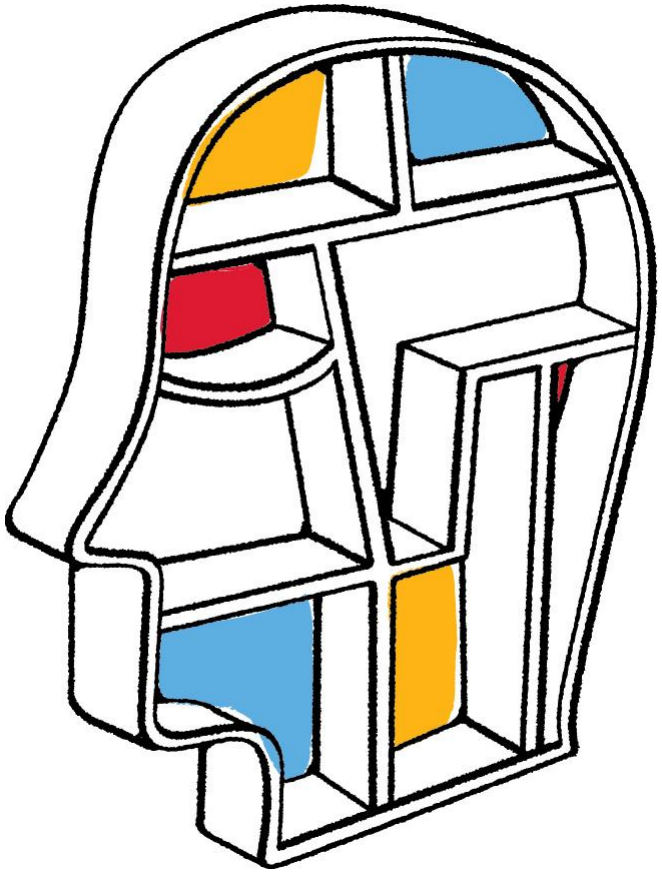
NetApp

Go further, faster®

NFSv4.1 dynamic slot allocation

Trond Myklebust
<trond@netapp.com>



What is dynamic slot allocation?

- A tool for managing **global** session resources
  - Allows dynamic resizing of the replay cache on a per-client, per-load basis
    - The client communicates to the server whether or not it can fill all slots.
    - The server then decides how many slots it should allocate to that client in the future.
    - Communication occurs via the SEQUENCE operation, which means that updates occur on every COMPOUND.

How does the client communicate load?

- The session slots are numbered from 0…n.
- The client is required to allocate all slots from 0…n-1, before it can use slot n.
- In each SEQUENCE call, the client fills the sa_highest_slotid field to reflect the highest slot number in use *at the time the SEQUENCE was sent*.

How does the server reply?

- The server fills the sr_highest_slotid with the highest slotid that the client is allowed to use.
  - This is the highest slotid for which the server is caching the sequence number.
- It fills the sr_target_highest_slotid with the highest slotid that the client should use in the future.
  - IOW: as soon as the client sees this target, it should stop allocating new slotids > target.

Some notes

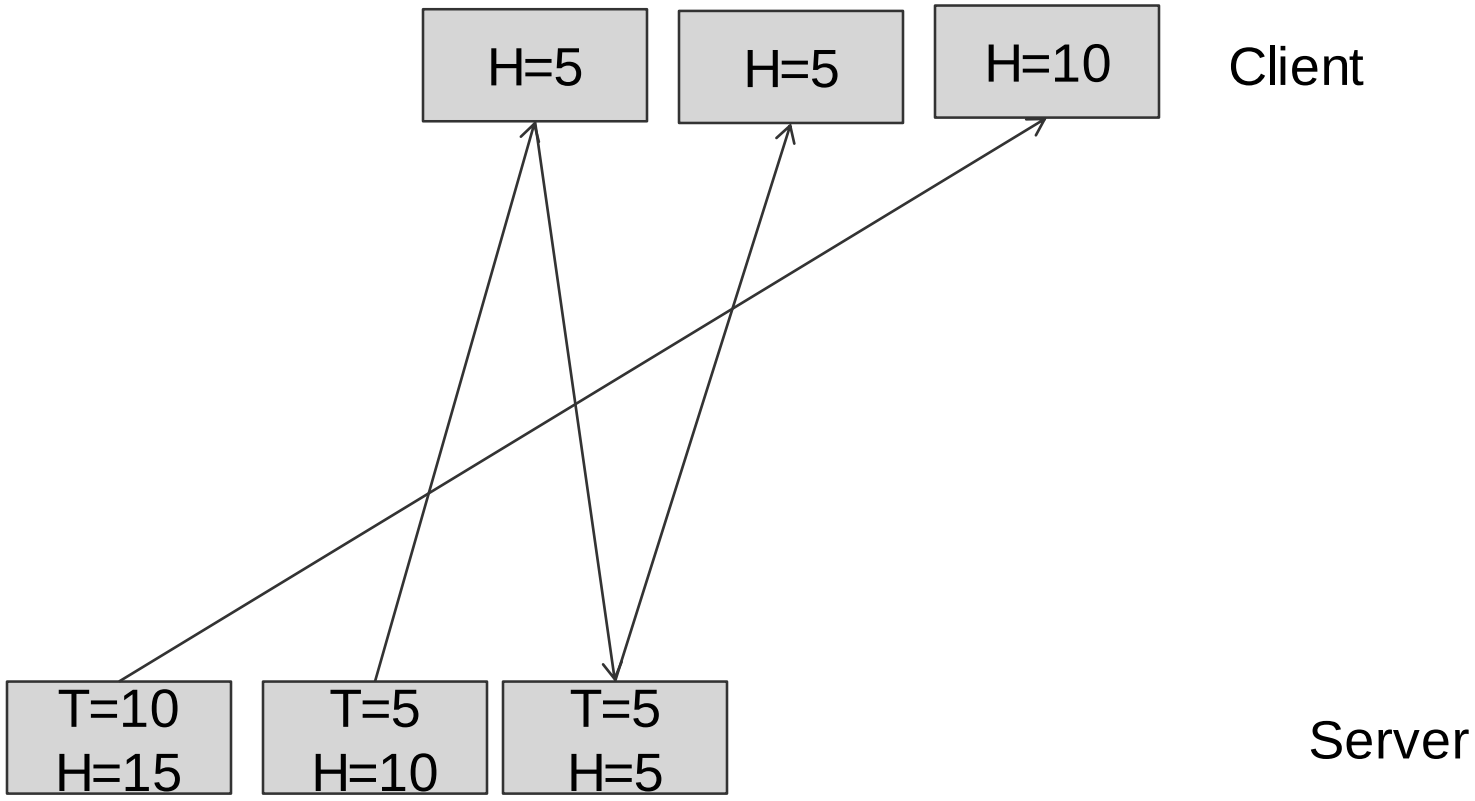
- sr_target_highest_slotid <= sr_highest_slotid
- Since dynamic slot allocation is not a mandatory feature (but a really useful one), then servers SHOULD ensure that for clients that don't support dynamic slot allocation, sr_highest_slotid >= csr_fore_chan_attrs.ca_maxrequests-1 (see CREATE_SESSION).

Sounds easy. Where's the catch?

- Asynchronous nature of communication means that the client and server need to be careful when updating their ideas of highest slotid, and target_highest_slotid.
  - SEQUENCE requests/replies on different slots can be reordered w.r.t. each other.
- RFC5661 does not say what happens to the sequence id for a "new" slot, when the server raises sr_highest_slotid.
  - Should it be initialised to '0' on the server? Probably not, due to corner cases.

How does reordering create problems?

- Client sees incorrect limits:



Thank you