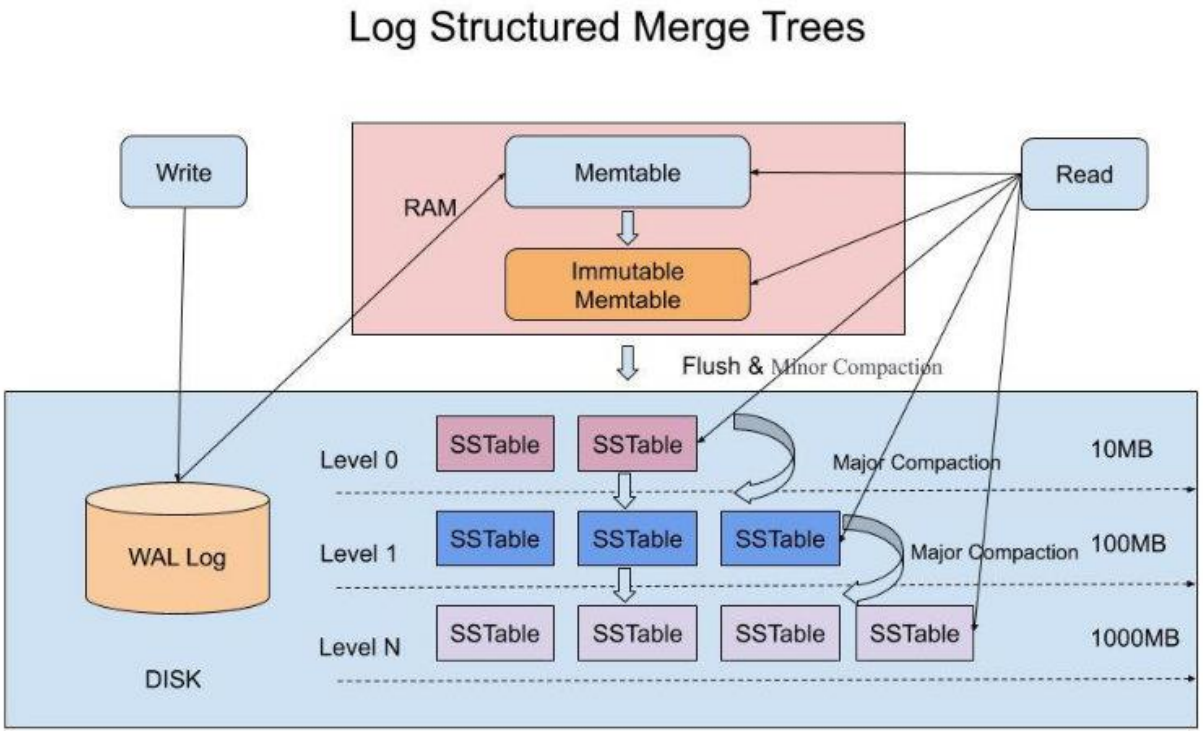
LEAPER: A LEARNED PREFETCHER FOR CACHE INVALIDATION IN LSM-TREE BASED STORAGE ENGINES

Chenwei Xu

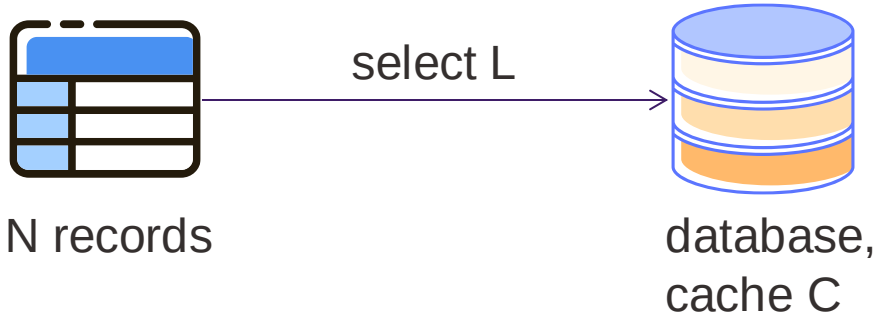
Northwestern | McCORMICK SCHOOL OF ENGINEERING

Recap: What is LSM-Tree



Cache Problems?

- Cache Replacement Problems
  - How to maximize the Probabilty of request records is cache need

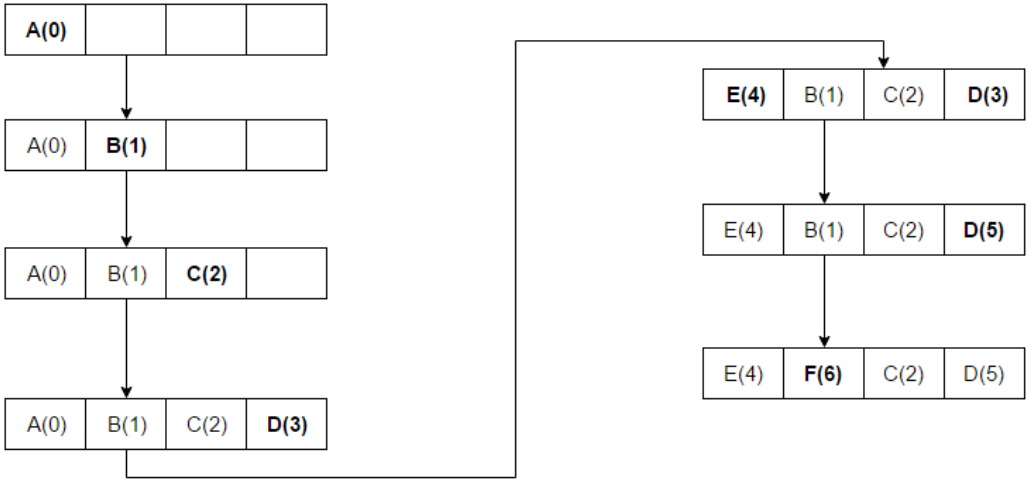


Cache Problems?

- Cache Replacement Problems
  - How to maximize the Probabilty of request records is cache need
- Cache Invalidation Problems
  - How to minimize the invalidated records in cache

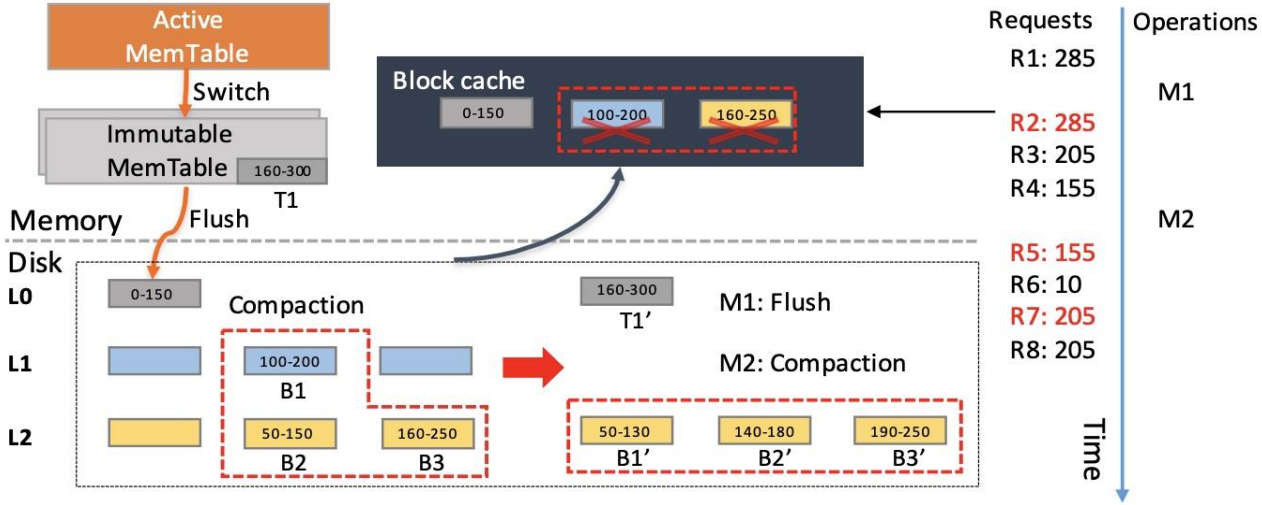
Some solution...

- LRU



Some solution...

- LRU
  - Does not fit for LSM-Tree



LEAPER: LEARNED PREFETCHER

Offline Analysis

Key Range Selection

Instead using per record, they use range of keys(many records)

0(not assessed or not perdicted)
1(assessed or predicted)
1
1

Feature Generation

- Read/Write Arrival Rate
- Prediction Timestamp
- Precursor Arrive Raye

Feature Generation

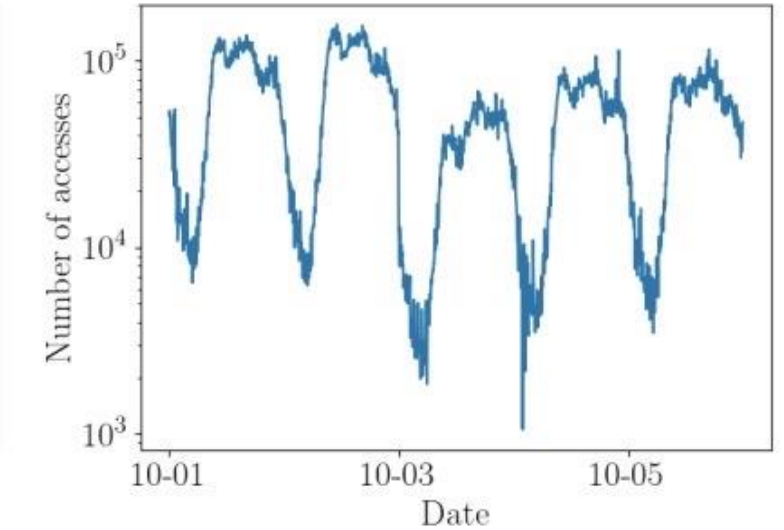
- Read/Write Arrival Rate
- Prediction Timestamp
- Precursor Arrive Rate

0	1	2	3	4	5	6	7	8	9
R			R	R		R	R		
	W		W		W			W	

Feature Generation

- Read/Write Arrival Rate
- Prediction Timestamp
- Precursor Arrive Rate

day	hour	sec



Feature Generation

- Read/Write Arrival Rate
- Prediction Timestamp
- Precursor Arrive Rate
  - Precursor: Another key range
  - n precursors

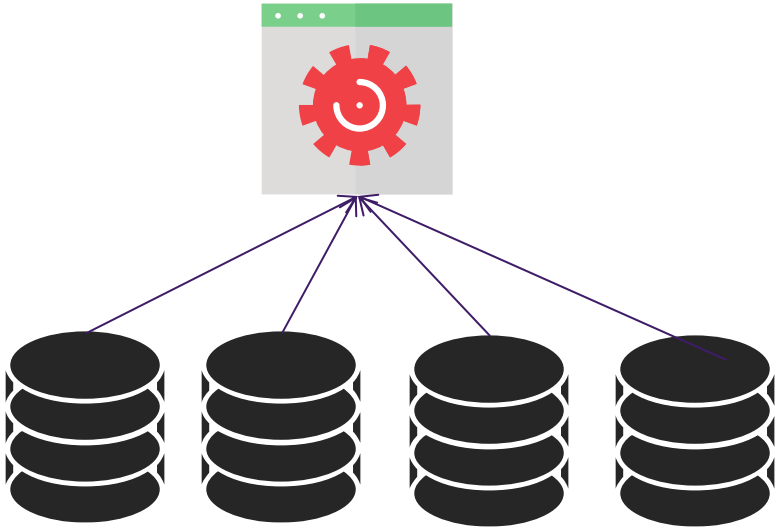
1	2	3	4
10	20	30	40
2	4	6	8
3	9	27	81
5	10	15	20

1	2	3	4	1	2	3	4
10	20	30	40	8	7	6	5
2	4	6	8	2	4	6	8
3	5	7	11	3	4	5	6
9	10	11	12	5	10	15	20

Online Processing

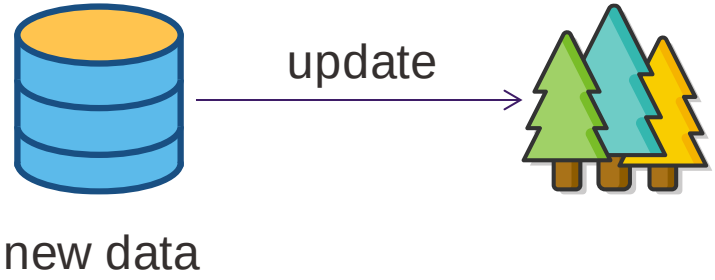
Collect Statistics

- Locking Mechanism
- Sampling
- Atom Operation



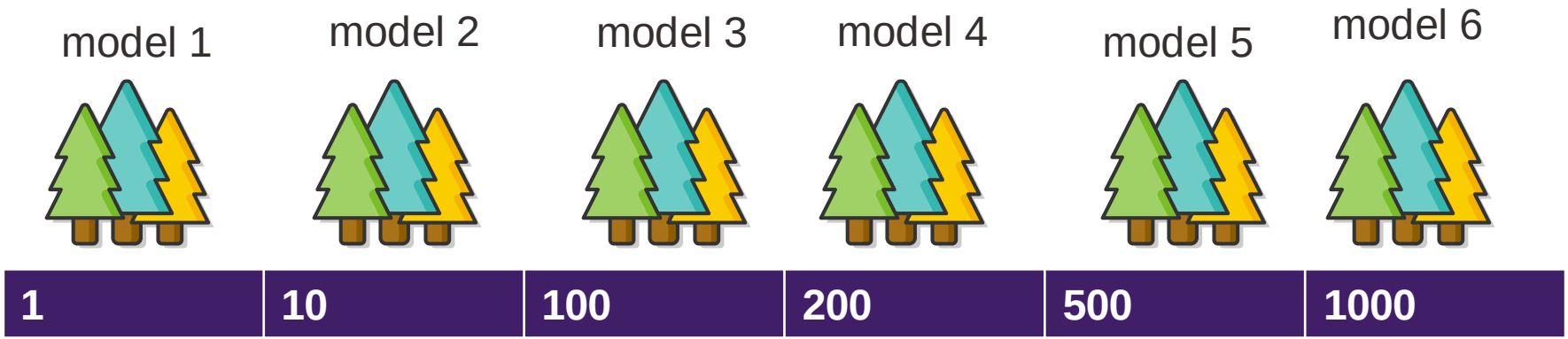
Inference Model

- LightGBM, GBDT



Optimization for Compaction

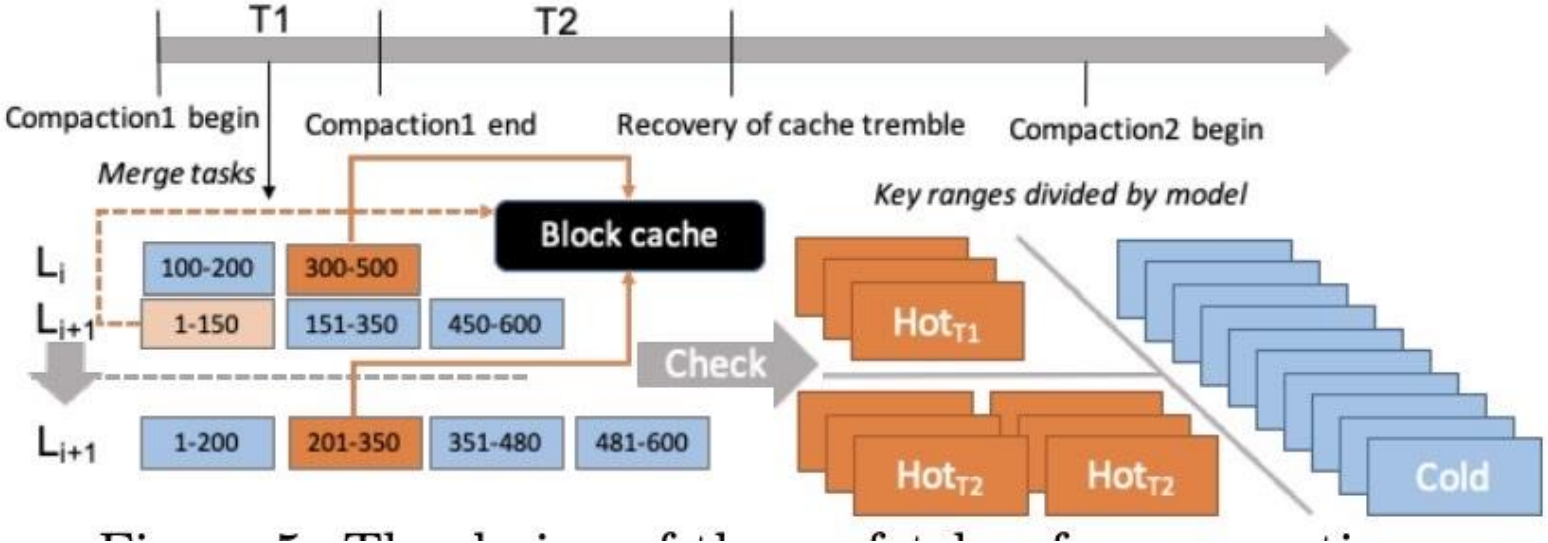
Muti-step Prediction



Two-phase Prefetcher

$hot_{T\,1} = hot_{1t} \cup \cdots \cup hot_{k1t}$,

$hot_{T\,2} = hot_{(k1+1)t} \cup \cdots \cup hot_{(k1+k2)t}$,



Summary

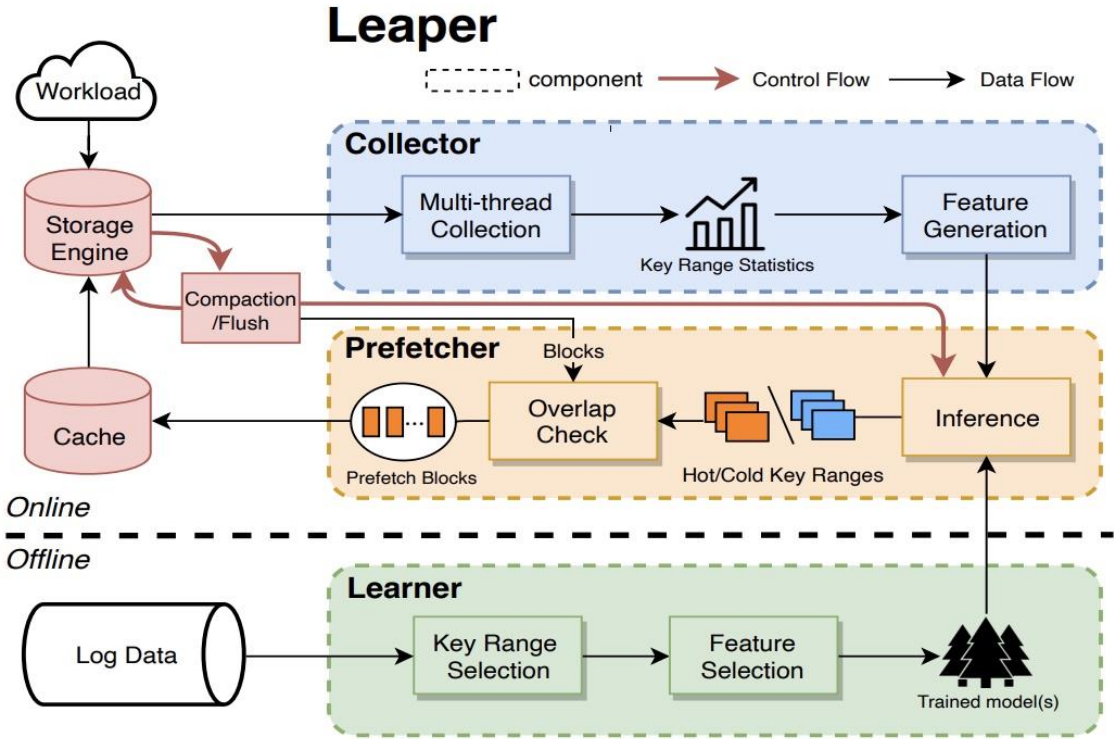


Figure 3: Workflow of LSM-tree storage engine with Leaper.

Thank you!

Northwestern | McCORMICK SCHOOL OF ENGINEERING