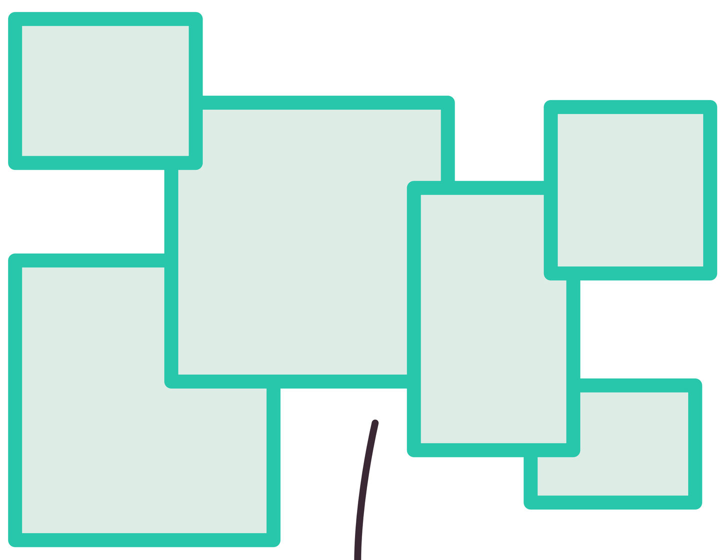
Design of memory hierarchy with NVM for two caching middleware

Shahram Ghandeharizadeh
Sandy Irani
Jenny Lam

how do you design a memory hierarchy for caching?

cached objects have different parameters

time to reobtain · popularity · size



many types of storage technologies to choose from

magnetic disk, NAND Flash, DRAM, PCM, Memristor, STT-RAM, FeRAM

with different characteristics

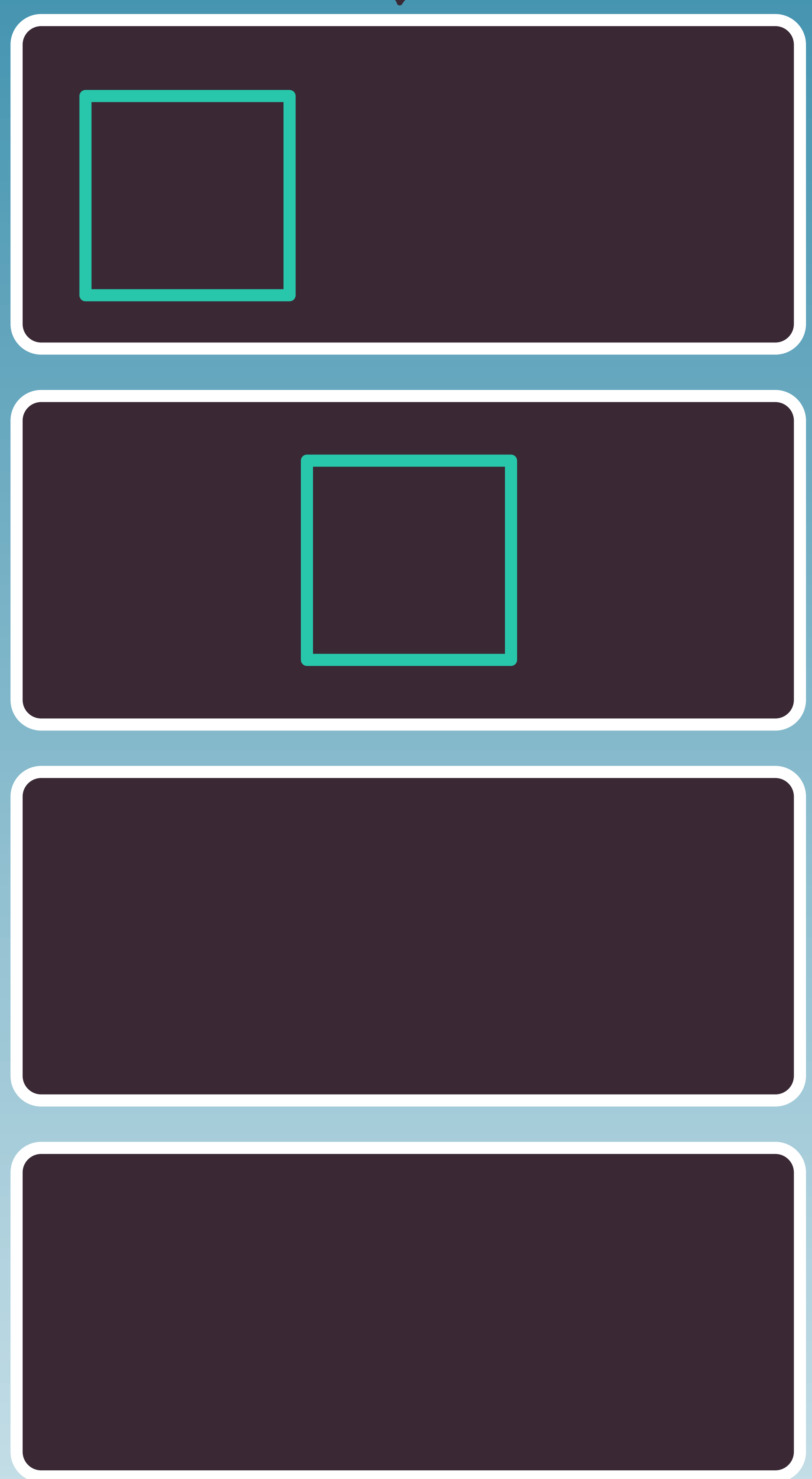
price · read/write performance · rate of failure · volatility

which storage types to use and how much to buy?

many design decisions

is replication across tiers worth it?

which tiers should each object be placed on?



different workloads to accommodate

email · satellite maps · social networking

Step 1: model expected service time per request as a function of placement

contribution to expected service time for an object assigned to a specific group of memory tiers:

time to read the object
size of object
probability of reading that object
bandwidth of fastest tier

time to write object
size of object
probability of a write to that object
bandwidth of slowest tier if writing to tiers in parallel

time to recover from failure
time to read from the running tier
time to write to all tiers that have failed

Step 2: find amount of each memory type and placement to minimize expected service time

cast optimization as multiple-choice knapsack problem

multiple choice
each object has a choice of being assigned to any group of memory tiers

benefit of assigning an object to specific tiers
time savings from storing it on that group of memory tiers rather than not storing it anywhere in the cache

cost of an assignment
purchase cost associated with the amounts of different types of memory that the object will take up

efficient approximation algorithm to solve MCKP
fractional MCKP can be solved by a greedy algorithm in time O(number of objects, times number of memory types) with minimal fractional assignments

Example: host-side cache for mail server

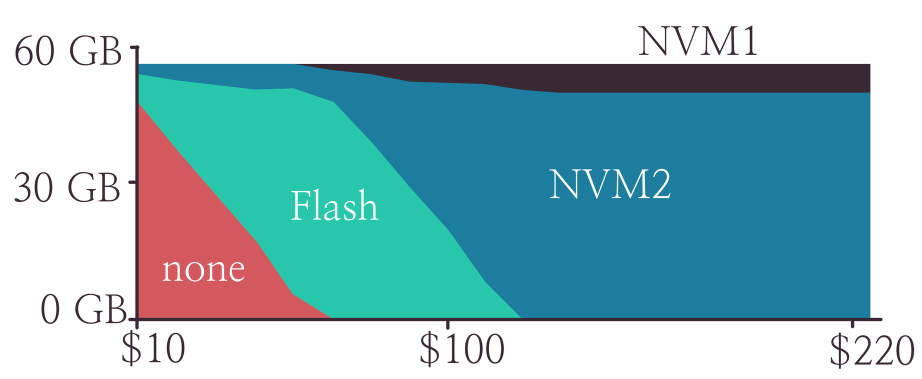
the storage media

	NVM1	NVM2	Flash
read latency	30 ns	70 ns	25 μs
write latency	95 ns	500 ns	200 μs
read bandwidth	10 GB/s	7 GB/s	200 MB/s
write bandwidth	5 GB/s	1 GB/s	100 MB/s
price	\$4/GB	\$2/GB	\$1/GB
MTTF/MTBF+MTTR	2.5 yrs	5 yrs	10 yrs

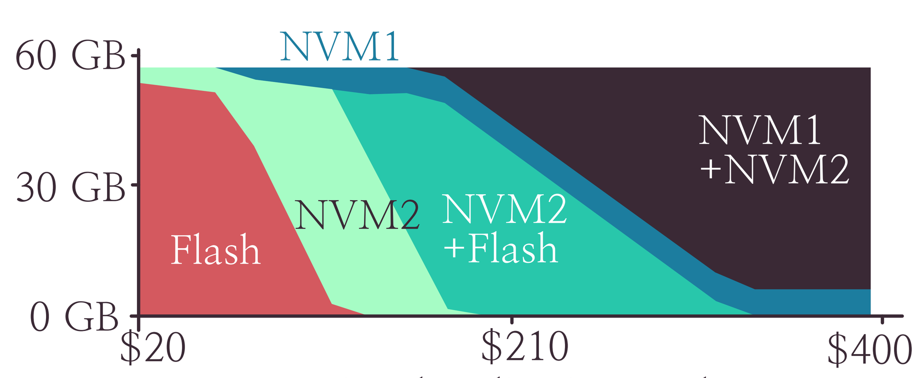
the trace
458 million requests over 18 hours
14.7 million requested blocks
56.25 Gigabytes

Source: Koller and Rangaswami. I/O deduplication: Utilizing content similarity to improve I/O performance. FAST 2010

the design
optimal partitioning of objects as a function of budget



tiering mode (no replication)



optional replication mode

Lessons
optional replication is appropriate with certain failure rates
tiering is superior with slim failure rates

Conclusion
proposed a systematic method for determining the optimal cache configuration given a fixed budget, based on info available

Directions for future research
understand how robust cache design is to changes in workload

Full paper
http://dblab.usc.edu/Users/papers/CacheDesTR2.pdf, 2015.