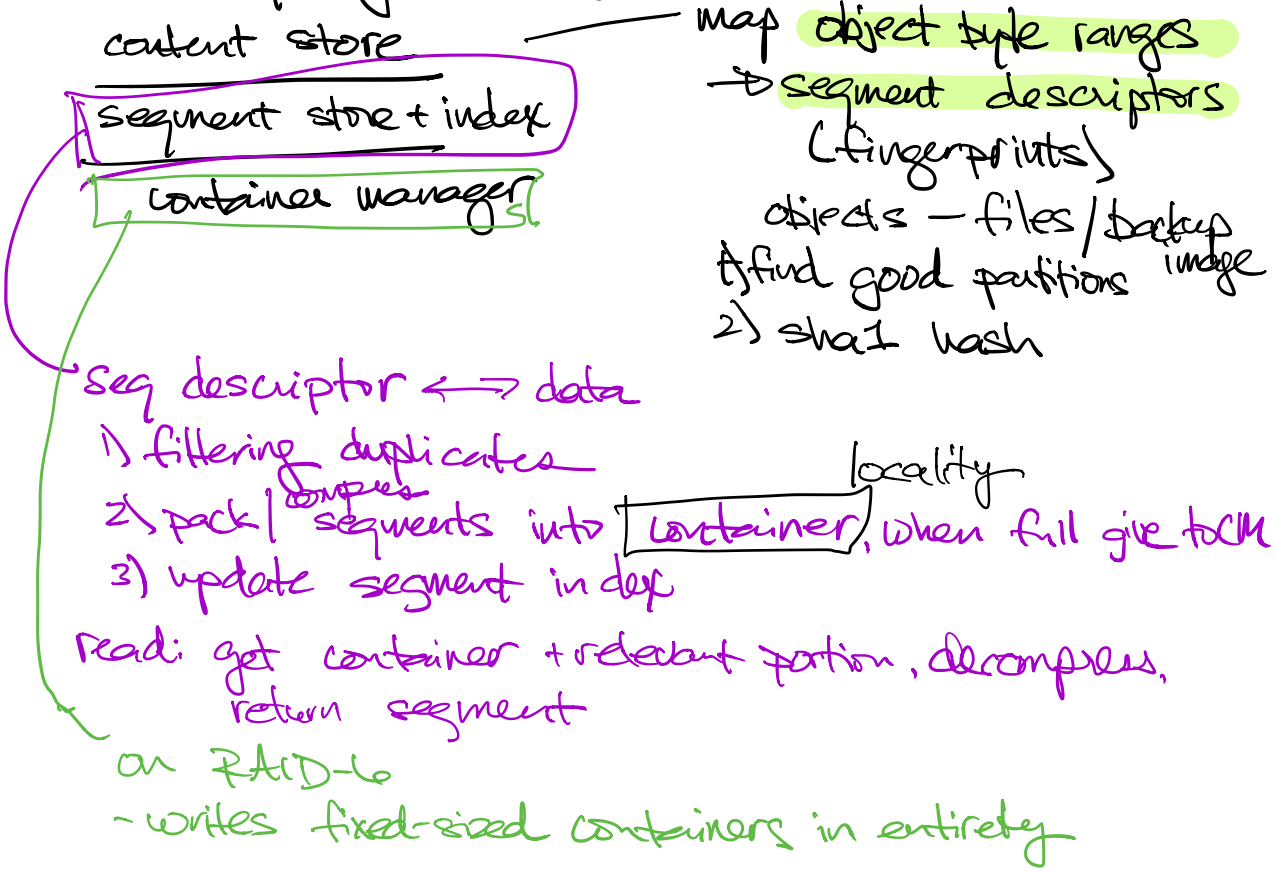
Data Domain Deduplication - 2008

- Reduce storage requirements for data archive
- Improve performance
  1) Bloom filters (summary vector)
  2) Spatial locality (stream informed segment layout for disk accesses)
  3) Locality preserved caching - prefetch segment fingerprints together

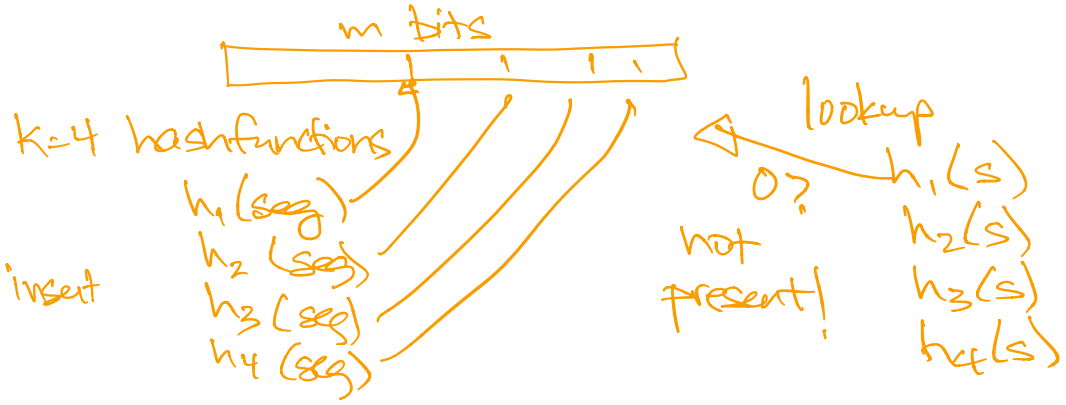
Better layering in system



Techniques

1) Summary vector: avoid going to disk if data doesn't exist (not in index)

Bloom filter - false positives, not false negatives
might have to check index sometimes



2) Stream Informed Layout

Observation? segments tend to occur together in same order (eg. backup same file, or similar files)

When new data contains dup. segment x, high prob that other nearby segments here were near x in other locales

Dedicate container to hold segments in logical order (+ descriptors)

⟹ Fewer I/Os since get all @ once

3) Locality Preserved Caching

Traditional caching of fingerprints for index $
— random, not effective (high miss rate)

Cache all fingerprints from container together
- Replace entire container

Steps on segment write

1) In segment cache? done
2) summary vector? no → new → append to current container
3) yes? check index (no → new)
   - yes → duplicate
     - insert into segment cache
       - all fingerprints for this container (remove LRU)

Best performance results?

→ Most useful?

Table 4: Huge reduction in disk I/O w/ Bloom filter + locality preserved caching