

# IMS

## Backend Conditional Requests aka Cache Refresh/Refresh

**VUG5**

**March 23, 2012**

**Geoffrey Simmons  
Nils Goroll**



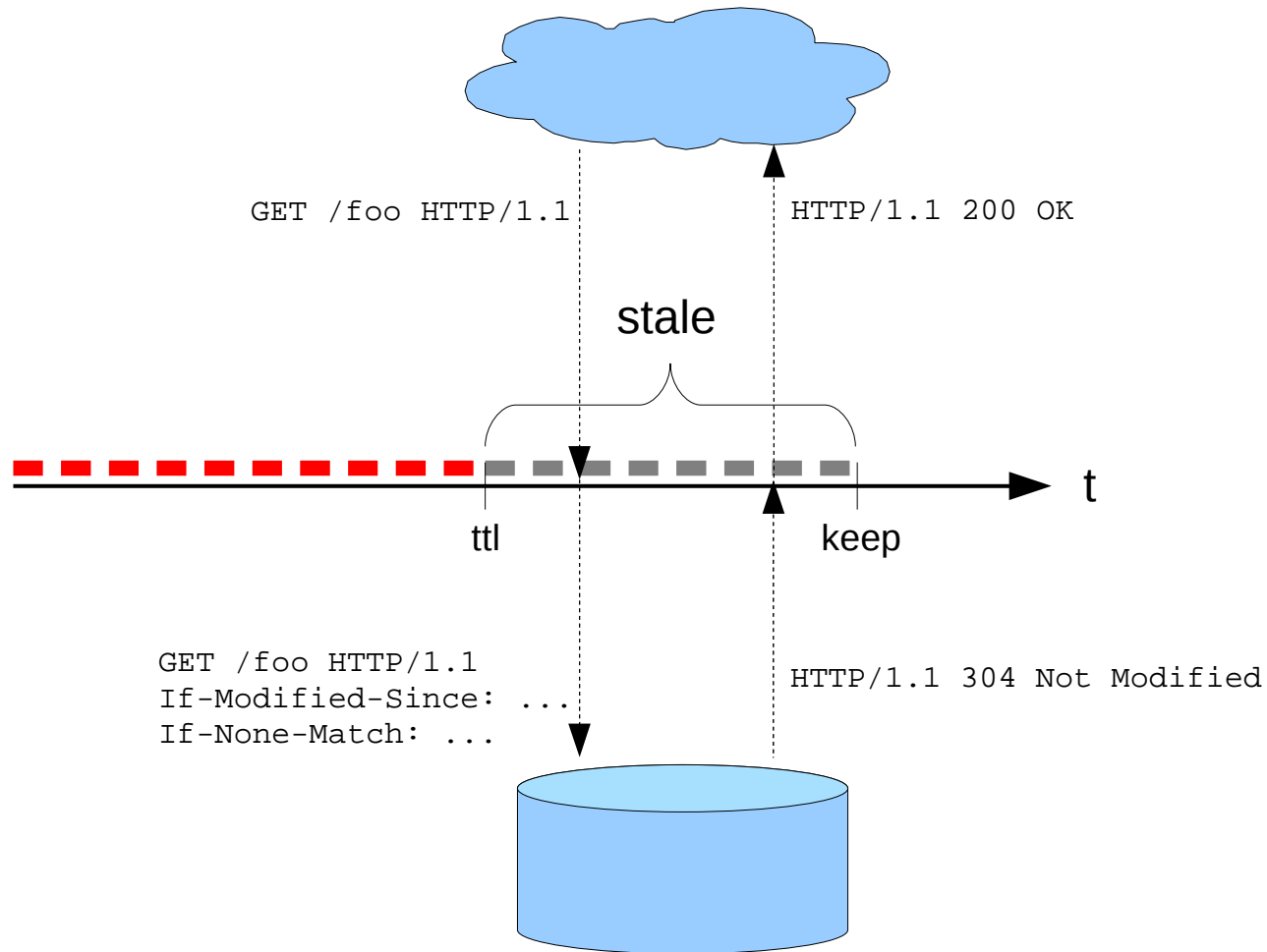
# Agenda

- **brief history**
- **experimental-ims**
  - How it works
  - Refresh and storage
- **Configuration**
  - keep, stale\_obj
  - vcl\_stale(), vcl\_refresh()?
- **Too complex?**

# Brief History

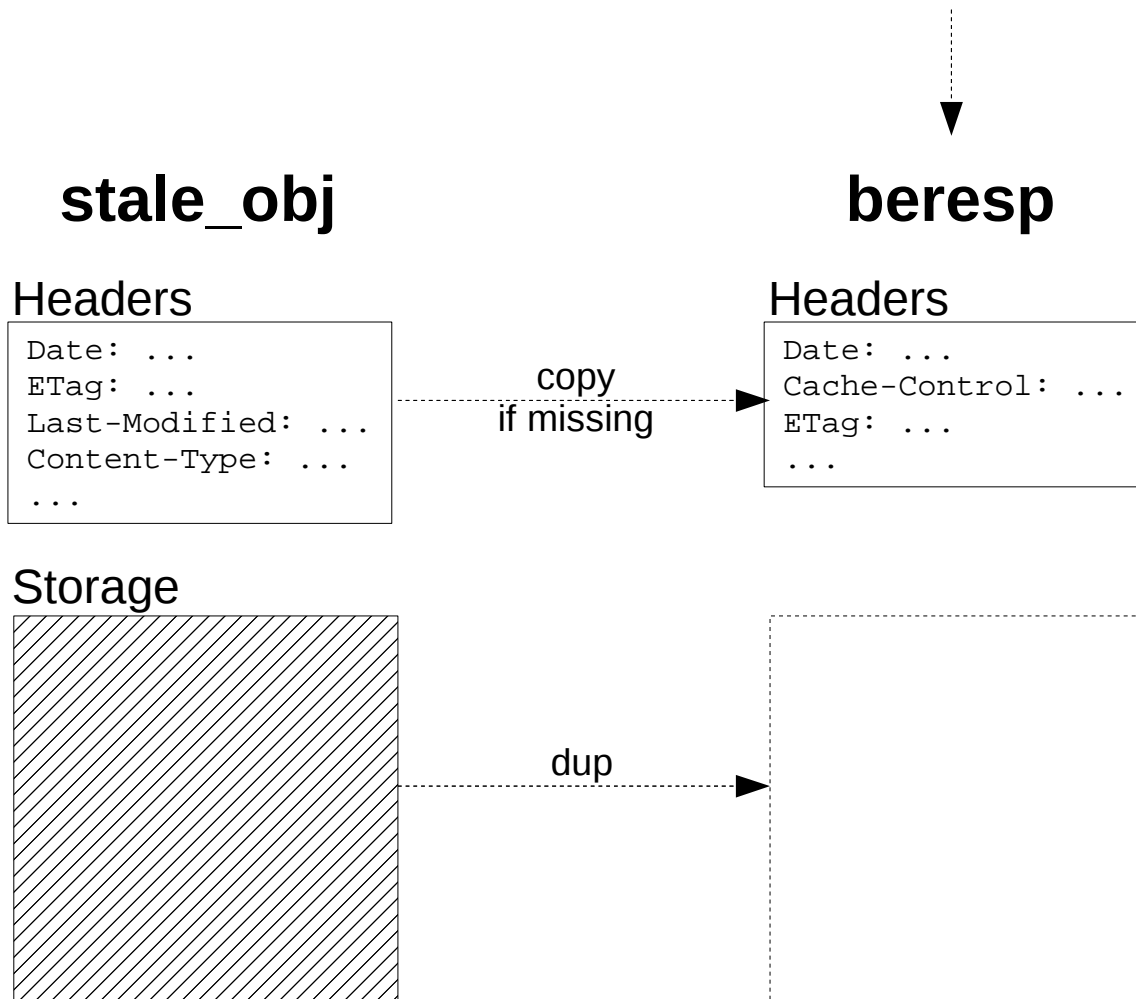
- **2010-09-10**
  - **Received first Rackspace Implementation**
- **2010-09-23**
  - **Proposed design on varnish-dev**
- **2011-03-01**
  - **First patch posted**
- **Since 2011-06-02**
  - **Maintaining git branch experimental-ims**

# experimental-ims



# Refresh & storage

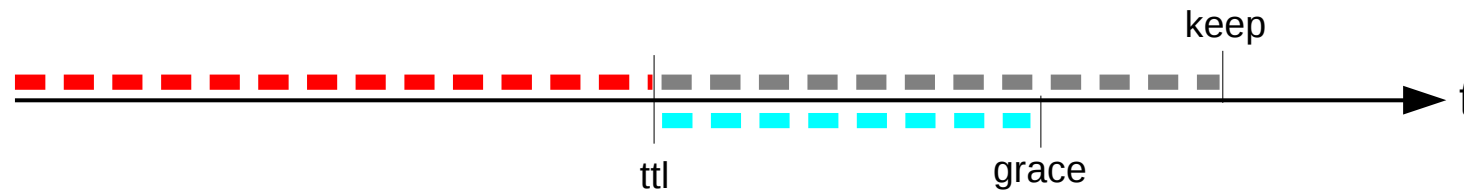
HTTP/1.1 304 Not Modified



# Storage dup

- **dup == copy**  
**is unavoidable for -spersistent**
- **dup for other stevedores**
  - **Should be:**
    - **New object points to stale\_obj's storage**
    - **Refcounts for storage**
  - **Currently in experimental-ims:**
    - **Copy for all stevedores**

# keep & grace



## In experimental-ims:

- **Both intervals begin after ttl**
  - **keep == grace, keep < grace, keep > grace**  
**all possible**
- **Expiration after ttl + max(keep, grace)**
- **grace has priority in overlapping intervals**

## Currently in experimental-ims:

```
sub vcl_miss {  
    if (stale_obj) {  
        set bereq.http.X-Foo = stale_obj.http.X-Bar;  
    }  
}
```

- **All fields read-only**
  - Otherwise just like obj
- **In vcl\_miss(), vcl\_fetch(), vcl\_error()**
  - May or may not exist!



# stale\_obj

**# DON'T DO THIS!**

```
sub vcl_miss {  
    set bereq.http.X-Foo = stale_obj.http.X-Bar;  
}
```

# Always check stale\_obj for existence

```
sub vcl_miss {  
    if (stale_obj) {  
        set bereq.http.X-Foo = stale_obj.http.X-Bar;  
    }  
}
```

- **No way to prevent this error**
  - SHM log records a VCL\_error message
- **Does stale\_obj need to be exposed?**

# vcl\_stale(), vcl\_refresh()

**vcl\_stale() ↔ vcl\_miss()**  
**vcl\_refresh() ↔ vcl\_fetch()**

- **stale\_obj always exists**
  - **stale\_obj could just be obj**
- **Two new vcl\_subs()?**

# Generalized vcl\_stale()

- **Either the grace or stale/keep scenario**
  - Alternative to vcl\_hit() & vcl\_miss()
  - Just one timer after ttl?
- **Idea: Pull C-logic into vcl**
  - pseudocode for HSH\_Lookup  
(after searching for the "best" object)

```
if (oc && (now < ttl))
    next step is HIT;
else if (oc && (now < ttl + keep))
    next step is STALE;
else
    next step is MISS;
```

# vcl\_stale() in default.vcl

```
sub vcl_stale {  
    /* existing grace logic */  
    if (req.busy || !req.backend.healthy)  
        return (deliver);  
    return (fetch)  
}
```

- **deliver**: deliver the stale object
- **fetch**: attempt a (possibly conditional) fetch
- **pass, error, restart**: as usual

# vcl\_stale()

```
sub vcl_stale {  
    /* only part of the current grace logic */  
    if (!req.backend.healthy)  
        return(deliver);  
  
    /* override conditional requests */  
    if (req.url ~ "^/always-fetch") {  
        unset bereq.http.If-Modified-Since;  
        unset bereq.http.If-None-Match;  
        return(fetch);  
    }  
  
    /* deliver for restarts */  
    if (req.url ~ "^/app" && req.restarts > 0)  
        return (deliver);  
}
```

# vcl\_refresh() ?

## Expose the stale object after fetch?

If so ...

```
sub vcl_fetch {  
    if (stale_obj && stale_obj.hits > 10000)  
        set beresp.ttl = 30m;  
}  
  
/* ... or ... ? */  
  
sub vcl_refresh {  
    if (obj.hits > 1000)  
        set beresp.ttl = 30m;  
}
```

# Thanks!

## Questions?

- [nils.goroll@uplex.de](mailto:nils.goroll@uplex.de), +49-170-2723133
- [geoff.simmons@uplex.de](mailto:geoff.simmons@uplex.de), +49-176-63690917

