YAHOO!

Stale While Revalidate (Cache Open Write Fail)

Sudheer Vinukonda

Introduction

ATS Cache vs HTTP GET Request

- ❑ Cache Lookup (lookup for the dirent)
  - ➢ Success - try to get a read lock
  - ➢ Failure (cache miss) - goto Cache Open Write
- ❑ Cache Open Read (try to get read lock)
  - ➢ Success - serve from cache, Done
  - ➢ Failure - goto Cache Open Write
- ❑ Cache Open Write (try to get write lock)
  - ➢ Success - download the object into cache and to the client in parallel
  - ➢ Failure - disable cache, and download to the client in a proxy-only mode

ATS Connection Collapsing (Open Write Fail)

- ❑ ATS connection collapsing (with "Read-While-Writer") only works after response headers arrive
- ❑ Until then, ATS leaks all traffic upstream, overloading that layer
- ❑ Typical large scale CDN deployments may have multiple cache layers (e.g. Edge, Mid, Origin)

- ❑ ATS 6.0 can detect Write Lock failure - TS-3549
  - ➢ Cache Revalidation – return stale copy
  - ➢ Cache miss – return 5xx error with special @header
- ❑ Custom TS plugin can intercept the 5xx error, perform an internal 3xx redirect (with delay)
  - ➢ Configurable delay N X T msec
  - ➢ If the response headers are available before the retries are exhausted, can stream all the blocked requests in parallel (Read-While-Writer)

Stale While Revalidate using Open Write Fail

❑ stale-while-revalidate plugin

- ➢ Returns stale copy on the first request
- ➢ Blocks all but one request due to Write Lock

❑ Open Write Fail feature

- ➢ Returns stale copy for all but first request
- ➢ To be enhanced: return stale copy even for the first request (refresh in the background)
- ➢ To be enhanced: RFC 5861 compliance (SWR headers, max-stale-age etc)

Summary

- Open Write Fail feature solves the Thundering Herd problem
- Proven to work with near perfect connection collapse for HLS live streaming use case
- Improves latency significantly
- Allows upstream capacity requirements to be predictable
- Allows to improve the SWR solution in the core