

ALTO caching and subscription

draft-alto-caching-subscription-00

Sreekanth Madhavan	Huawei
Nandiraju Ravishankar	Huawei

ALTO Caching – Current Scenario

- ALTO clients/proxies cache map information for further use
 - Reuse existing HTTP caching mechanisms
- Issues
 - Responses to filtered map queries are not cached (because filtered map queries use POST)
 - Cache invalidation is based on HTTP mechanisms which applies to the whole document. No support for caching/invalidating a subset of content(Single PID level information)

ALTO caching mechanism

- Specify expiration time for complete map or subset level maps
 - Add expires value in the JSON object level

Advantages:

- 1) Clients/proxies cache only the specific PID level information
- 2) On expiration, clients/proxies can query specific map information and need not query the whole map information.

Disadvantage

- 1) Weak consistency

Cache consistency

- Strong consistency
 - Polling every time
 - Notification mechanism
- Weak consistency
 - TTL based

Tradeoffs

- Assume that interleave of requests and file modifications happens as “RRRMMMRRMRRRMMRM”
- Let R be the number of times a node views a resource and RI be the number of intervals during which client repeatedly asks for resource while resource is changed

Messages	Polling every time	Notification	TTL based
GET requests	1	1	1
If-Modified-Since	$R - 1$	0	TTLmissed -1
304 response	$R - RI$	0	TTLmissed – TTLmissed_res_change
Subscription + Notification	0	$1 \text{ Sub} + RI * \text{Not}$	0
Total control message	$2R - RI$	$1 \text{ GET} + 1 \text{ Sub} + RI * \text{Not}$	$2 * \text{TTLmissed} - \text{TTLmissed_res_change}$
File Transfers	RI	RI	RI – Stale hit intervals
Stale document	No	No	Yes

Notification mechanism

- Clients subscribe for event changes
 - Notified when event changes.
 - Supports refresh and delete subscription.
- Supports incremental updates.
 - Sends changed information for e.g: vtag and changed map information
 - Reuses HTTP conditional request and IM headers for delta encoding(RFC 3229)

Incremental updates

- Encoding algorithms can be indicated by client using A-IM header.(diffe, gzip, vcdiff)
- How many base instances server needs to store ?
 - Does not store any old instances
 - Store the most recent prior instance
 - Store deltas between specific base instances and subsequent instances

Questions ?

Backup slides

Incremental update example

GET /foo.html HTTP/1.1

Host: bar.example.net

If-None-Match: "123xyz"

A-IM: vcdiff, diffe, gzip

HTTP/1.1 226 IM Used

ETag: "489uhw"

IM: vcdiff

Date: Tue, 25 Nov 1997 18:30:05 GMT

Cache-Control: no-store, im, max-age=30

...