ATS Summit: CARP Plugin

Eric Schwartz

YAHOO!

Outline

- CARP Overview
- CARP Plugin Implementation
- Yahoo! Insights
- CARP vs. Hierarchical Caching
- Other CARP Plugin Features
  - Blacklist/Whitelist
  - Pre- vs. Post-remap Modes
  - Object Replication
- Questions

What is CARP?

- Cache Array Routing Protocol
- Internet-Draft from Microsoft
- Notable implementation: Squid, used for load-balancing across a cluster of proxy servers
  - Very similar to ATS + CARP plugin
- Nodes healthcheck each other to maintain list of available nodes

How CARP Works (Simplified)

```
For each incoming request:
HashNode node = GET_SCORE_FROM_HASH(URL)
If node == self:
    Handle request normally
Else:
    Forward request to node (with added loop prevention header), sets node with TSHttpTxnServerAddrSet
    Do not cache result
    Respond to client
```

CARP Request Lifecycle 1

CARP Request Lifecycle 2

CARP Request Lifecycle 3

CARP Request Lifecycle 4

CARP Request Lifecycle 5

CARP Request Lifecycle 6

Configuring CARP

- Used as a global plugin.
- Example of how to configure:

```
# An Example Config:
#
[Servers]
# port 4080 on host1.cacheservice.com with weight factor of 2
host1.cacheservice.com:4080 weight=2
# port 80 on host2.cacheservice.com with (default) weight factor of 1
host2.cacheservice.com
[Values]
healthcheck=http://healthcheck.cacheservice.com:8080/status.html
healthfreq=30
blacklist=healthcheck.cacheservice.com,hc.cacheservice.com
mode=pre-remap
```

Other Things to Mention About CARP/Yahoo! Insights

- We use consistent hashing so that if we lose a node only the objects set to be cached on that box have to be rehashed.
- Tradeoff: probability that a request will be forwarded to another host in the cluster increases with cluster size.
  - For us, in practice this adds about a few ms of latency to majority of requests.
  - Better than having to go all the way back to origin for a much larger proportion of requests.
- Hosts can be assigned different weights for % of objects they own. Useful if storage space of hosts in cluster differs.
- Healthchecks currently peer-to-peer with n^2 scaling. Less than ideal.

Other Things to Mention About CARP/Yahoo! Insights

- CarpHashFunction is easily extendable/replaceable.
  - Easy to modify the logic in getRemapProxy and getRemapProxyList, the heart of how CARP determines peer to forward to.
- Example:
  - Could modify hash function to keep track of “hot objects” and cache these on multiple hosts so that frequently requested assets are available to be served from cache on every host.

Other Things to Mention About CARP/Yahoo! Insights

- Because every forwarded request hits two boxes, you'll get an extra TCP_MISS in logs on the first box for each request of this variety.
- Can still calculate this for entire cluster from your logs with by tracking ATS-Carp-Forwarded header.

```
cluster_miss = cache_miss - carp_forwarded
cluster_hit = cache_hit
cluster_chr = cluster_hit / (cluster_hit +
cluster_miss)
```

Other Things to Mention About CARP/Yahoo! Insights

- Health checking:
  - Each host periodically fetches a healthcheck URL from each other configured host, expects 200 response.
  - Obviously, this doesn’t scale well. Not something we’ve run into problems with, but for larger deployments could create a lot of internal network traffic.

CARP vs. Hierarchical Caching

- With hierarchical caching, if trafficserver does not have an object in its cache it can go to a parent (or series of parents) to search for it. With CARP, if host is not owner then it goes to at most one other node in the cluster for any given URL.
- With hierarchical caching, if trafficserver finds the object in a parent’s cache it will also cache it after responding. With basic configuration of CARP, each object will live on at most one node (perfect sharding of cache). (This behavior can be modified by plugins).

Other CARP Features: Blacklist and Whitelist

- CARP plugin allows for blacklisting and whitelisting of certain URLs. Useful for healthchecks as an example.
- If a host is on the blacklist or not on the whitelist then CARP will treat it normally, will not forward.
- TODO: want to extend black- and whitelists to support regex.

Other CARP Features: Pre- vs. Post-Remap Mode

- CARP can hash on either the pre- or post-remap URL.
  - If a lot of assets remap to similar URLs, running on post-remap URL can in practice increase overall cluster CHR pretty dramatically.
  - However, post-remap mode can behave poorly with some other plugins.
  - Post-remap forwarding happens on a separately configured incoming port to prevent abuse as a forward proxy. A header with the encoded version of the destination origin server is also added.
- Pre-remap is default/what we utilize more at this point.

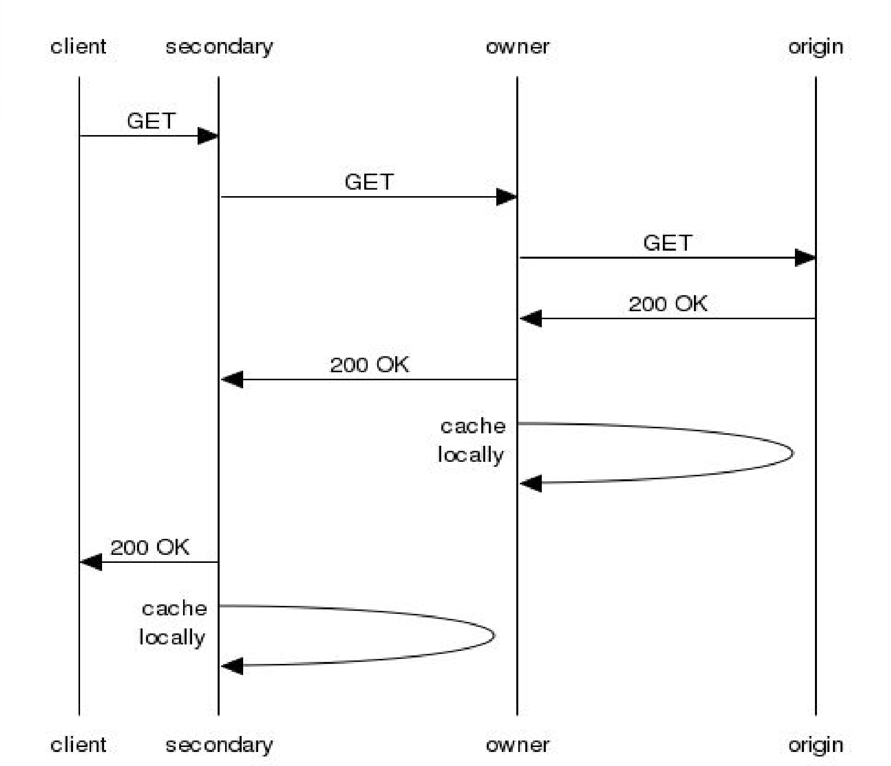
Other CARP Features: CARP Replication Motivation

- If we lose a host or group of hosts due to a network failure or for operational reasons, all of the objects on those hosts will rehash to different hosts in the CARP pool.
- Those hosts, having assumed ownership of the object, will go fetch it from the origin.
- If this happens for a large enough chunk of the CARP pool, it can cause those origins to fall over.

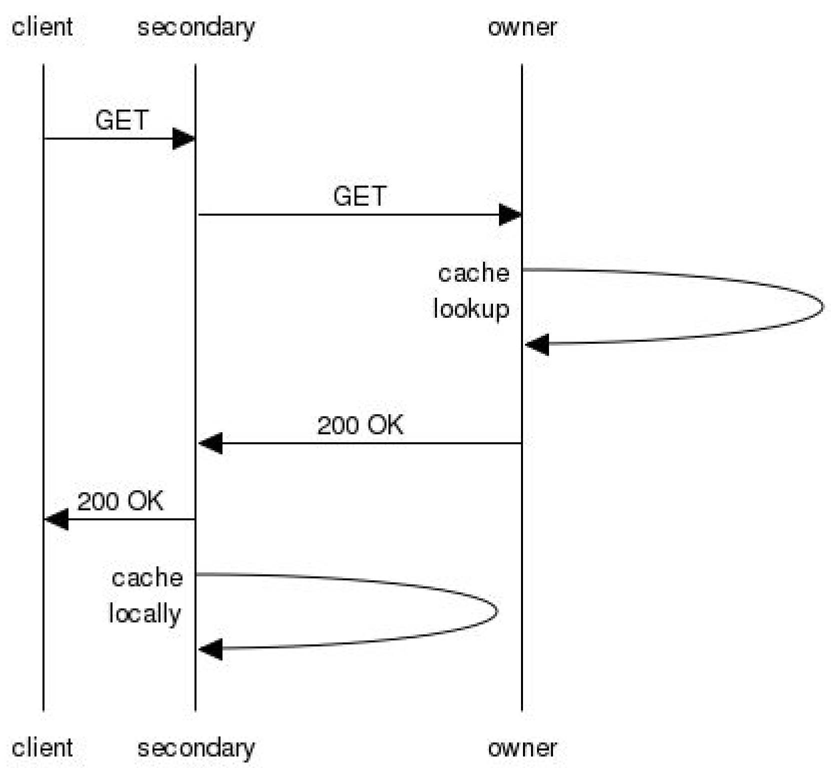
Other CARP Features: CARP Replication Algorithm

```
If object in local cache:
    Serve it.
Else:
    If Carp-Routed Header present:
        Fetch from origin and cache.
    Else:
        Generate owner list of size K for object.
        Get first in list
        If first is self:
            Fetch from origin.
        Else:
            Fetch from first in list (primary owner).
    Upon successful retrieval of the object, if this node is in owner list, cache the object.
```

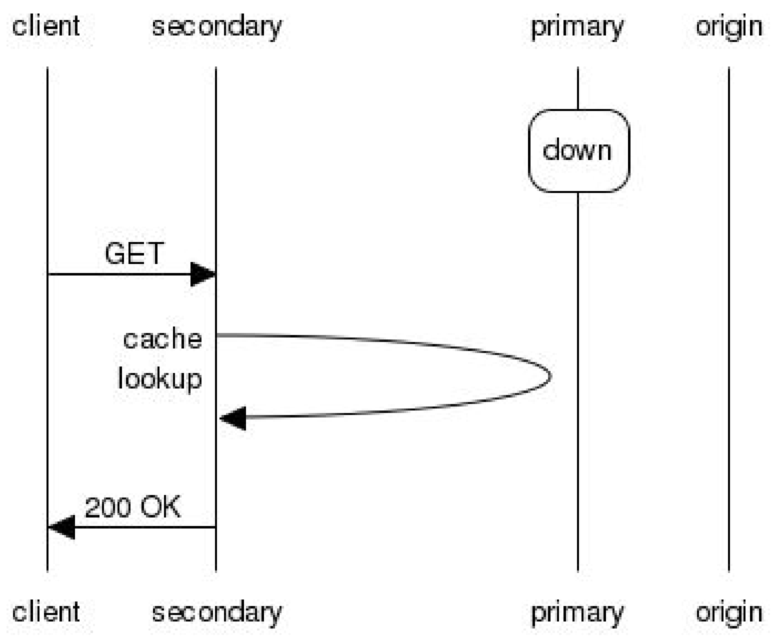
CARP Replication: Example Scenario 1



CARP Replication: Example Scenario 2



CARP Replication: Example Scenario 3



CARP Replication: Groups

- Sometimes useful to specify different groups of hosts when replicating objects.
  - Ex: network topography. Want objects to be replicated across multiple pods or multiple switches.
- This can be configured if replication is turned on. Objects will be spread “evenly” across groups according to config rather than simply across hosts.

The Big Question

- When will this be available in open source ATS?
  - Pull request going through code review now. Working on writing some tests for Grigor for this using Jason’s new framework. Will open source those too.
  - Shooting to be in 7.1.
  - TS-4723