STORAGE DEVELOPER CONFERENCE
SDC EMEA 21
BY Developers FOR Developers

Virtual Conference
June 8, 2021

Evaluating cache performance using cloud storage traces

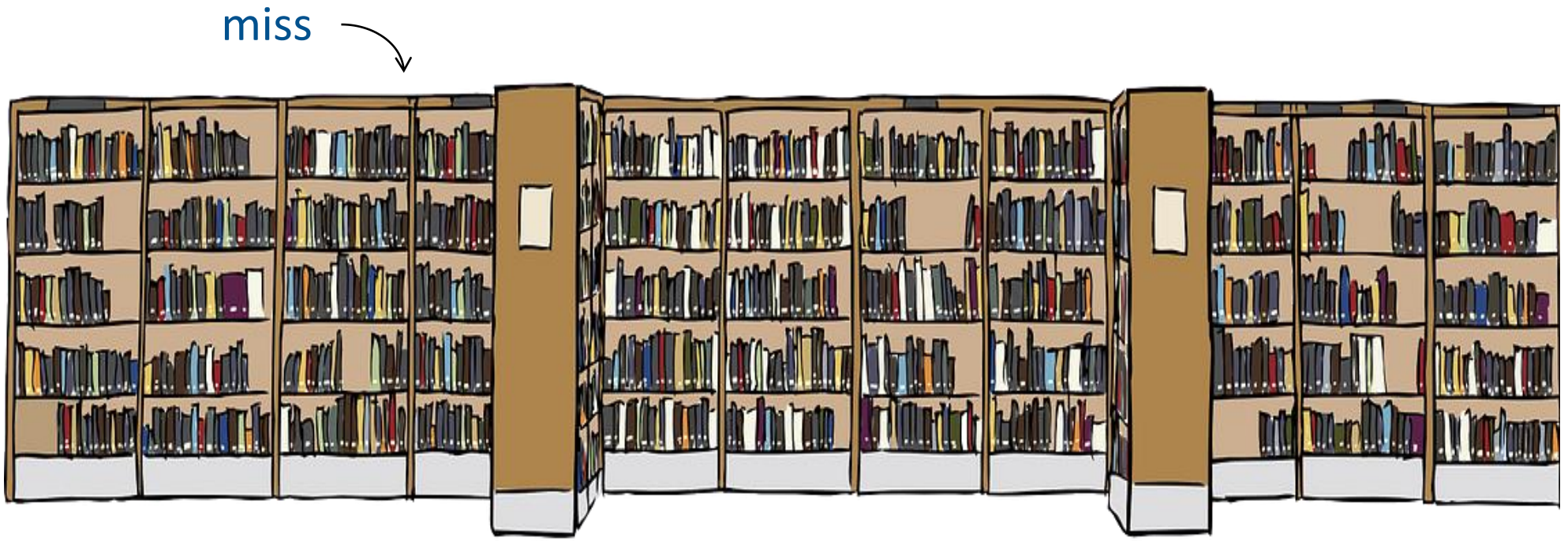
Effi Ofer, IBM Research
effio@il.ibm.com

The Essence of Caching

Speaker Photo Will Be Placed Here

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

The Essence of Caching

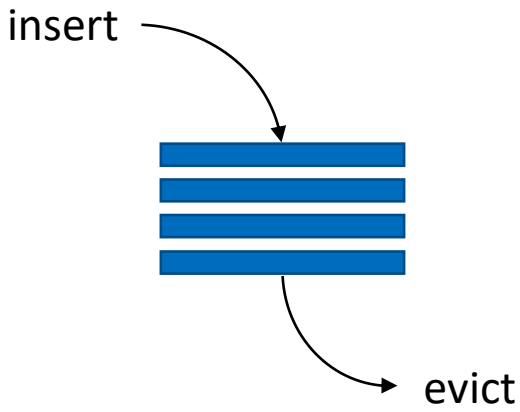


Speaker Photo Will Be Placed Here

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

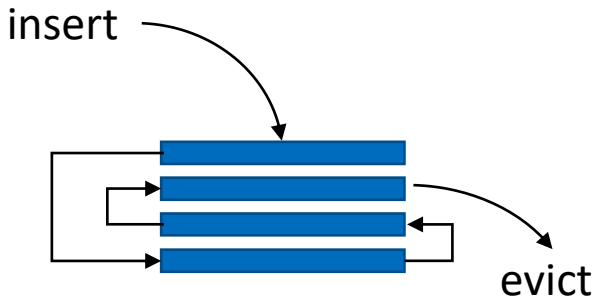
LRU and FIFO – Common Admission / Eviction Policies

- FIFO – items kept in a queue
  - On a miss: admits new item to the queue and evict the next in line
  - On a hit: no update is necessary



Speaker Photo Will Be Placed Here

- LRU – items kept in a list
  - On a miss: adds new item to the list tail and evict item from the list head
  - On a hit: move item to the list tail



Traditionally: LRU is Considered Better

Speaker Photo Will Be Placed Here

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

1990

Traditionally: LRU is Considered Better

Speaker Photo Will Be Placed Here

Finally, the LRU policy always perfor

our experiments.

1990

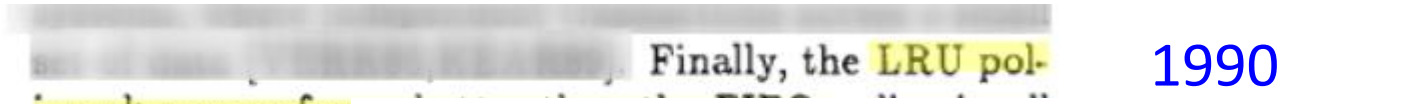
Practitioners voice

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

1991

Traditionally: LRU is Considered Better

Speaker Photo Will Be Placed Here



Finally, the LRU policy always perfor
our experiments.

1990

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

1991

LRU IS BETTER THAN FIFO UNDER THE INDEPENDENT REFERENCE MODEL

1992

Traditionally: LRU is Considered Better

Speaker Photo Will Be Placed Here

1990

Finally, the LRU pol-
icy always perfor
our experiments.

1991

Practitioners voice
the analysis does not make a dis-
tinction between LRU and FIFO, whereas in prac-
tice LRU is almost always superior 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

Traditionally: LRU is Considered Better

Does it still hold true?

Speaker Photo Will Be Placed Here

1990

Finally, the LRU policy always perfor
our experiments.

1991

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

It’s a New World

- New Workloads
  - Old world:
    - file
    - block storage
  - Today:
    - object storage
    - multi-media
    - social networks
    - big data
    - AI

Speaker Photo Will Be Placed Here

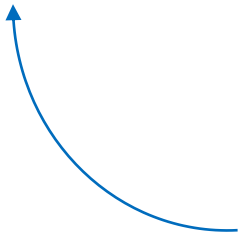
It’s a New World

- New Workloads
  - Old world:
    - file
    - block storage
  - Today:
    - object storage
    - multi-media
    - social networks
    - big data
    - AI
- New Scale of data
  - Orders of magnitude higher
  - cloud storage
  - persistent storage caches

Speaker Photo Will Be Placed Here

It's a New World

- New Workloads
  - Old world:
    - file
    - block storage
  - Today:
    - object storage
    - multi-media
    - social networks
    - big data
    - AI
- New Scale of data
  - Orders of magnitude higher
  - cloud storage
  - persistent storage caches



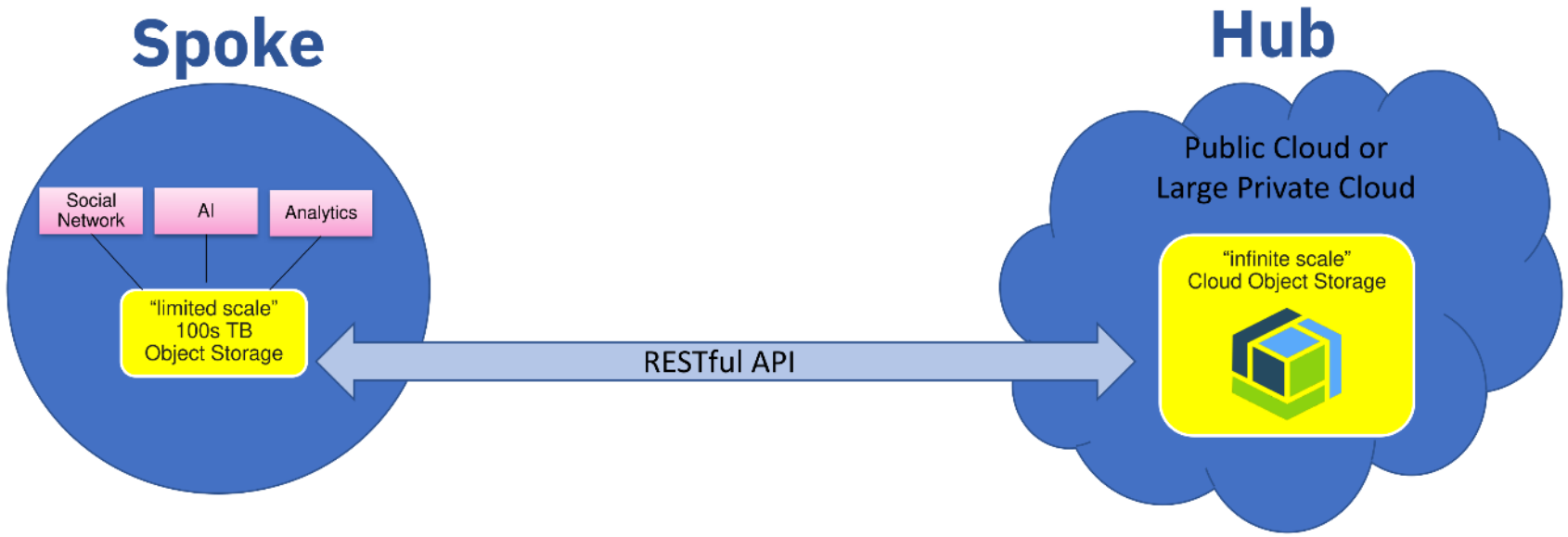
Cache meta-data can potentially surpass memory

Speaker Photo Will Be Placed Here

Designing for Cloud Scale

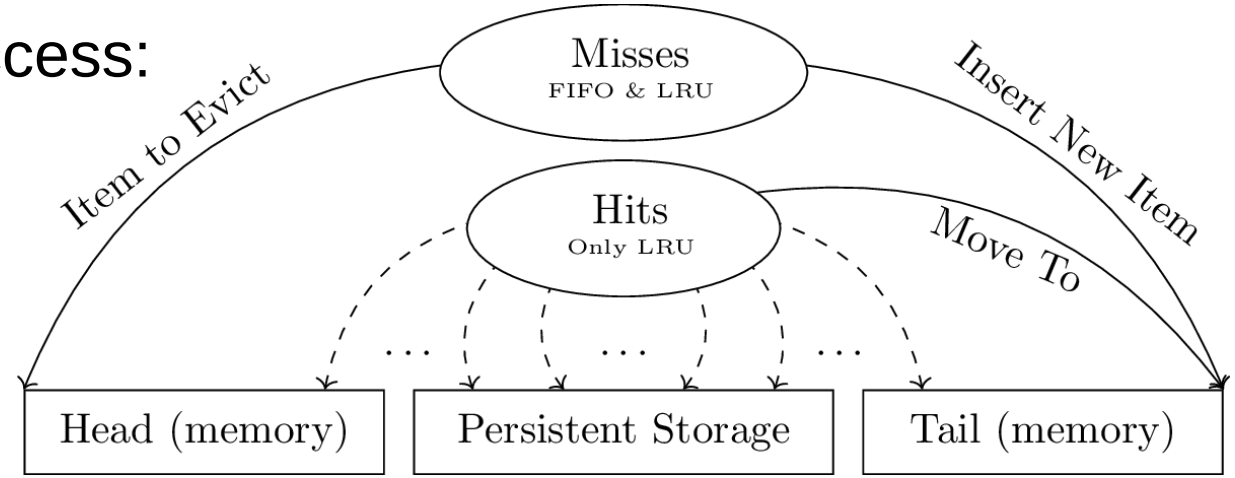
- Data resides on an “infinite scale” remote hub
- Local “limited scale” cache on a local spoke to improve latency
  - Possibly 100s of TBs in size
  - Some of the meta-data will have to reside on persistent storage

Speaker Photo Will Be Placed Here



Designing for Cloud Scale

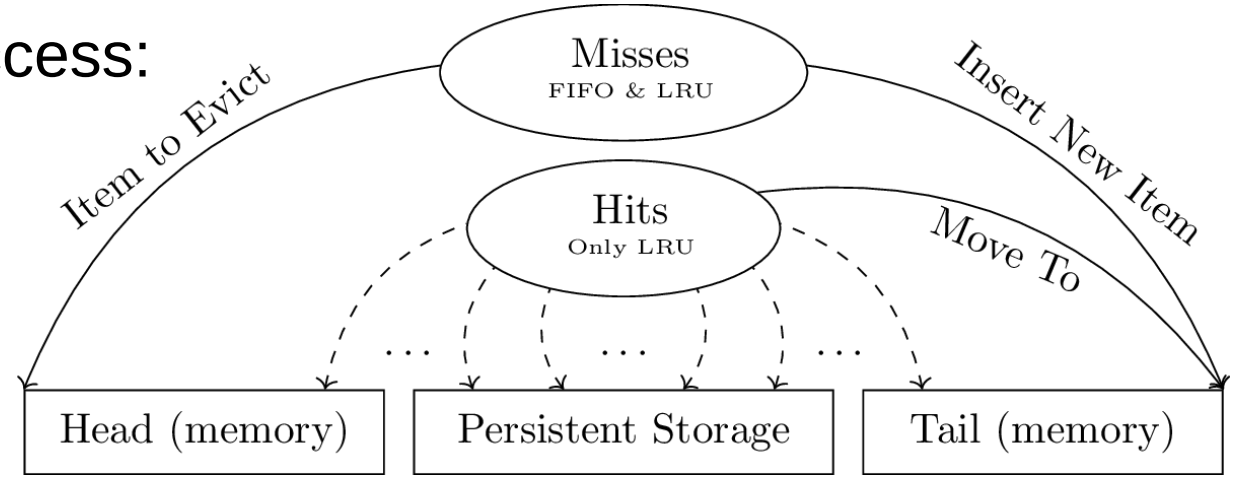
- Meta-data access:



Speaker Photo Will Be Placed Here

Designing for Cloud Scale

- Meta-data access:



Speaker Photo Will Be Placed Here

- Hit rate reveals only part of the picture
- Account for persistent storage latency:

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

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

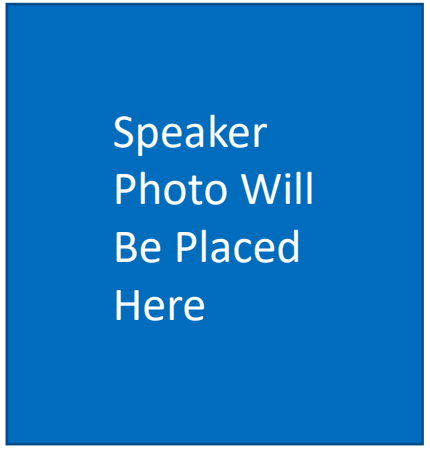
IBM Object Store Traces

- We collected traces from the public IBM Cloud Object Store
- Available on SNIA IOTTA Repository: http://iotta.snia.org/tracetypes/17

Speaker Photo Will Be Placed Here

IBM Object Store Traces

- We collected traces from the public IBM Cloud Object Store
- Available on SNIA IOTTA Repository: http://iotta.snia.org/tracetypes/17
- Each trace contains
  - REST operations issued against a single bucket in IBM Cloud Object Storage
  - During a single week in 2019
- Each trace was selected based on a single criteria:
  - contains some read requests
- Some Statistics
  - 98 traces
  - 88 GB in size
  - 1.6 Billion requests
  - 342 Million unique objects



IBM Object Store Traces

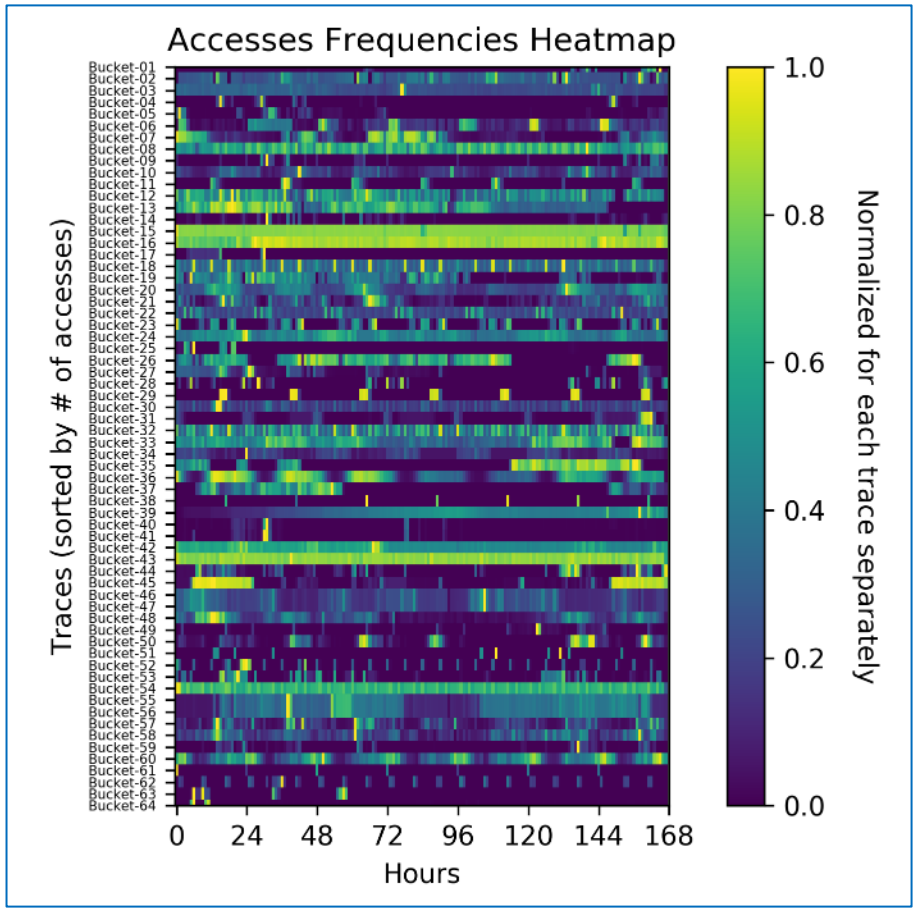
```
1219008 REST.PUT.OBJECT 8d4fcda3d675bac9 1056
1221974 REST.HEAD.OBJECT 39d177fb735ac5df 528
1232437 REST.HEAD.OBJECT 3b8255e0609a700d 1456
1232488 REST.GET.OBJECT 95d363d3fbdc0b03 1168 0 1167
1234545 REST.GET.OBJECT bfc07f9981aa6a5a 528 0 527
1256364 REST.HEAD.OBJECT c27efddbeef2b638 12752
1256491 REST.HEAD.OBJECT 13943e909692962f 9760
1256556 REST.GET.OBJECT 884ba9b0c6d1fe97 23872 0 23871
1256584 REST.HEAD.OBJECT d86b7bfefc63995d 12592
```

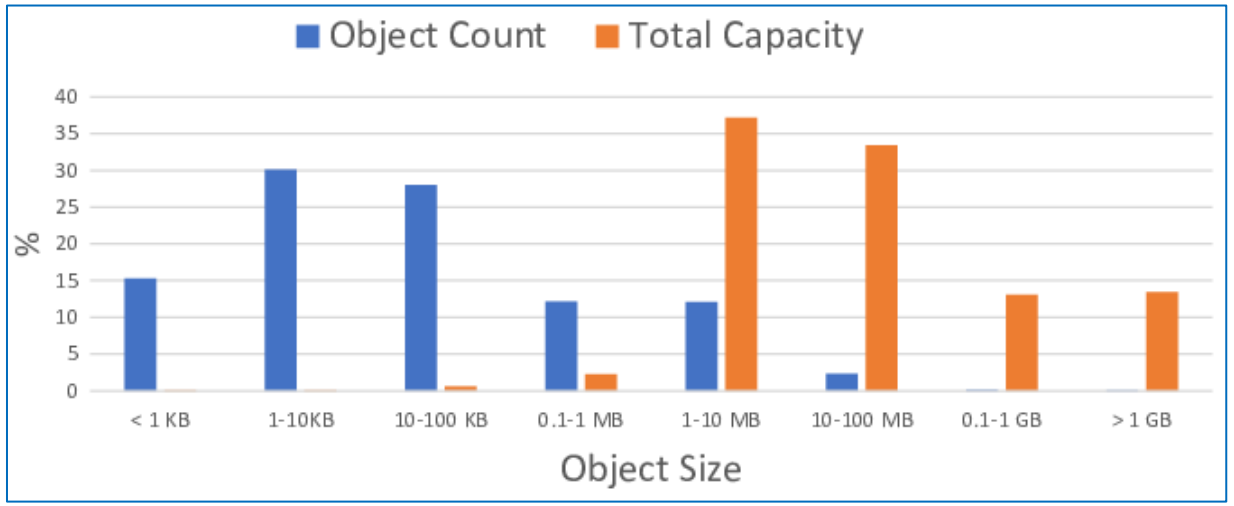
Speaker Photo Will Be Placed Here

- Each trace contains:
  - GET OBJECT
  - PUT OBJECT
  - HEAD OBJECT
  - DELETE OBJECT
- Requests include:
  - Timestamp in ms from the point in time where we began collecting the traces
  - Request type
  - Object ID
  - Size of the object
  - Start and end offset (optional)
- For typical latencies of the IBM Cloud Object Storage see: https://www.ibm.com/cloud/object-storage/resiliency.

IBM Object Store Traces

- Some observations about the data





Speaker Photo Will Be Placed Here

Evaluation

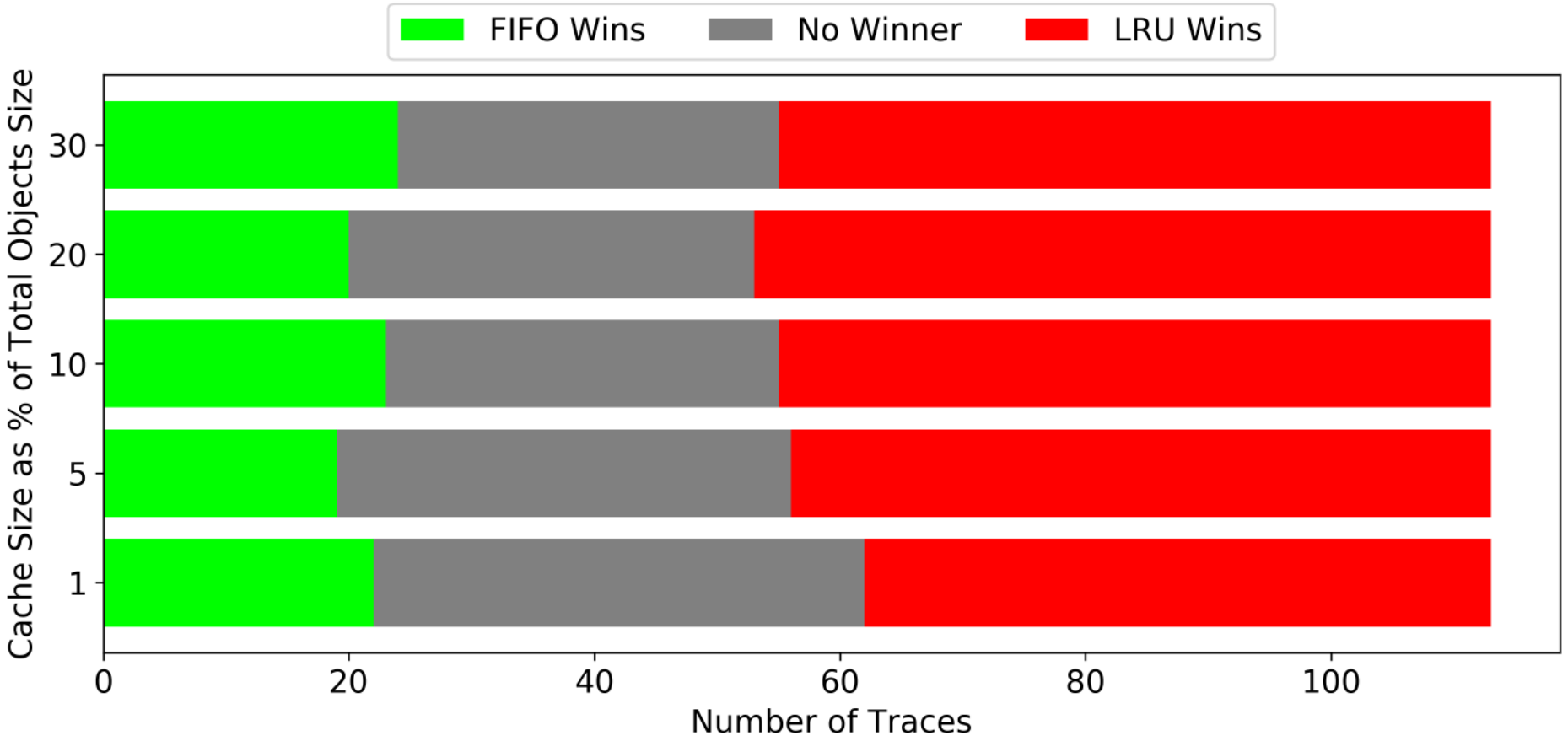
Speaker Photo Will Be Placed Here

- We evaluated FIFO vs LRU using 4 different sets of traces
  - MSR
  - SYSTOR
  - TPCC
  - IBM Object Storage

- Tested different cache size configurations
- Simulated different latencies of both the cache and the remote store

Evaluation

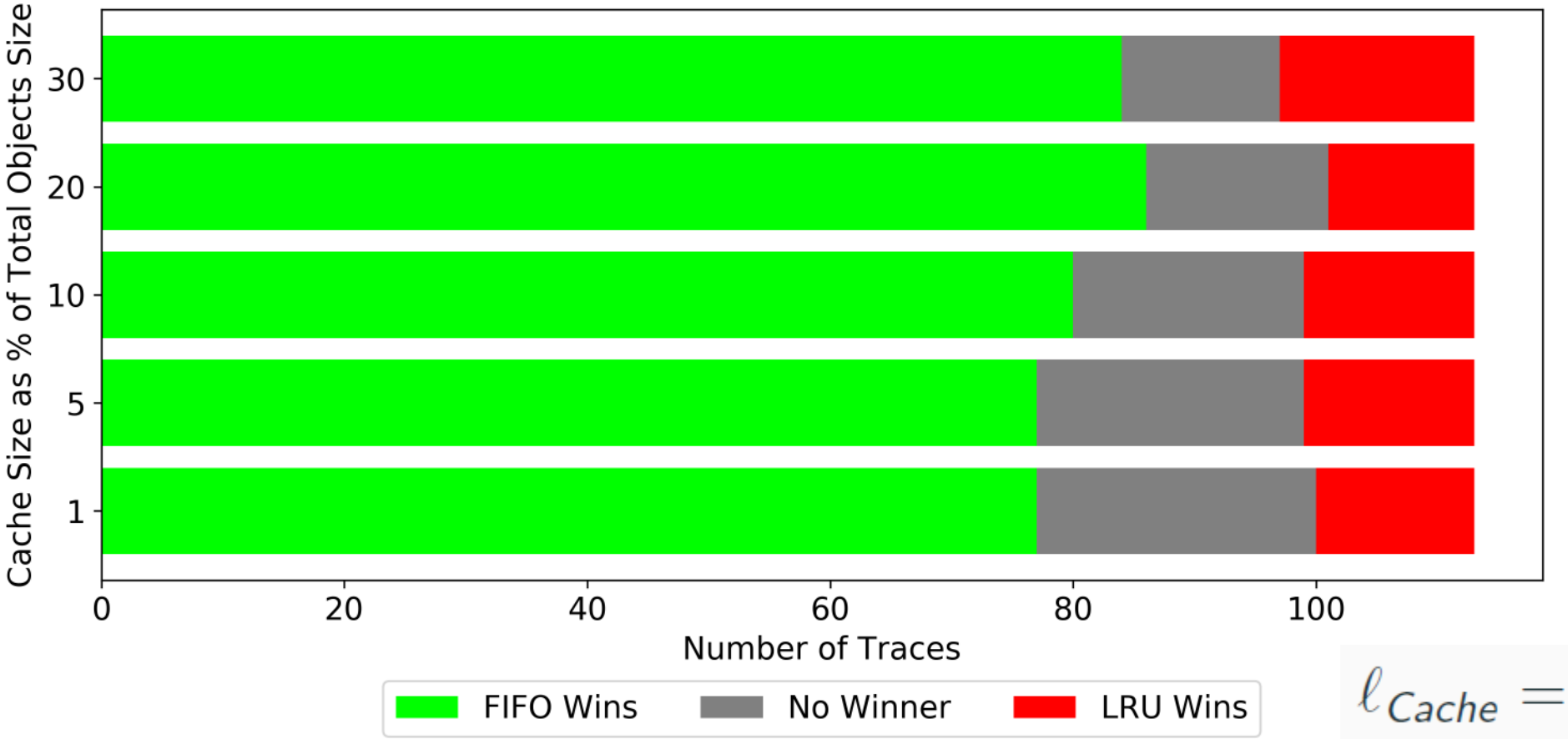
- Pure hit rate



Speaker Photo Will Be Placed Here

Evaluation

- Cost winners



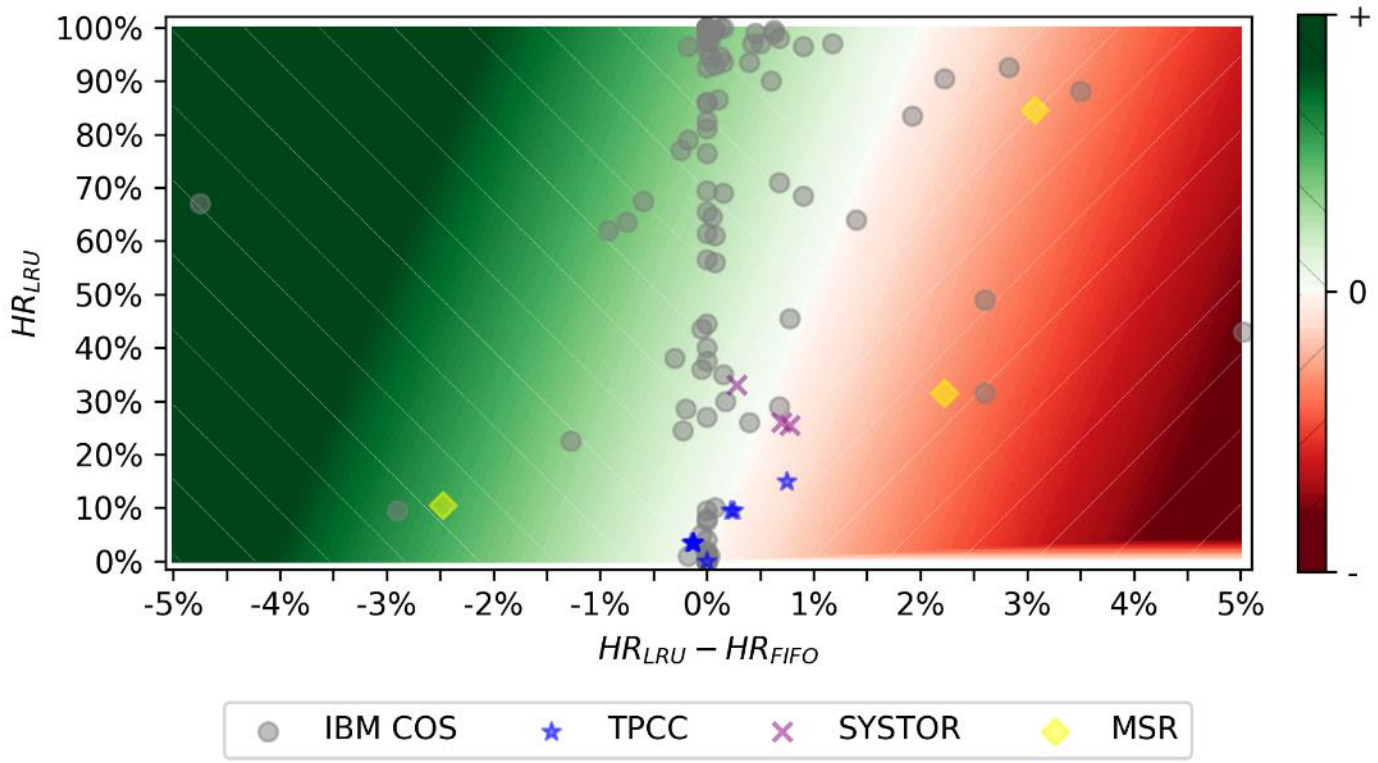
Speaker Photo Will Be Placed Here

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

Evaluation

- Cost heatmap

Speaker Photo Will Be Placed Here



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

$$Cache\ Size = 30\%$$

Discussion

- No longer clear that LRU is better than FIFO
- Hit rate does not tell the entire story
- IBM Object Store traces provide new insight and opportunities for research

Speaker Photo Will Be Placed Here

Thank You!

Ohad Eytan

Effi Ofer

Danny Harnik

Roy Friedman

Ronen Kat

Speaker Photo Will Be Placed Here

Please take a moment to rate this session.

Your feedback is important to us.

