It's Time to Revisit LRU vs. FIFO

Ohad Eytan[1,2], Danny Harnik[1], Effi Ofer[1],
Roy Friedman[2] and Ronen Kat[1]

July 13, 2020

HotStorage '20

[1] IBM Research
[2] Technion - Israel Institute of Technology

The Essence of Caching

- A fast but relatively small storage location
- Temporarily store items from the "real storage"

The Essence of Caching

- A fast but relatively small storage location
- Temporarily store items from the “real storage”
- Improves **performance** if hit-ratio is high

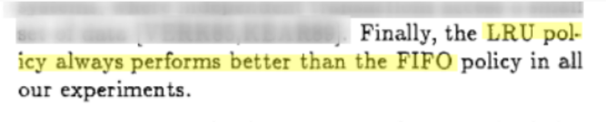
LRU & FIFO

Least Recently Used and First In First Out Policies

- The core component of the cache is the admission/eviction policy
- **FIFO** - holds the items in a queue:
  - ⋆ On a miss: admit new item to the queue and evict the next in line
  - ⋆ On a hit: no update is needed
- **LRU** - holds the items in a list:
  - ⋆ On a miss: add new item to list tail and evict item from list head
  - ⋆ On a hit: move item to the list tail
- Both are simple & efficient

Traditionally: LRU Considered Better

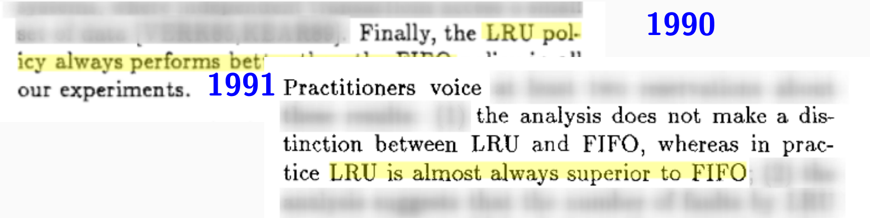
Traditionally: LRU Considered Better



[redacted]. Finally, the LRU policy always performs better than the FIFO policy in all our experiments.

1990

Traditionally: LRU Considered Better



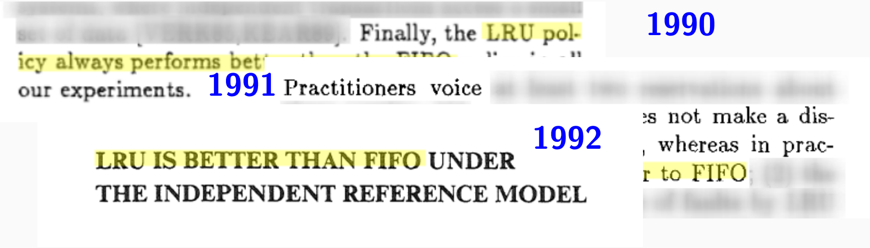
1990

[illegible] Finally, the LRU policy always performs bet[illegible] our experiments.

1991

Practitioners voice [illegible] the analysis does not make a distinction between LRU and FIFO, whereas in practice LRU is almost always superior to FIFO [illegible]

Traditionally: LRU Considered Better



1990

[illegible] Finally, the LRU policy always performs bet [illegible] our experiments.

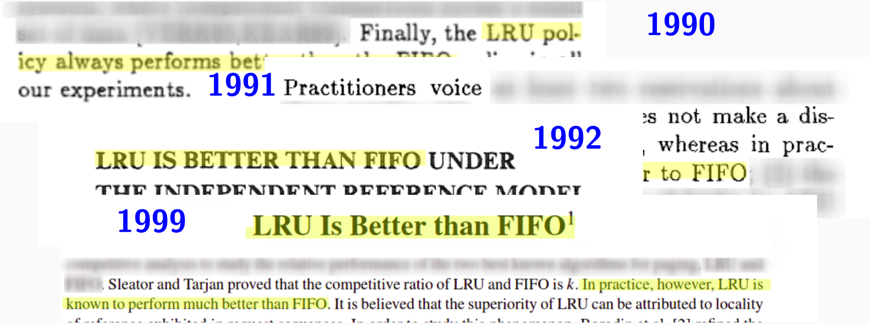
1991

Practitioners voice [illegible] es not make a dis- [illegible], whereas in prac- [illegible] r to FIFO; [illegible]

1992

LRU IS BETTER THAN FIFO UNDER THE INDEPENDENT REFERENCE MODEL

Traditionally: LRU Considered Better



1990

Finally, the LRU policy always performs bet

our experiments.

1991 Practitioners voice

es not make a dis-

, whereas in prac-

r to FIFO

1992

LRU IS BETTER THAN FIFO UNDER

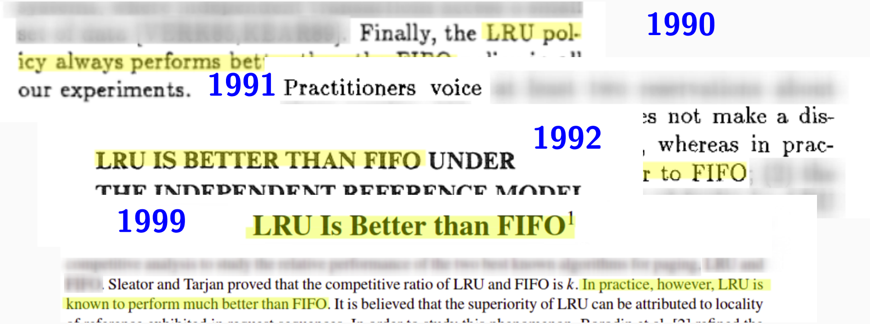
THE INDEPENDENT REFERENCE MODEL

1999

LRU Is Better than FIFO[1]

. Sleator and Tarjan proved that the competitive ratio of LRU and FIFO is k. In practice, however, LRU is known to perform much better than FIFO. It is believed that the superiority of LRU can be attributed to locality of reference exhibited in request sequences. In order to study this phenomenon, Borodin et al. [2] refined the

Traditionally: LRU Considered Better



1990

Finally, the LRU pol-
icy always performs bet
our experiments.

1991

Practitioners voice
es not make a dis-
, whereas in prac-
r to FIFO

1992

LRU IS BETTER THAN FIFO UNDER
THE INDEPENDENT REFERENCE MODEL

1999

LRU Is Better than FIFO[1]

Sleator and Tarjan proved that the competitive ratio of LRU and FIFO is k. In practice, however, LRU is known to perform much better than FIFO. It is believed that the superiority of LRU can be attributed to locality

Does it still hold?

New World

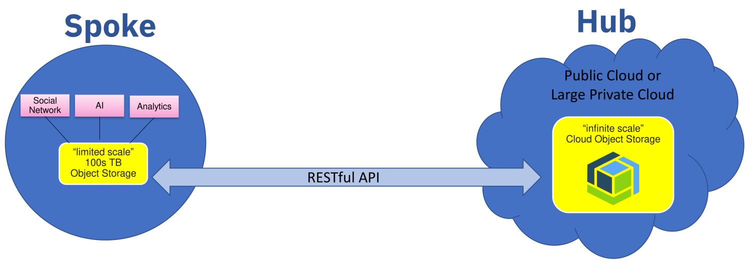
- New workloads:
  - ⋆ Old world: file and block storage
  - ⋆ Today: videos, social networks, big data, machine/deep learning
    - ◦ In particular we are interested in **object storage** (e.g. Amazon S3, IBM COS)

New World

- New workloads:
  - ⋆ Old world: file and block storage
  - ⋆ Today: videos, social networks, big data, machine/deep learning
    - ◦ In particular we are interested in **object storage** (e.g. Amazon S3, IBM COS)
- New scale of data:
  - ⋆ Orders of magnitude higher
  - ⋆ Emergence of cloud storage and persistent storage caches
  - ⋆ Cache metadata can potentially surpass memory

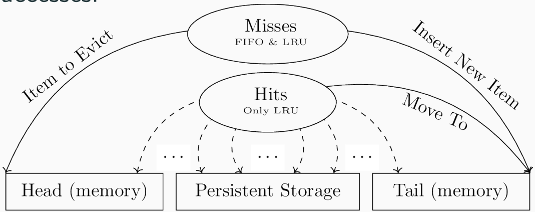
Motivation - Cloud Object Storage

- Data resides on an "infinite scale" remote hub
- Local "limited scale" on a local spoke to improve latency
  - ⋆ Possibly 100s of TBs in size
  - ⋆ Some of the metadata will have to reside on persistent storage


Spoke
Hub
Social Network
AI
Analytics
"limited scale" 100s TB Object Storage
RESTful API
Public Cloud or Large Private Cloud
"infinite scale" Cloud Object Storage

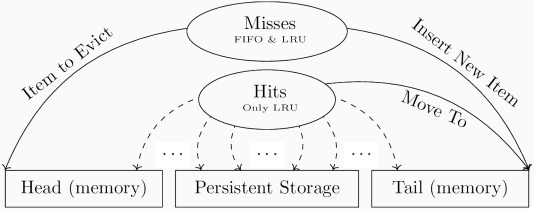
Our Cost Model

- Metadata accesses:



Our Cost Model

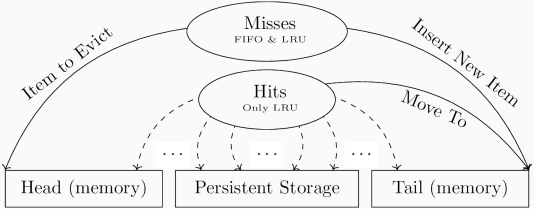
- Metadata accesses:



- Hit rate paints only part of the picture

Our Cost Model

- Metadata accesses:



- Hit rate paints only part of the picture
- We formulated a cost model that accounts also for persistent storage latency:

$$Cost_{LRU} = HR_{LRU} \cdot \overbrace{(\ell_{Cache} + \ell_{CacheMD})}^{data+metadata} + (1 - HR_{LRU}) \cdot \overbrace{\ell_{Remote}}^{data}$$

$$Cost_{FIFO} = HR_{FIFO} \cdot \overbrace{\ell_{Cache}}^{data} + (1 - HR_{FIFO}) \cdot \overbrace{\ell_{Remote}}^{data}$$

IBM Cloud Object Storage Traces

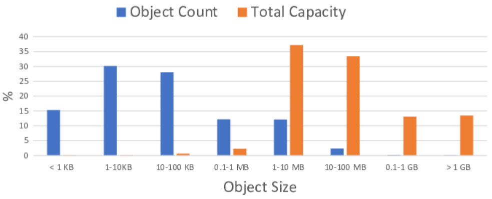
- We collected 99 traces from IBM public Cloud Object Storage service
- Over 850 millions accesses to over 150TB of data

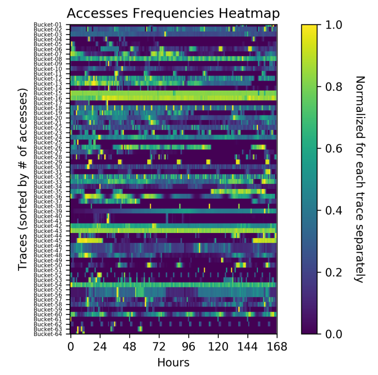
IBM Cloud Object Storage Traces

- We collected 99 traces from IBM public Cloud Object Storage service
- Over 850 millions accesses to over 150TB of data
- Some observations about the IBM traces:

Great variance in object sizes

Great variance in access patterns

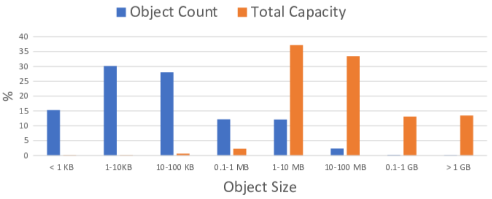




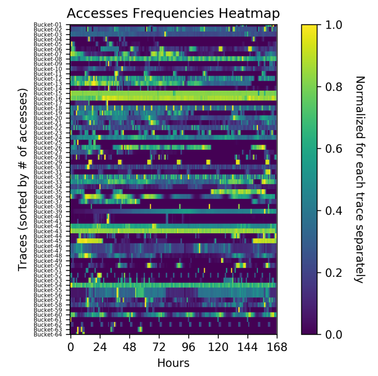
IBM Cloud Object Storage Traces

- We collected 99 traces from IBM public Cloud Object Storage service
- Over 850 millions accesses to over 150TB of data
- Some observations about the IBM traces:

Great variance in object sizes



Great variance in access patterns



- We are publishing the traces and encourage you to use it

Evaluation

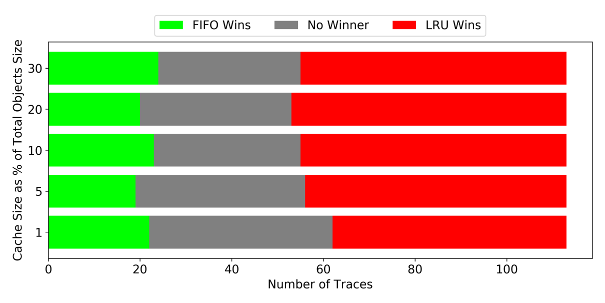
- We evaluated FIFO vs. LRU using 4 sets of traces:

Group	**Traces**	**Accesses**	**Objects**	**Objects Size**
Name	#	*Millions*	*Millions*	*Gigabytes*
MSR	3	68	24	905
SYSTOR	3	235	154	4,538
TPCC	8	94	76	636
IBM COS	99	858	149	161,869

- Tested different cache sizes (as percentage of trace object size)
- Simulated different ratios between latency of cache and remote

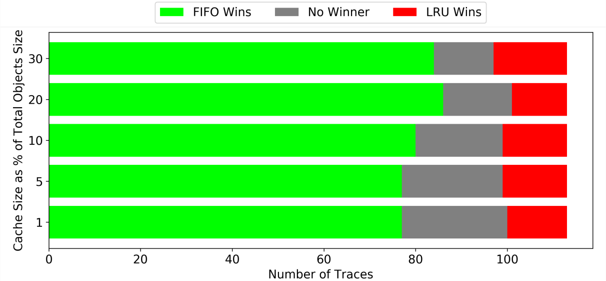
Results

Pure Hit Rate:



Results

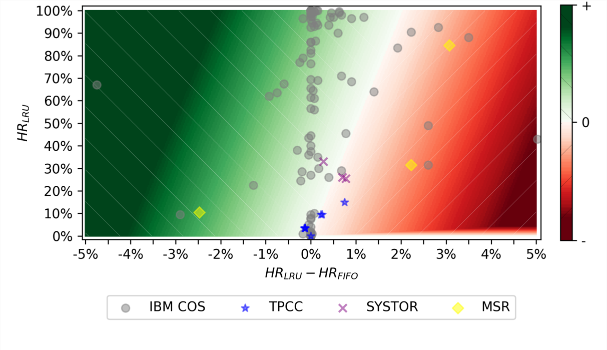
Cost Winners:



$$\ell_{Cache} = 1, \quad \ell_{Remote} = 50$$

Results

Cost Heatmap:



$$\ell_{Cache} = 1, \quad \ell_{Remote} = 50$$

$$Cache\ Size = 30\%$$

Conclusions & Discussion

- It's no longer clear that LRU is a better choice than FIFO
- Hit rate doesn't tell the entire story
- Our IBM COS traces can provide new insights and opportunities for research

Thank You!

Ohad Eytan

ohadey@cs.technion.ac.il

Effi Ofer

effio@il.ibm.com

Danny Harnik

dannyh@il.ibm.com

Roy Friedman

roy@cs.technion.ac.il

Ronen Kat

ronenkat@il.ibm.com