

pixiv의 인프라스트럭처

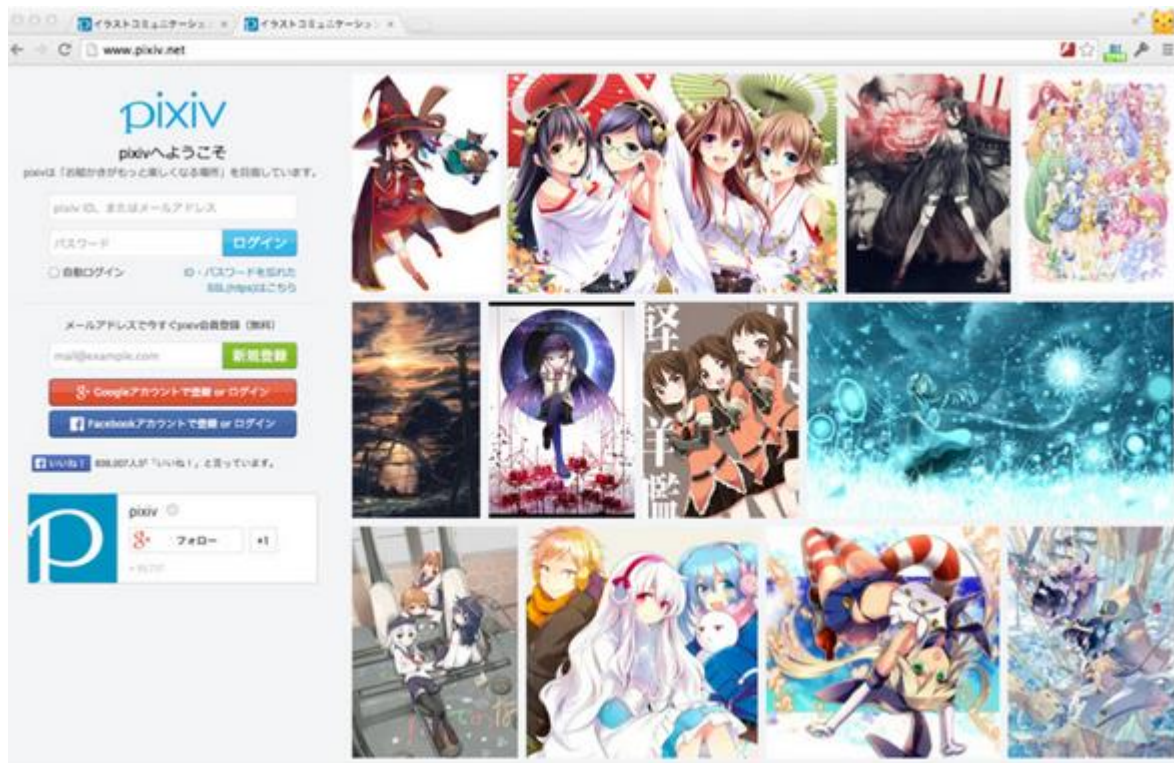
최흥배

<https://github.com/jacking75/choiHeungbae>

<https://speakerdeck.com/harukasan/the-technology-behind-pixiv-infrastructure>
(2013. 11.30)

글을 번역

pixiv?



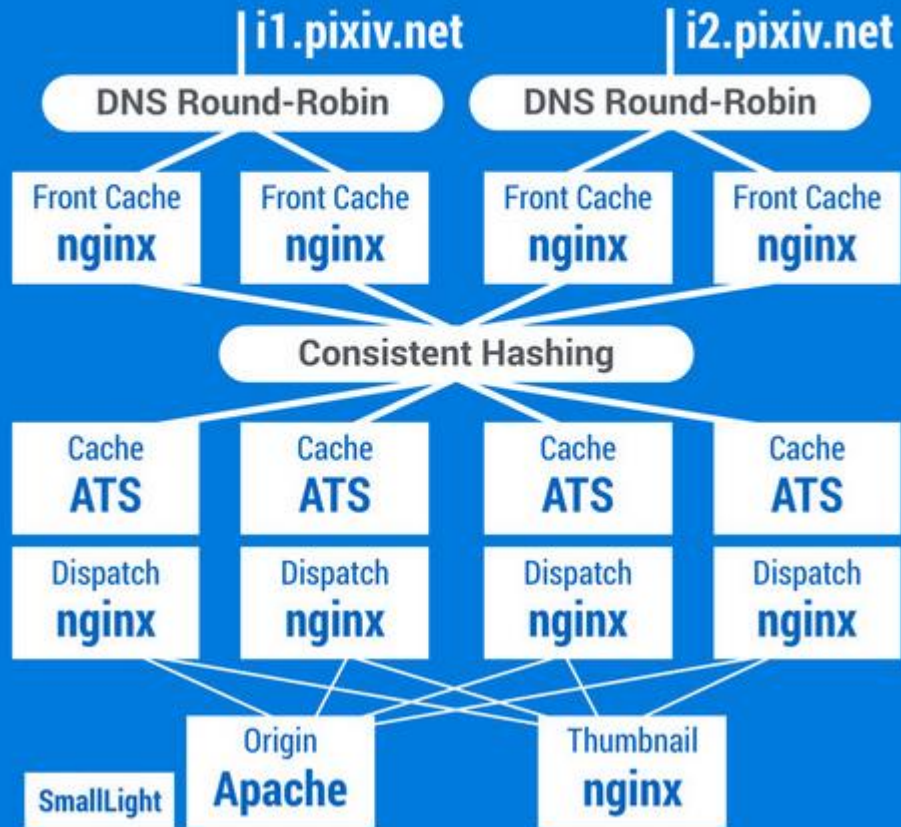
Monthly PV	3.7 Billion
Server	400+
Traffic	10Gbps+

팀원 6명

pixiv Image Cluster

- 이미지 전송 클러스터.
- 2010년부터 운용 개시.
- 메인 콘텐츠인 일러스트를 초고속으로 처리하기 위해 최적화.
- 모든 트랙픽의 90% 이상

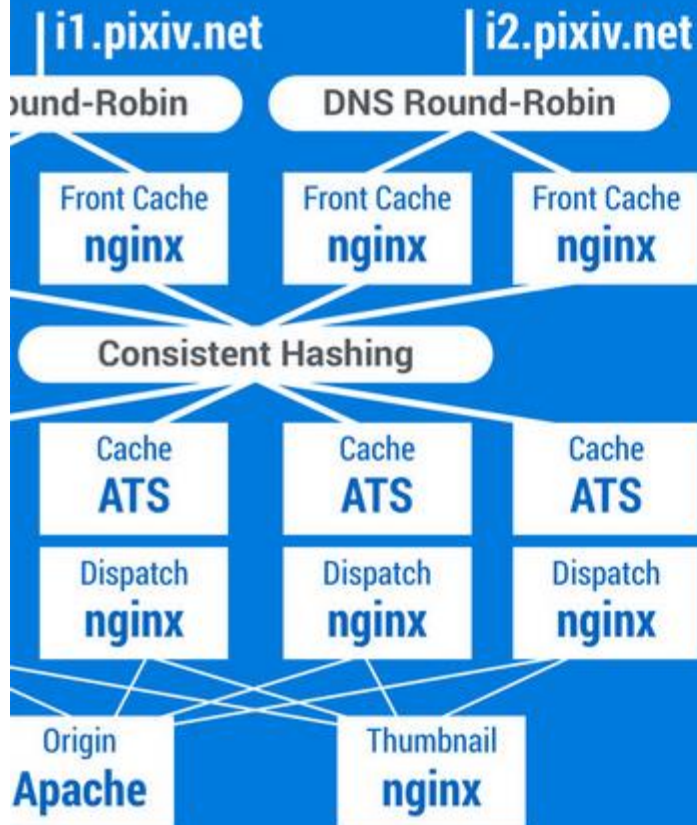
Image Cluster



Cache 전략

- 메모리와 디스크의 2단계 캐시 구성.
- 트래픽이 증가함에 따라 스위치간 트래픽을 무시할 수 없게 되었다.
- 네트워크 트래픽을 억제하면서 캐시 용량을 확보하는 것이 필요

Cache strategy



RAM 64GB Memory

- nginx cache on tmpfs
- cache hit rate: 50%
- reduce network traffic

SSD 256GB SSD x3 RAID0

- Apache Traffic Server (standalone)
- cache hit rate: 80-90%

HDD Original & BIG Thumb.

SSD Small Thumbnails

Aggregate Image domains

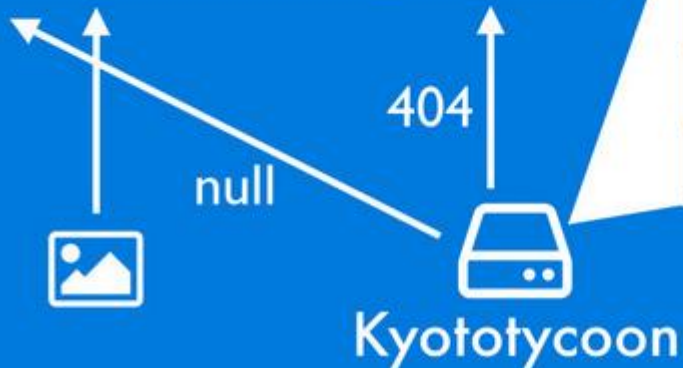
- 이미지 서버는 유저 ID 베이스로 분산
img01.pixiv.net - img1xx.pixiv.net
- 1 페이지로 40~60회 DNS 리퀘스트가 발생
-> 가정용 라우터로는 DNS 해석이 불가능
- 모든 이미지의 URI를 변경해서 대응

New image store

- ‘작성일 + id베이스’로 시킨스한 URI
- 한번 쓴 파일은 Read Only
- 재 투고 처리는작성일 갱신
- ngx_lua/OpenResty를 이용한 논리 삭제

Logical Delete

GET /img01.png GET /img03.png



/img02.png	403
/img03.png	404
/img05.png	404
/img08.png	403

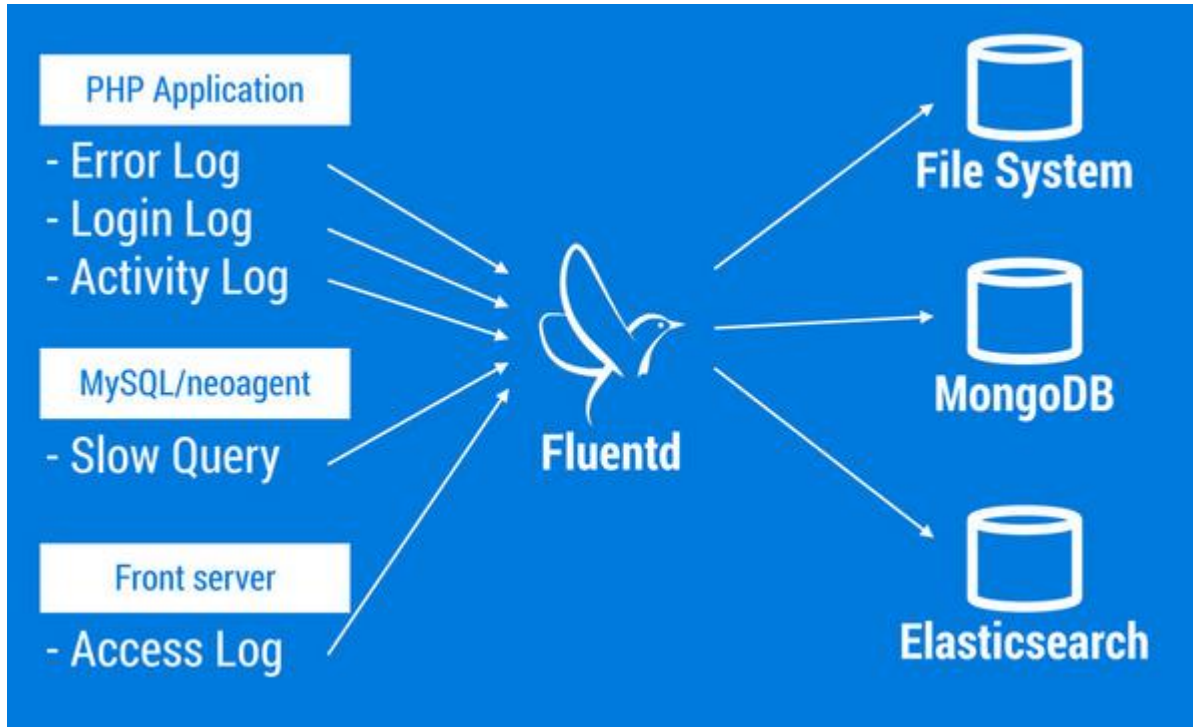
Logical Delete

```
location / {  
    access_by_lua_file logical_delete.lua;  
}
```

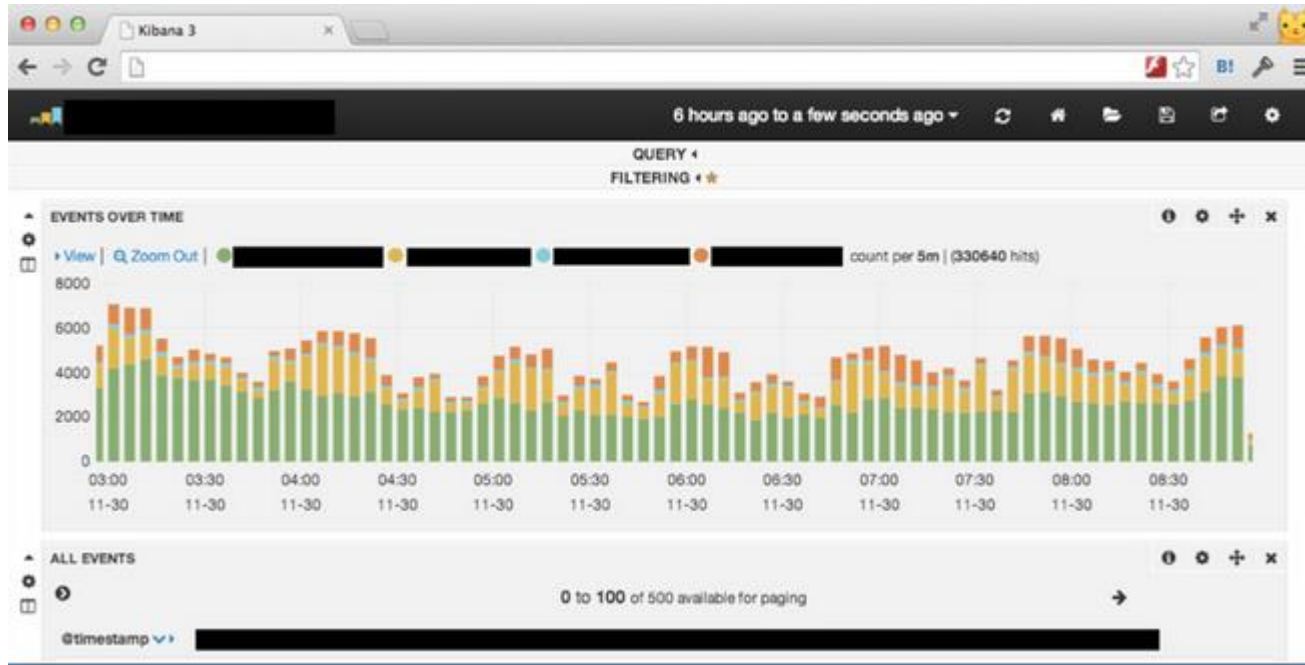
logical_delete.lua

```
local memcached = require "resty.memcached"  
local uri = ngx.var.request_uri  
local memc = memcached:new()  
.  
.  
.  
local val, flags, err = memc:get(request_uri)  
if val and val ~= "200" then  
    exit(tonumber(val))  
end
```

Log analysis Basic



Kibana 3



Output Log



- JSON으로 써서 Fluentd가 tail 하는 사양
- 프로세스가 대량으로 기동하여도 스루풋이 떨어지지 않음

Monitoring servers/services

극히 일반적인 감시 프로덕트를 사용하고 있다.

Mumin 리소스 그래프

Nagios 호스트/서비스 감시

Monit 호스트 내 프로세스 감시

Cron 스크립트

Cluster Admin

The screenshot shows the Cluster Admin web interface. At the top, there is a browser window titled "New Cluster Admin" and a page header "Cluster Admin - 各ホストのハードウェア情報、用途の管理する". A refresh indicator says "毎60秒00に自動更新されます".

The main content is a table with columns: LAN IP, VIP, use, memo, mem, disks, and machine. The table contains several rows of host data. A callout box labeled "監視状態" points to the status column, and another labeled "ホストの用途" points to the 'use' column.

Below the table, a detailed hardware view for a specific machine is shown. It includes fields for Machine, Serial, OS Version, eth0, and eth1. The CPU1 section shows "CPU1" and "16G 168G-128G-128G". The disks section shows four entries, each labeled "4G (Not Specified)". A callout box labeled "ハードウェア情報" points to this detailed view.

	LAN IP	VIP	use	memo	mem	disks	machine
▼	✓						
▼	✓						
▼	✓						
▼	✓						
▼	✓						
▲	✓				16G 168G-128G-128G		

Machine		CPU1			
Serial		/dev/sda		4G (Not Specified)	4G (Not Specified)
OS Version		/dev/sdb		4G (Not Specified)	4G (Not Specified)
eth0		/dev/sdc			
eth1					

Capistrano/Subversion

- /etc/이하의 설정 파일이 그대로 서버버전의 관리하의 디렉토리에
- 설정 반영은 **capistrano**를 사용하여 모든 머신에 디플로이
- 호스트 일람은 **API**를 통해서 얻을 수 있도록

ex: update DNS Record

```
$cap dns:update  
$cap dns:check  
$cap dns:reload
```

Management Tools

- LVS 관리 화면
- MySQL의 느려짐 감시
- 서비스 리소스 모니터링