Misses: 56835391 (4.2%)

Cache Information	
Current Items(total)	926794 (66927546)
Hits	1292978479
Misses	56835391
Request Rate (hits, misses)	350.52 cache requests/second
Hit Rate	335.76 cache requests/second
Miss Rate	14.76 cache requests/second
Set Rate	17.38 cache requests/second

cache_money

- 出自twitter开发团队之手
- 可能是目前最强大的ruby cache框架
- 支持分页查询缓存，支持条件查询缓存

页面片段缓存

- JavaEye大量使用页面片段缓存
  - 网站首页、新闻频道、个人博客左边导航条等等
  - 可以有效降低ruby应用的负载

JavaEye遇到的问题

- ruby负载远远超过db，Web Server的load是DB的两倍
- 单纯对象缓存不能减轻ruby的负载

ruby的性能问题

- ruby的正则表达式运算性能很糟糕
- erb的性能也不好
- 大量字符串运算性能差
- 解决之道：用片段缓存降低ruby字符串处理频率

不直接缓存post的内容，改为缓存post内容生成的html片段

```
def format_post_text(post, options = {})
  result = Post.get_cache("#{post.id}/body/#{post.modified_at.to_i}", :expires_in => 14.days) do
    post.bbcode? ? bbcodeize(post.body) : post.body
  end
  unless options.empty?
    result = Dryopteris.strip_tags(result) if options[:summary] || options[:abbrev]
    result = result.gsub(/\n|\r|<|>|"/, ' ') if options[:abbrev]
    result = cut_text(result, options[:len]) if options[:len]
  end
  return result
end
```

不直接缓存post的内容，改为缓存post内容生成的html片段

```
def format_post_text(post, options = {})
  result = Post.get_cache("#{post.id}/body/#{post.modified_at.to_i}", :expires_in => 14.days) do
    post.bbcode? ? bbcodeize(post.body) : post.body
  end
  unless options.empty?
    result = Dryopteris.strip_tags(result) if options[:summary] || options[:abbrev]
    result = result.gsub(/\n|\r|<|>|"/, ' ') if options[:abbrev]
    result = cut_text(result, options[:len]) if options[:len]
  end
  return result
end
```

JavaEye的缓存参考

- memcached缓存命中率96%
- cache get : sql query = 4 : 1

JavaEye全文检索方案

- ferret
- ferret+rmmseg-cpp中文分词
- 自制山寨全文检索服务

为何要自己开发?

- ferret+rmmseg在大数据量的时候内存占用很高
- ferret无法承担每天几十万次全文检索请求

fastcgi
(ruby)

json protocol

socket

SearchServer
(java)

long lived processes

long lived threads

如何实现多域名

- DNS服务器要做泛域名解析
- Web Server配置多域名支持
- ruby应用程序自己根据域名做相应的处理

```
$TTL 86400
@       IN      SOA     javaeye.com. webmaster.javaeye.com.
                                2009030400 ; Serial
                                28800 ;      Refresh
                                14400 ;      Retry
                                3600000 ;    Expire
                                86400 );     Default_ttl
        IN      NS      ns1
        IN      NS      ns2
        IN      MX 10   mail
        IN      A       221.130.189.113
ns1     IN      A       221.130.189.113
ns2     IN      A       221.130.189.120
mail    IN      A       221.130.189.113
www     IN      A       221.130.189.113
ftp     IN      A       221.130.189.113
forum   IN      A       221.130.189.113
job     IN      A       221.130.189.113
app     IN      A       221.130.189.113
stat    IN      A       221.130.189.113
video   IN      A       221.130.189.113
cvs     IN      A       61.152.175.100
*       IN      A       221.130.189.113
```

```
else $HTTP["host"] =~ "^([a-zA-Z0-9\-\.]+).javaeye.com$" {
  server.document-root = "/home/webuser/webroot/javaeye/current/public"
  server.errorfile-prefix = "/home/webuser/webroot/javaeye/current/public/error-"
```

```ruby
def homepage
  if www?
    render :template => 'main/homepage', :layout => false
  elsif app?
    if logged_in?
      render :template => 'app/base/index', :layout => 'app'
    else
      redirect_to homepage_url
    end
  elsif job?
    render :template => 'job/homepage', :layout => 'job'
  elsif channel?
    @channel = request.subdomains[0]
    if File.exists?("#{RAILS_ROOT}/app/views/channel/#{@channel}/index.rhtml")
      render :template => "channel/#{@channel}/index", :layout => 'channel'
    else
      render :file => "#{RAILS_ROOT}/public/404.html", :status => 404
    end
  elsif blog?
    @blogs = @blog_owner.blogs_by_page params[:page]
    render :template => 'blog/index/index', :layout => 'blog'
  elsif group?
    Group.increment_counter(:visit_count, @group.id)
    render :template => 'group/index/index', :layout => 'group'
  else
```

Thank you!