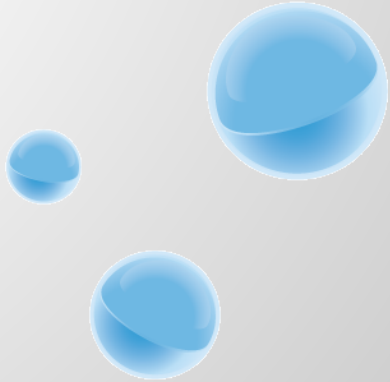
Cache invalidation strategies

with Varnish Cache

Per Buer / CTO / Varnish Software

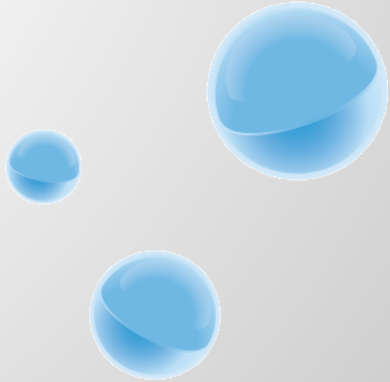


“There are only two hard things in Computer Science: cache invalidation and naming things.”

Phil Karlton

About Varnish Cache

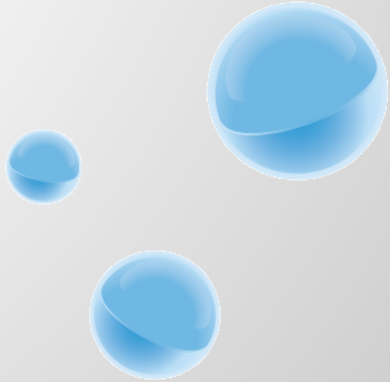
- Web app accelerator
- Fast
- Flexible




ORIGINAL
HEIDELBERG
CYLINDER
38 x 52 cm
15" x 20½"

About Varnish Software

- The company behind Varnish Cache
- Offers subscriptions with
  - Software
  - 24/7 support
  - Professional services

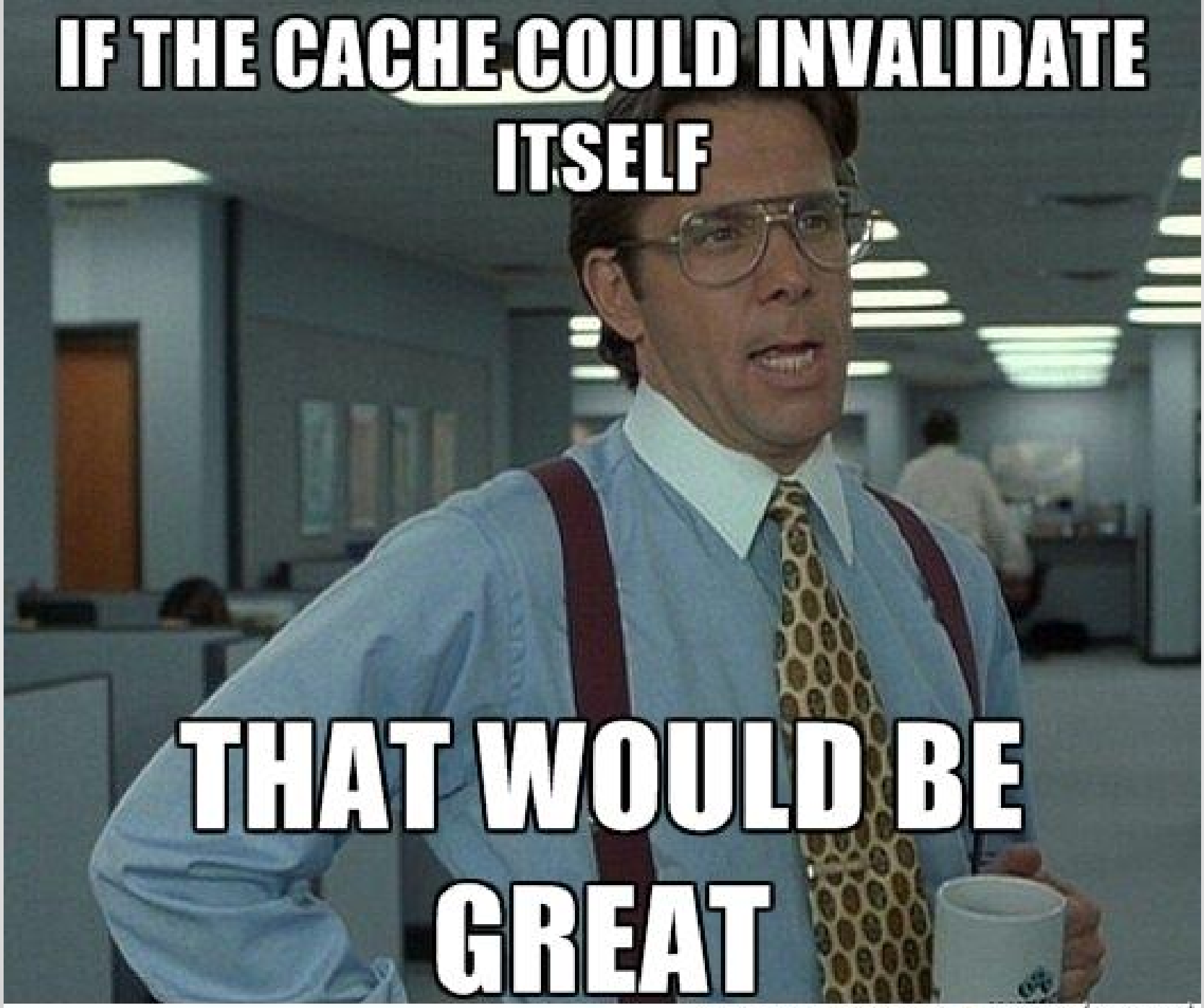


Goal

Run an efficient website with Varnish Cache

Why do cache invalidation?

- Allows for longer TTLs
  - Higher cache hit ratios
  - Better UX
  - Lower backend usage
- Instantaneous updates when content changes


IF THE CACHE COULD INVALIDATE ITSELF
THAT WOULD BE GREAT
MEMES & FUNNIEST GENERATED.COM

Components in Varnish

Components in Varnish we'll be covering

- PURGE
- ban
- Soft PURGE
- Soft Ban
- "Smart" bans
- Ban/purge distribution - VAC Super Fast Purger
- Hashninja

HTTP PURGE

- HTTP verb
- Takes URL as parameter
- Can purge all variants
- Derived from Squid

HTTP PURGE

```
PURGE /foo HTTP/1.1
Host: www.bar.com
```

PURGE VCL

```
acl purge {
   "localhost";
   "192.168.55.0"/24;
}

sub vcl_recv {
 if (req.request == "PURGE") {
  if (!client.ip ~ purge) {
    error 405 "Not allowed.";
    }
  return (lookup);
 }
}
```

```
sub vcl_hit {
 if (req.request == "PURGE") {
  purge;
  error 200 "Purged.";
 }
}

sub vcl_miss {
 if (req.request == "PURGE") {
  purge;
  error 200 "Purged.";
 }
}
```

PURGE VCL

```
acl purge {
   "localhost";
   "192.168.55.0"/24;
}

sub vcl_recv {
 if (req.request == "PURGE") {
  if (!client.ip ~ purge) {
    error 405 "Not allowed.";
    }
  return (lookup);
 }
}
```

```
sub vcl_hit {
 if (req.request == "PURGE") {
  purge;
  error 200 "Purged.";
 }
}

sub vcl_miss {
 if (req.request == "PURGE") {
  purge;
  error 200 "Purged.";
 }
}
```

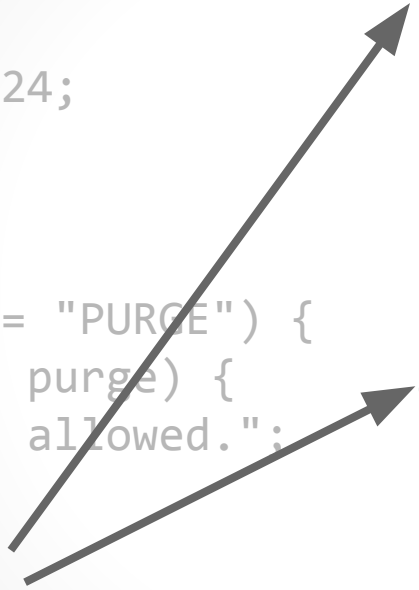
PURGE VCL

```
acl purge {
   "localhost";
   "192.168.55.0"/24;
}

sub vcl_recv {
 if (req.request == "PURGE") {
  if (!client.ip ~ purge) {
    error 405 "Not allowed.";
    }
  return (lookup);
 }
}
```

```
sub vcl_hit {
 if (req.request == "PURGE") {
  purge;
  error 200 "Purged.";
 }
}
```

```
sub vcl_miss {
 if (req.request == "PURGE") {
  purge;
  error 200 "Purged.";
 }
}
```



PURGE VCL

```
acl purge {
   "localhost";
   "192.168.55.0"/24;
}

sub vcl_recv {
 if (req.request == "PURGE") {
  if (!client.ip ~ purge) {
    error 405 "Not allowed.";
    }
  return (lookup);
 }
}
```

```
sub vcl_hit {
 if (req.request == "PURGE") {
  purge;
  error 200 "Purged.";
 }
}

sub vcl_miss {
 if (req.request == "PURGE") {
  purge;
  error 200 "Purged.";
 }
}
```

PURGE VCL

```
acl purge {
   "localhost";
   "192.168.55.0"/24;
}

sub vcl_recv {
 if (req.request == "PURGE") {
  if (!client.ip ~ purge) {
    error 405 "Not allowed.";
    }
  return (lookup);
 }
}
```

```
sub vcl_hit {
 if (req.request == "PURGE") {
  purge;
  error 200 "Purged.";
 }
}

sub vcl_miss {
 if (req.request == "PURGE") {
  purge;
  error 200 "Purged.";
 }
}
```

HTTP PURGE

- Fast
- Efficient
- Knows nothing about relationships between pages
- Doesn’t know about *grace*

Varnish bans

- Fast
- Flexible - can match almost any pattern
- Regular expressions on obj or req
- Not efficient
- Doesn’t know about *grace*

Varnish bans

CLI:

```
ban req.http.host == "example.com" &&
req.url ~ "\.png$"
```

HTTP:

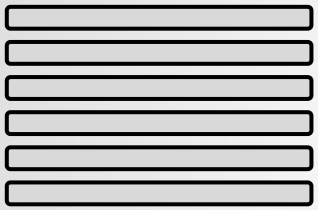
```
BAN /foo HTTP/1.1
Host: www.bar.com
```

Ban VCL

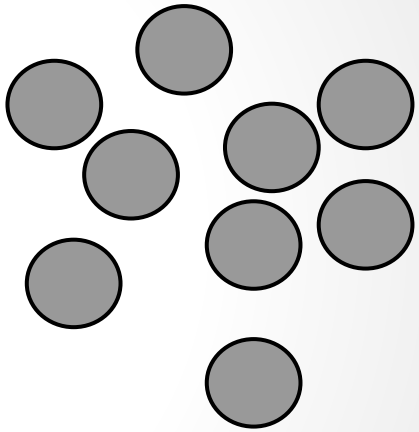
```
sub vcl_recv {
        if (req.request == "BAN") {
                if (!client.ip ~ purge) {
                        error 405 "Not allowed.";
                }
                ban("req.http.host == " + req.http.host +
                      "&& req.url == " + req.url);
                # Throw a synthetic page so the
                # request won't go to the backend.
                error 200 "Ban added";
        }
}
```

Ban list

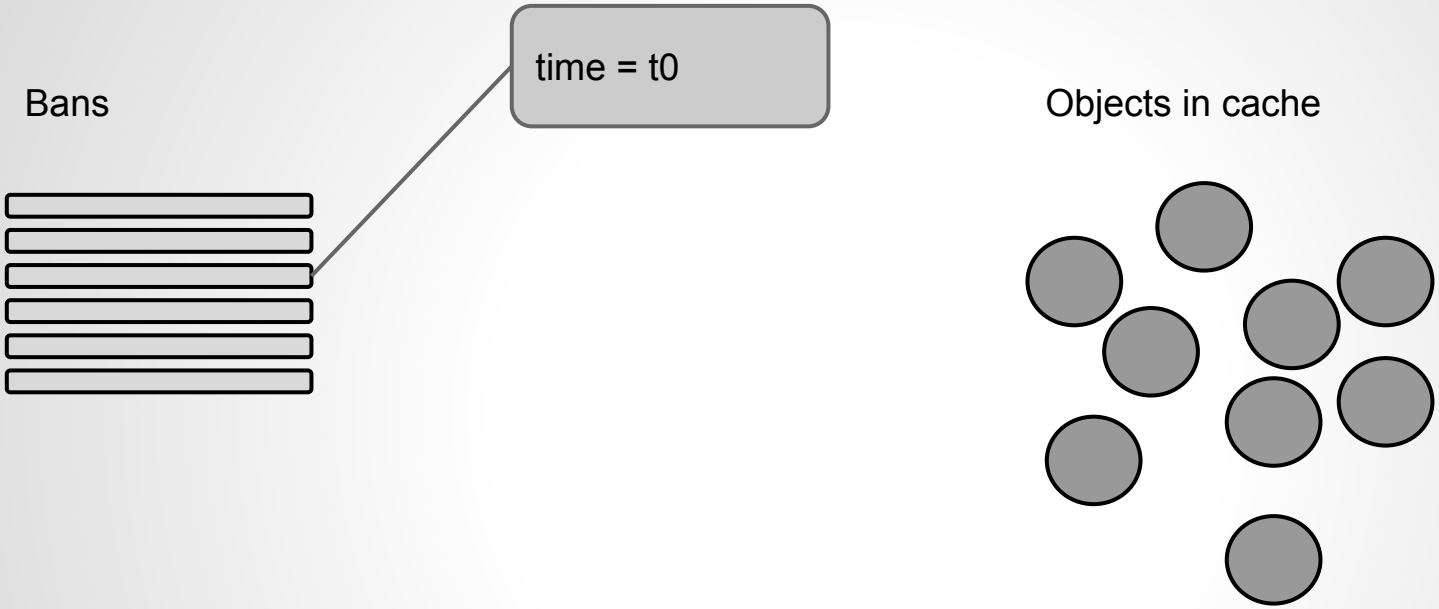
Bans



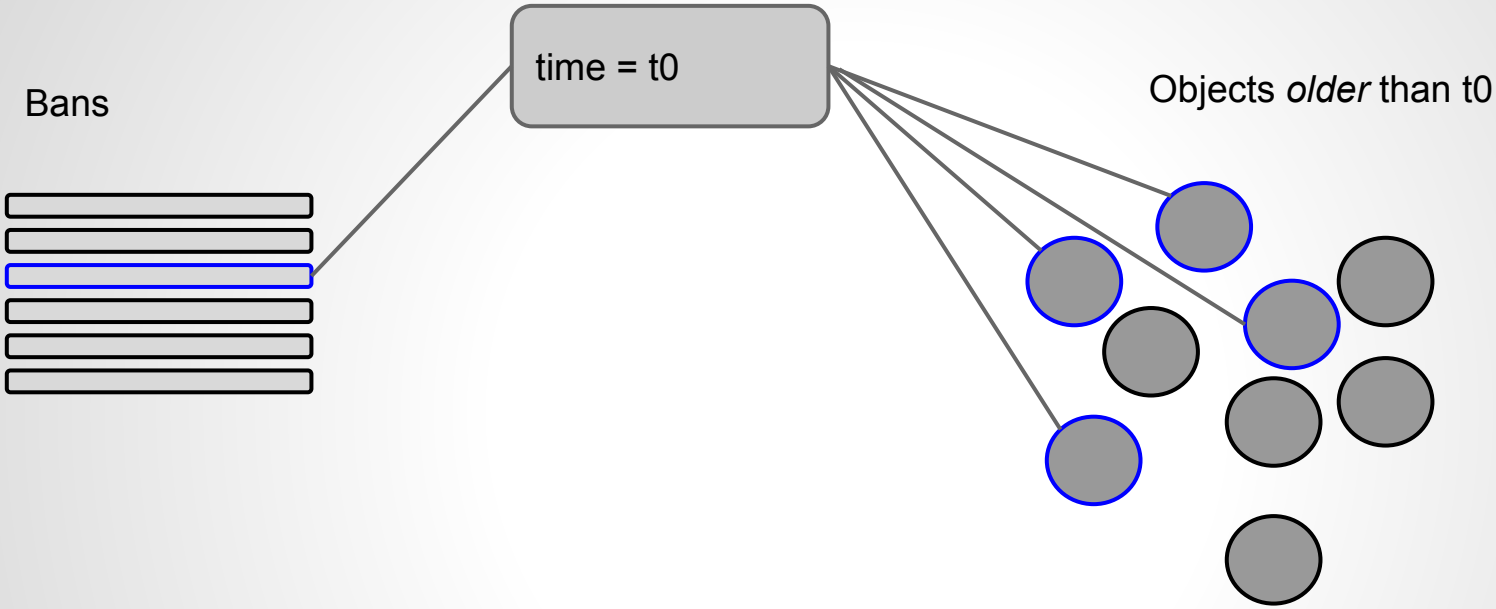
Objects in cache



Ban list



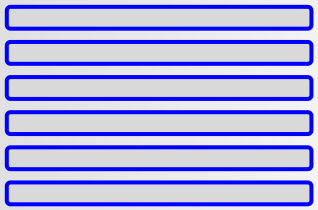
Ban list



- Each object matched only once against each ban.
- Potentially killed.

The ban lurker

Bans



- Worker thread
- Evaluates each ban against objects older than it
- Works only for bans on obj.*
- Kills a ban when it is matched against all objects older than t0.

Please do *smart bans*

- Avoid banning on req.*
- Copy the bits from req to beresp in vcl_fetch
- Keep an eye on the ban list and regex/sec
- Trim cache

Graceful cache invalidation

Graceful cache invalidation

- Problem: Purge object - backend goes down. No graced objects left to serve.
- “There is VMOD for that!”
- Marks objects as stale instead of killing them
- https://www.varnish-cache.org/vmod/soft-purge

What about graceful bans?

- Same as regular bans but objects are still subject to grace

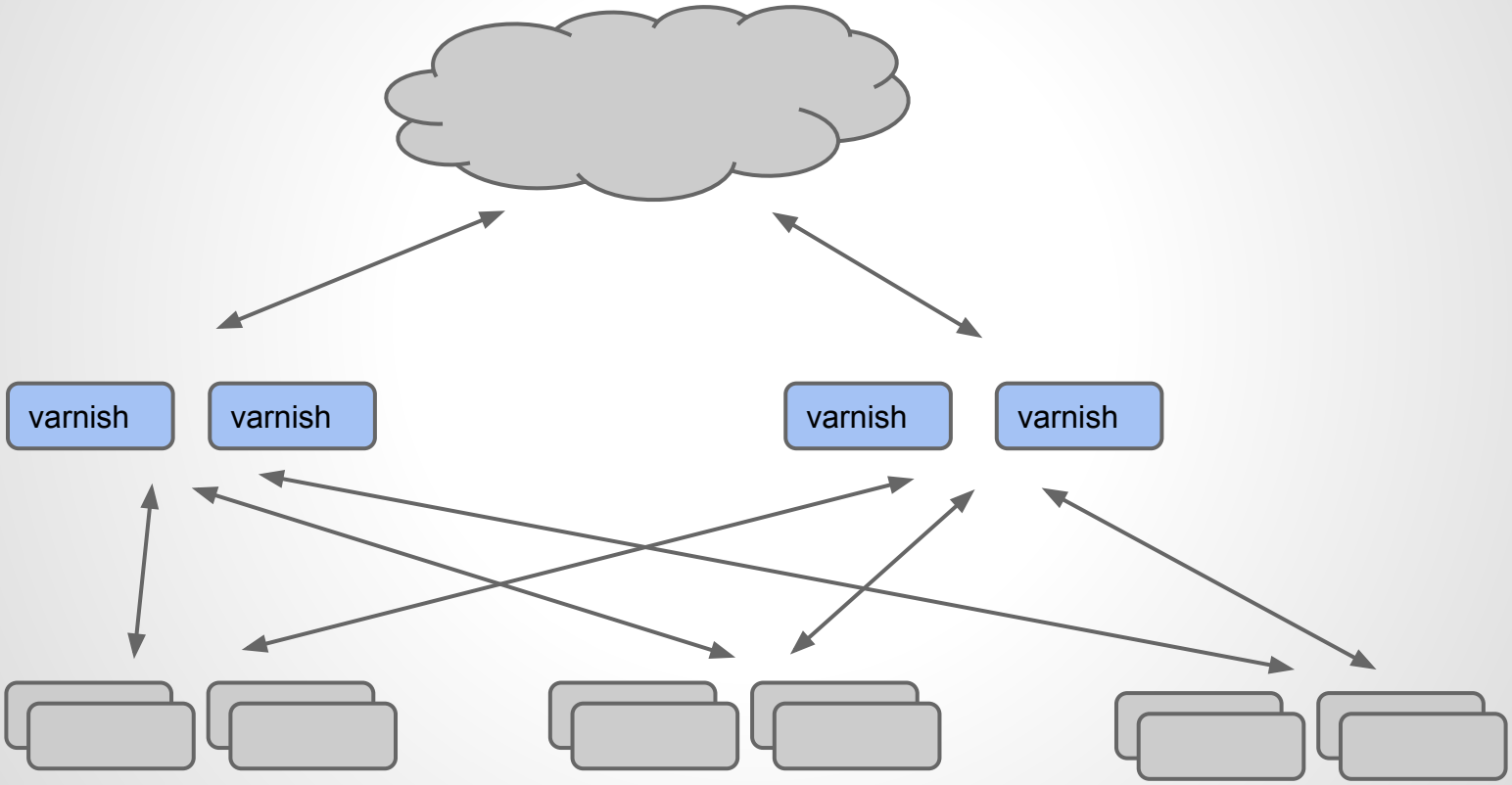
- Requires a patch for Varnish Cache - in VS Enhanced Varnish Cache.


Advanced topics

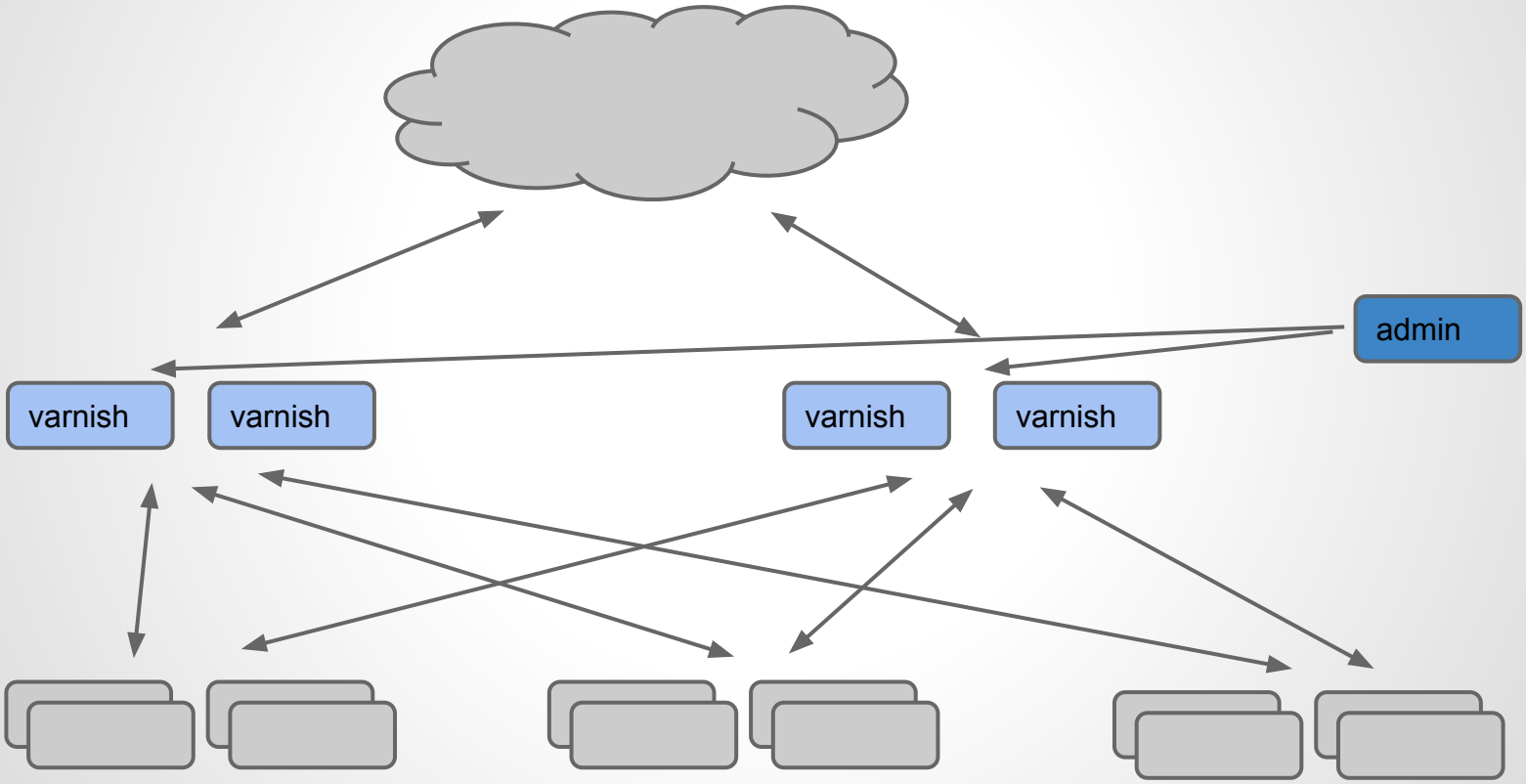
Distribution of invalidation events

- You don’t want every webapp to know about every varnish server
- Distribute invalidation events from a single point

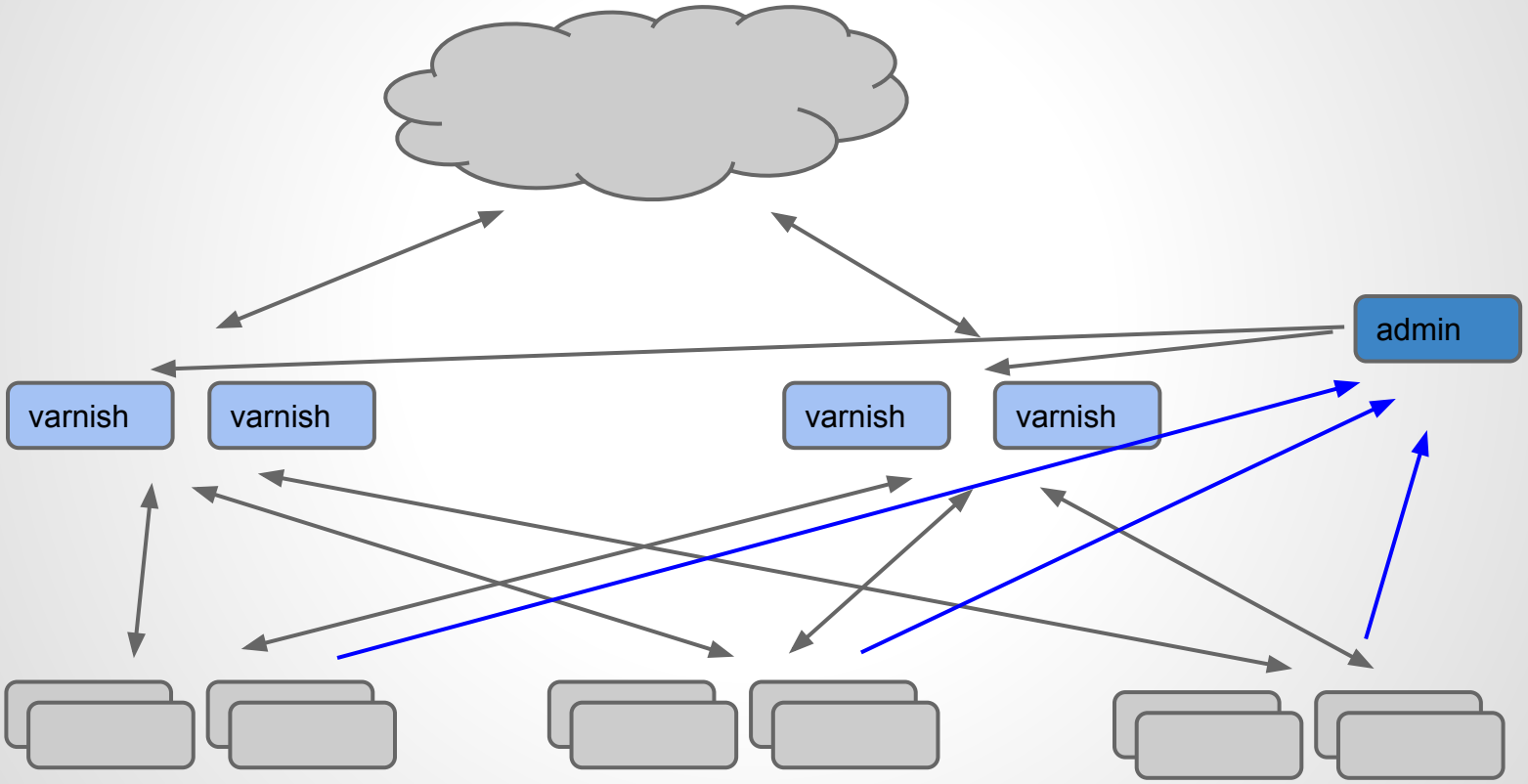
Distribution of invalidation events



Distribution of invalidation events



Distribution of invalidation events



Simplest invalidation distributor

```
nc -l 2000 | while true
 do read url
 for srv in “alfa” “beta” “gamma”
  do curl -m 2 -x $srv -X PURGE $url
 done
done
```

VAC Fast purger

- Fast API for event distribution
- 40 Kreq/s across datacenters

```
curl -X POST –user user:pw -H ‘Content-Type: text/plain’
-d ‘req.url ~ “/articles/FOO”‘
http://vac.local/api/v1/cachegroup/production/ban
```

Invalidation based on content relationship

- You have a web page with content from 8 different objects
- One object is updated
- Which pages to purge?

Samsung Galaxy Tab 3 7.0

www.bestbuy.com/site/Samsung+-+Galaxy+Tal

Processor Speed: 1.2GHz

See All Specs

Allow you to capture high-quality images.

microSD card slot
Lets you to access data stored on compatible cards (

Micro USB port
For fast digital video, audio and data transfer.

Bluetooth 3.0
Easily link with other Bluetooth-enabled devices, such

Weighs only 10.7 oz. and measures just 0.4"
For lightweight portability.

Preloaded apps
Include Google Play, YouTube, Dropbox, ChatOn, Ch
Paper Artist, Polaris Office, S Planner, S Voice, Sam
more.

Related Products

Samsung - Refurbished Galaxy Tab 2 10.1 with 16GB Memory - Titanium Silver

$249.99

Samsung - Refurbished Galaxy Tab 2 7.0 with 8GB Memory Verizon - Midnight Black

$249.99

Samsung - Refurbished Galaxy Tab 2 10.1 with 16GB Memory - Titanium Silver

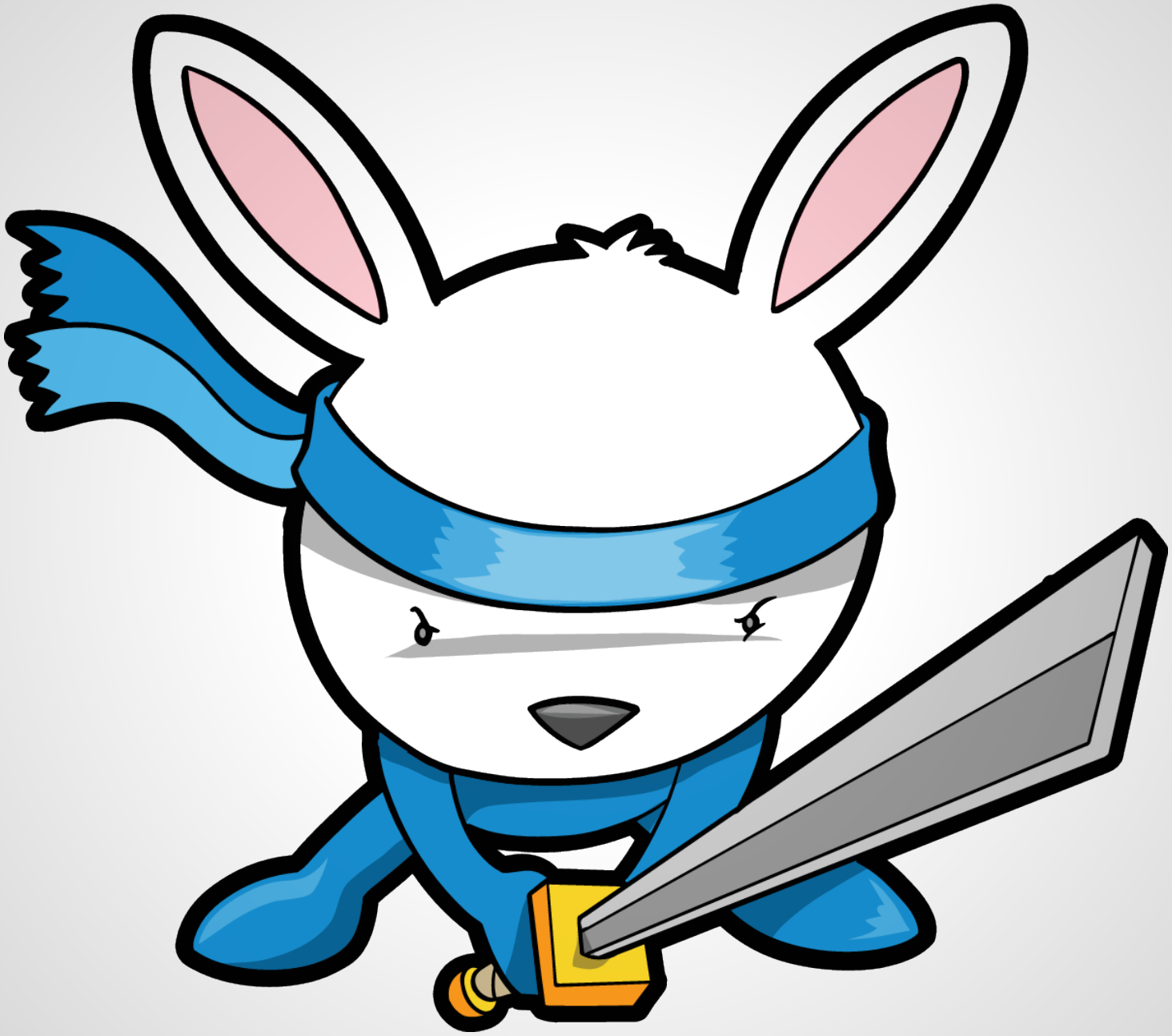
$259.99

Content tagging in Varnish

- Add X-Keys to each object (SKUs, article IDs or similar unique IDs)
- Identifies each object that is on the page
- Then you invalidate based on that unique ID.
- Every page that mentions that ID will be invalidated

Banning based on tagged content

- ban obj.http.x-keys ~ “[,]$ID\D”
- Suitable for low volume updates
- CPU usage will increase due to bans
- On high volumes you should check out….



Hashninja

- Maintains a hash with keys⇔pages
- Many-to-many
- Very low overhead, high performance
- Requires subscription + Proprietary VMOD
- Suited for e-commerce and digital media

Summing up

- Purges
- Bans
- Soft purges and bans
- Smart bans
- Hashninja and content tagging

Thanks!

Questions and comments, please.

Get in touch: per.buer@varnish-software.com