SA-LSM: Optimize Data Layout for LSM-tree Based Storage using Survival Analysis

Teng Zhang, Jian Tan, Xin Cai, Jianying Wang, Feifei Li, Jianling Sun

Alibaba Group

Zhejiang University

Backgrounds

- The amount of data grows at an unprecedented rate
- Lower-latency media can account for the lion's share of the total cost for OLTP databases
- Reducing the storage cost on cloud databases has become one of the primary challenges for cloud vendors

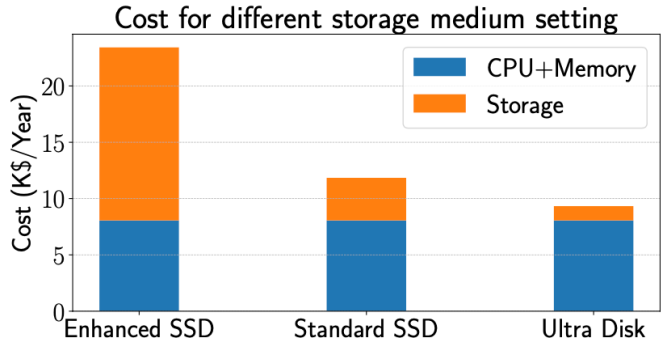


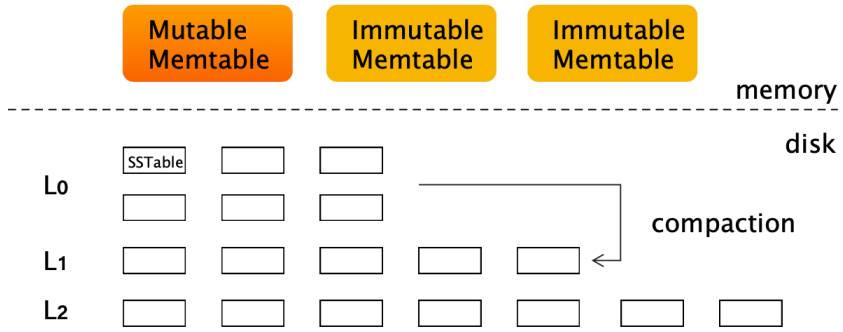
Figure 1: Cost to host a database instance for 1 year using different storage media on Alibaba Cloud ECS *ecs.hfg6.8xlarge* instance with 2 TiB storage [1]; Enhanced SSD (ESSD) storage cost can be more than 2× the cost of CPU and memory

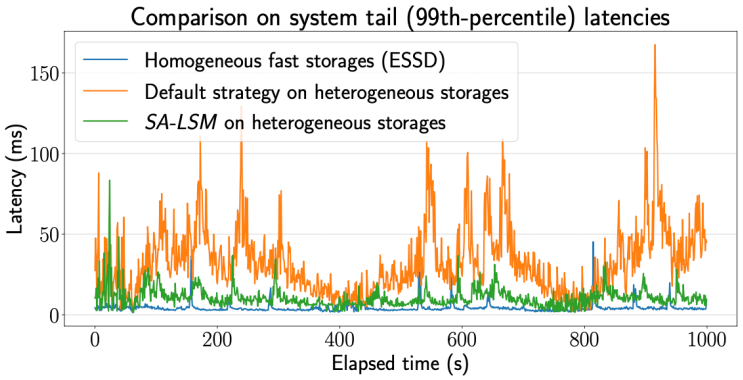
Backgrounds

- LSM-tree introduces multiple layers for heterogeneous storages

- Compaction strategy severely impacts the system performance of LSM-tree
  - Relay 3-day representative workload of Alibaba e-commerce business
  - L0 and L1 on ESSD, L2 on HDD
  - 99th latency on heterogenous storages is 7 times longer than homogeneous storage
  - SA-LSM is almost close to the performance using homogeneous storage

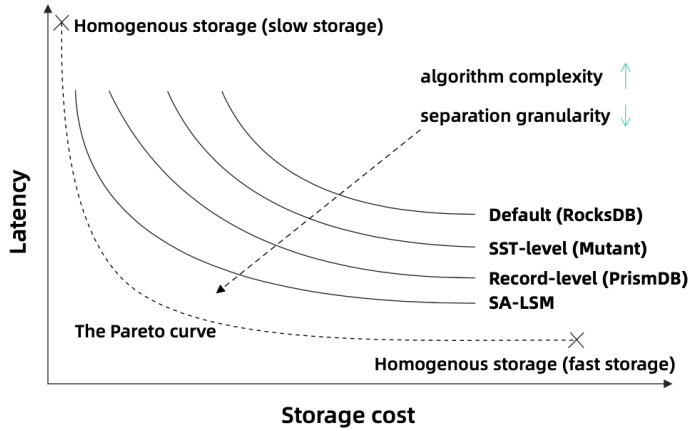
- Heterogeneous storages are cost efficient but only effective if cold and hot data can be well separated





Backgrounds

- LSM- tree with heterogeneous storages
  - Storage cost vs. system performance
  - Computation cost vs. algorithm precision

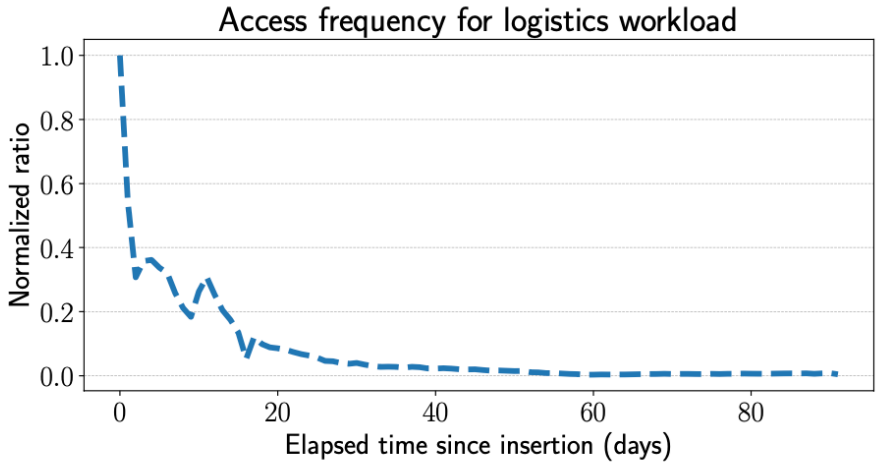


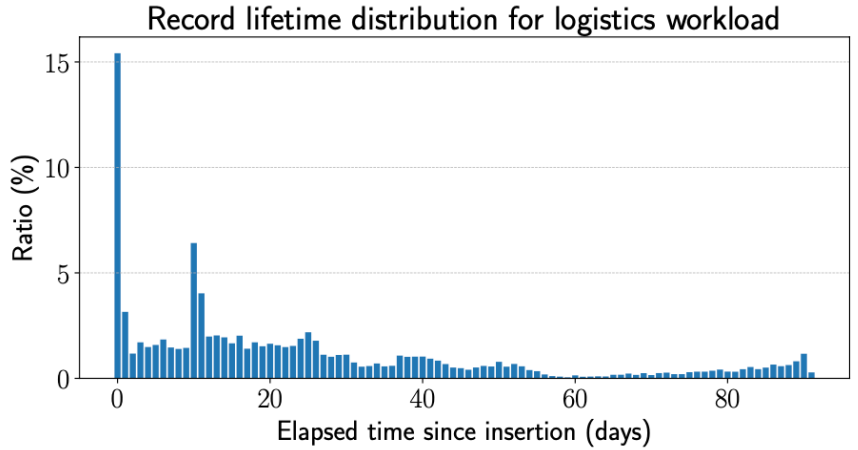
	Granularity	*Separation algorithm*	*Promotion*	*Demotion*	*Trigger time*	*Integration with DB kernel*
RocksDB [19]	SSTable	Random	No	Yes	Compaction	Coupled
Mutant [59]	SSTable	Exponential smoothing	Yes	Yes	Compaction	Coupled
PrismDB [46]	Record	LRU	Yes	Yes	Compaction	Coupled
SA-LSM	Record	Survival analysis	No	Yes	Active trigger	Decoupled

Promotion is the process of moving data from lower layers to higher layers; demotion represents the opposite direction.

Backgrounds

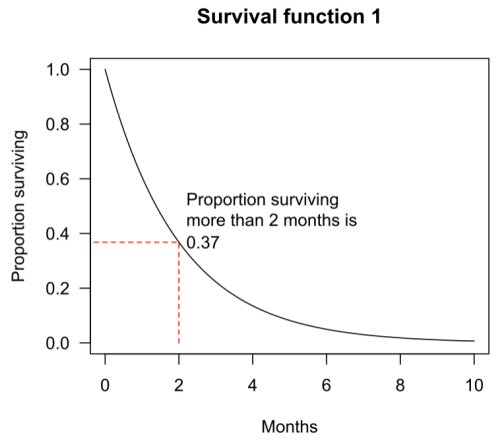
- The popularities of the data records tend to decrease over time
  - Track 50K data records and investigate their popularities over time in 90 days
  - The overall popularity decreases to less than 1% after 50 days and remains at a low level afterwards
- The distribution of lifetime is long-tailed, simply using a fixed time threshold is difficult to separate the cold and hot data
  - Lifetime: the interval between the creation time and the last access time point





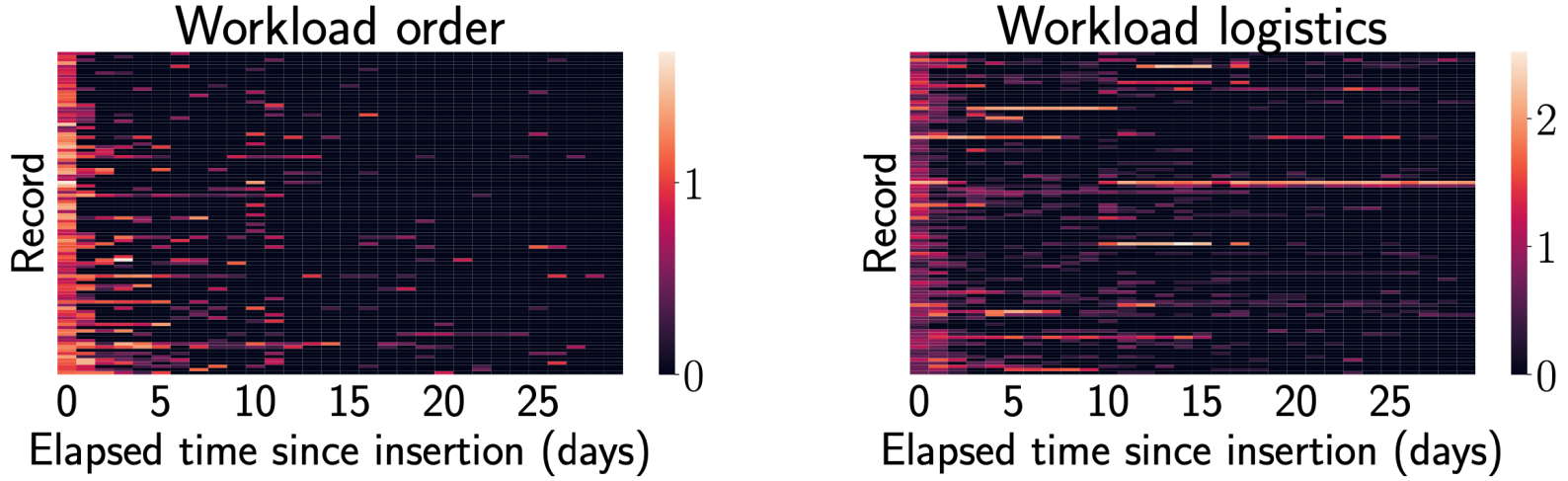
Backgrounds

- Survival Analysis
  - Commonly used in clinical studies, where the time-to-event prediction is usually on the occurrence of a naturally observed end point of interest
    - e.g., relapse or death for patients
  - Some data points do not occur within the observation window (censored data)
  - Survival function: $S(t)=Pr(T \geq t)$
    - Gives the probability that an object of interest will survive pass time t
    - T is a nonnegative random variable denoting the time to the event of interest
  - Models
    - Cox, RSF, deep learning based methods, …



Challenges

- Granularity to separate hot and cold data
  - Dynamic and complex access pattern in an SSTable
  - Simply compacting the whole data on an SSTable can incur unexpected accesses to the cold tiers



(a) Heat map for *order*

(b) Heat map for *logistics*

Challenges

- Censored data due to an limited observation window
  - Inaccuracy and bias could be incurred in the prediction without properly handling these censored events
- Non-intrusive algorithm service deployment
  - Avoid resource contentions with query processing
- Targeting workloads (Archival Write and Point Read Intensive ($AWPI$))
  - The popularities of the data records often keep decreasing over time
  - The access patterns of records can be obtained from the SQL log

Contributions

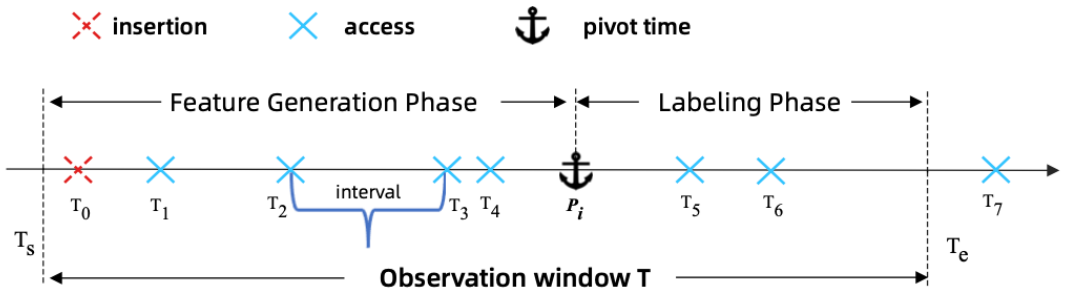
- We revisit the current design of LSM-tree for heterogeneous storages.
- We propose ***SA-LSM***, based on survival analysis, to identify the cold data as a time-to-event prediction problem with censored data.
- We implement a light weight communication protocol between the ***SA-LSM*** external service and the database kernel. We implement SA-LSM for X-Engine, a commercial-strength open-source LSM-tree storage

Cast Cold Data Identification as a Survival Analysis problem

- Objective: rank the records using the predicted access events

- Two-step sampling for pivot selection to avoid bias
  - A sequence of access time points $\{T_1, T_2, \cdots, Tk\}$ within the observation window for data record i.
  - Select an index uniformly from [1, 2, $\cdots$, k] without replacement as ξ
  - Sample a time point uniformly at random on $[T_{\xi-1}, T_\xi]$ as P

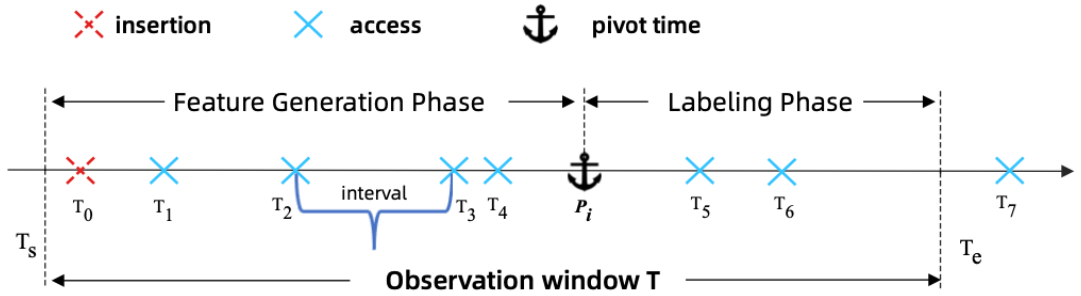
- Random pivots split the observation window into two phases for each data record



Cast Cold Data Identification as a Survival Analysis problem

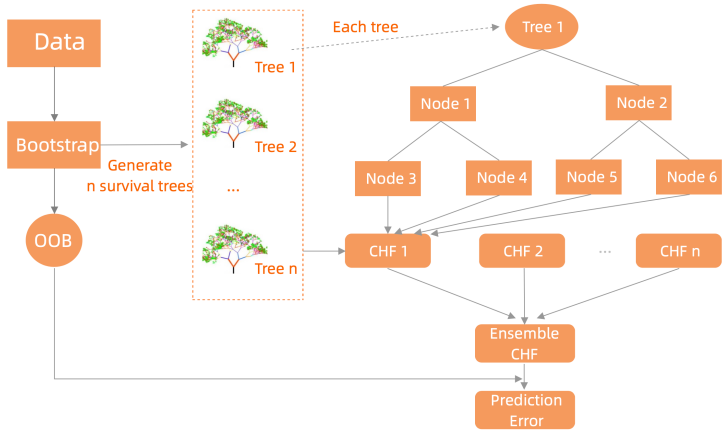
- Label generation
  - τ-event. A τ-event of a data record i is the first access at time $T\nu$ after the pivot time P such that the interval $T_\nu - T_{\nu-1}$ between this access and its immediate previous one is less than τ . The corresponding event time e_i is defined to be $T_\nu - P$.
  - e.g., T_6 - P_i for pivot P_i if $T_5 - T_4 > \tau$ and $T_6 - T_5 < \tau$
- Define the label (y_i , δ_i) for an event of data record i
  - δ_i is a binary event indicator
  - y_i is the label time equal to the minimum of the event time and the interval length of the labeling phase

$$y_i = \begin{cases} e_i & \text{if } \delta_i = 1 \\ T_e - P_i & \text{if } \delta_i = 0 \end{cases}$$



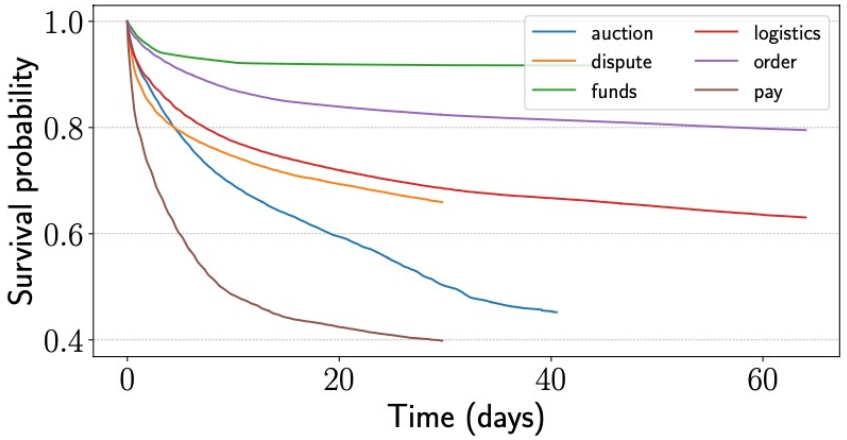
Cast Cold Data Identification as a Survival Analysis problem

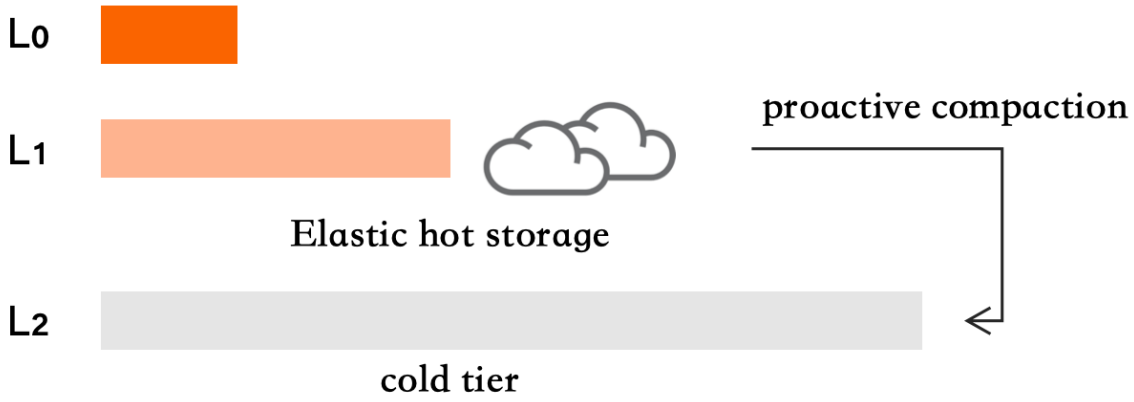
- Feature selection
  - Access features
    - Last access interval (the interval between the pivot and the previous access)
    - Look backward to consecutively select another ξ (e.g., 7) number of access intervals
    - Time difference between the insertion time and pivot time, etc.
  - Semantic features
    - Total numbers of UPDATE and SELECT operations per day as well as the individual numbers on each of the columns
- Survival model
  - Random Survival Forest model (RSF)



SA-LSM in Practice

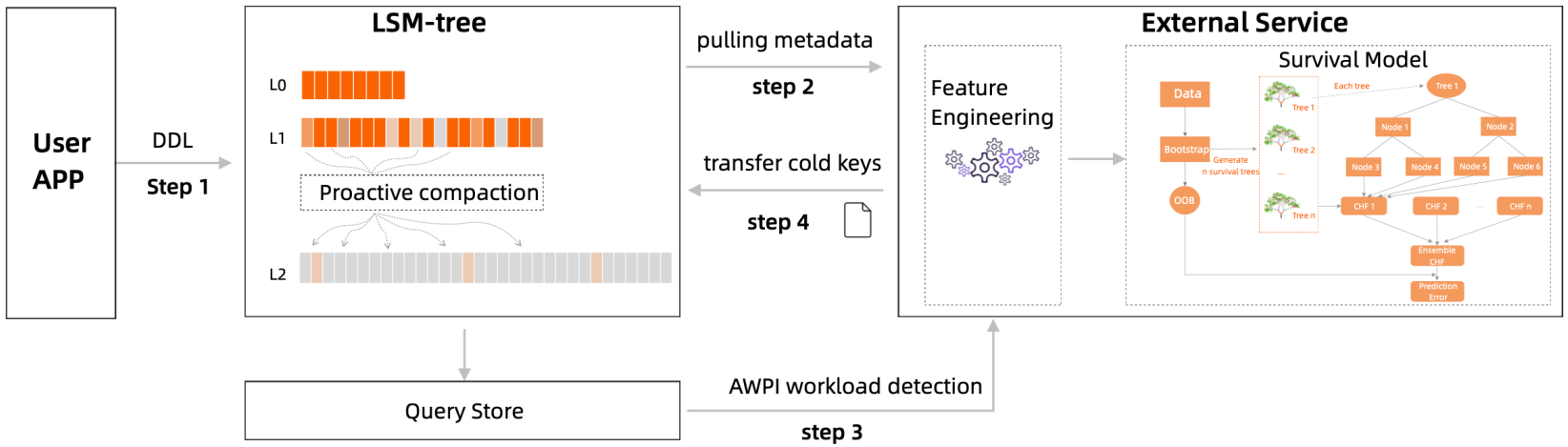
- Proactive vs. Passive Compactions
  - Passive compaction is triggered by the predefined size threshold of L_1
  - Proactive compaction adjust the size of the LSM- tree L_1 layer tailored to individual workloads based on the algorithm results
  - Survival probability curves of read-world workloads imply a dedicated survival model for each workload to capture its unique characteristics





SA-LSM in Practice

- The workflow of SA-LSM
  - Use a DDL to enable proactive compaction for a individual table
  - The external service checks the information_schema for meta data
  - Fetches the log data, trains survival models and identify the keys for the cold data
  - Transfers the results and trigger compaction



Experiments

- Testbed
  - Intel Xeon Platinum 8163 2.5 GHz 20-core CPUs
  - 88 GB Samsung DDR4-2666 main memory
  - Linux 5.10.23
  - Hot storage: Alibaba Cloud ESSD (L_0, L_1)
  - Cold storage: RAID 0 consisting of 8 HDDs (L_2)
- Workloads
  - Preload 20 days' traces of the data records
  - Manually trigger compactions between L_1 and L_2 based on different algorithms
  - Replay the next three days' traces to benchmark the system performance
  - The block cache is set to 30% of the total data size

	#traces	#records	r/w ratio	#samples	censored %
AUCTION	95 million	754 k	3.7	4,081 k	55.2%
DISPUTE	68 million	192 k	42.5	963 k	71.6%
FUNDS	41 million	6,295 k	1.9	32,780 k	91.8%
LOGISTICS	173 million	2,826 k	16.8	18,732 k	67.8%
ORDER	117 million	4,505 k	3.6	28,570 k	81.9%
PAY	124 million	1,736 k	13.8	10,017 k	45.0%

Experiments

- Methods
  - Rule-based algorithms: LRU, LFU, LIRS, LRU-k
  - ML-based algorithm: GBDT, ANN
  - Survival analysis based algorithm: Cox, **RSF**, DeepSurv, CoxTime

- Metric
  - c-index
    - most commonly used metric for survival models

$$\hat{c} = \frac{1}{M} \sum_{i:\delta_i=1} \sum_{j:y_i<y_j} I\left[\hat{S}(y_j|X_j) > \hat{S}(y_i|X_i)\right]$$

  - cold data false identification rate
    - FP for incorrectly predicted cold data
    - TP for correctly predicted cold data

$$cold\_fir = \frac{FP}{FP + TP}$$

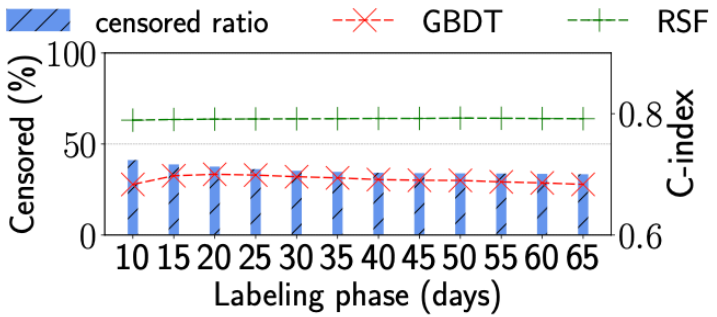
Experiments

- Algorithms performance
  - Random Survival Forest (RSF) is the most robust survival analysis based method, it outperforms GBDT by **8.9% ~ 30.3%** on c-index metric

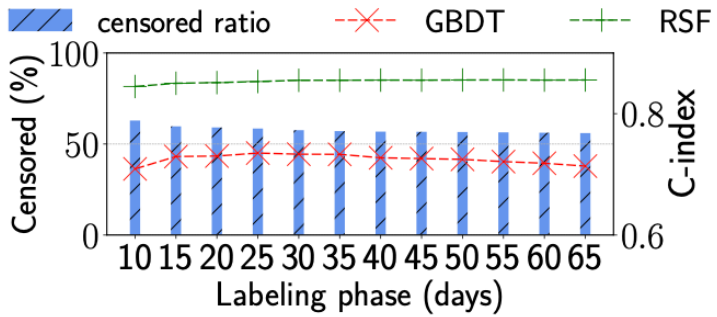
Workload	*auction*	*dispute*	*funds*	*logistics*	*order*	*pay*
ANN	0.701	0.695	0.732	0.715	0.701	0.681
ANN (shared)	0.627	0.659	0.725	0.702	0.681	0.651
GBDT	0.750	0.751	0.758	0.747	0.752	0.722
Cox	0.673	0.789	0.918	0.805	0.819	0.747
DeepSurv	0.717	0.809	0.985	0.825	0.843	0.782
CoxTime	0.761	0.824	0.976	0.794	0.887	**0.865**
CoxCC	0.728	0.810	0.983	0.825	0.838	0.794
RSF (v.s. GBDT)	**0.817 (+8.9%)**	**0.864 (+15.0%)**	**0.988 (+30.3%)**	**0.872 (+16.7%)**	**0.909 (+20.9%)**	0.842 (+16.6%)

Experiments

- Impacts of the observation window
  - the τ -event has a high probability to be still absent in a long observation window



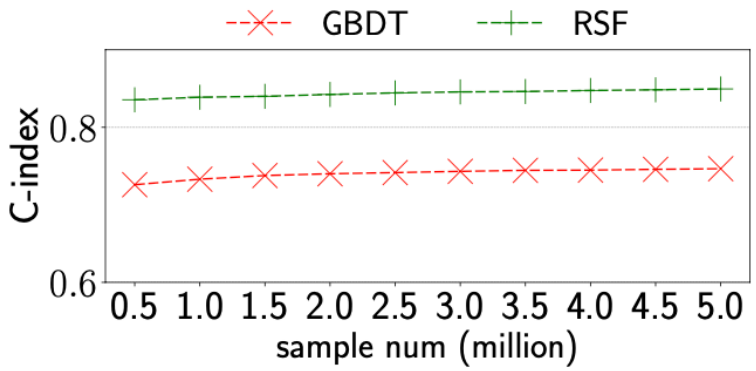
(a) *Logistics* workload



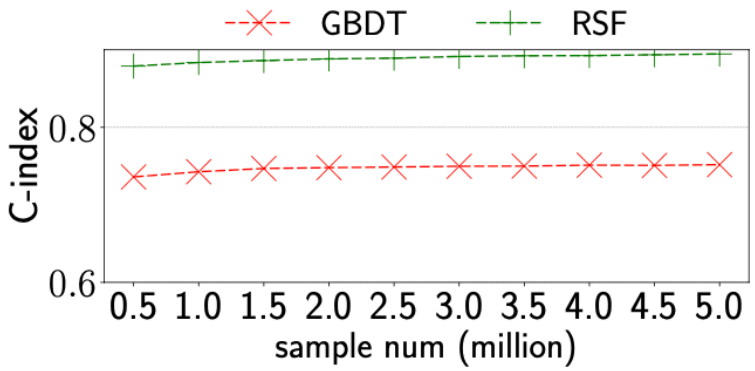
(b) *Order* workload

Experiments

- Impacts of the censored data
  - RSF can outperform GBDT even when only utilizing 10% of the latter’s training samples through properly handling the censored data



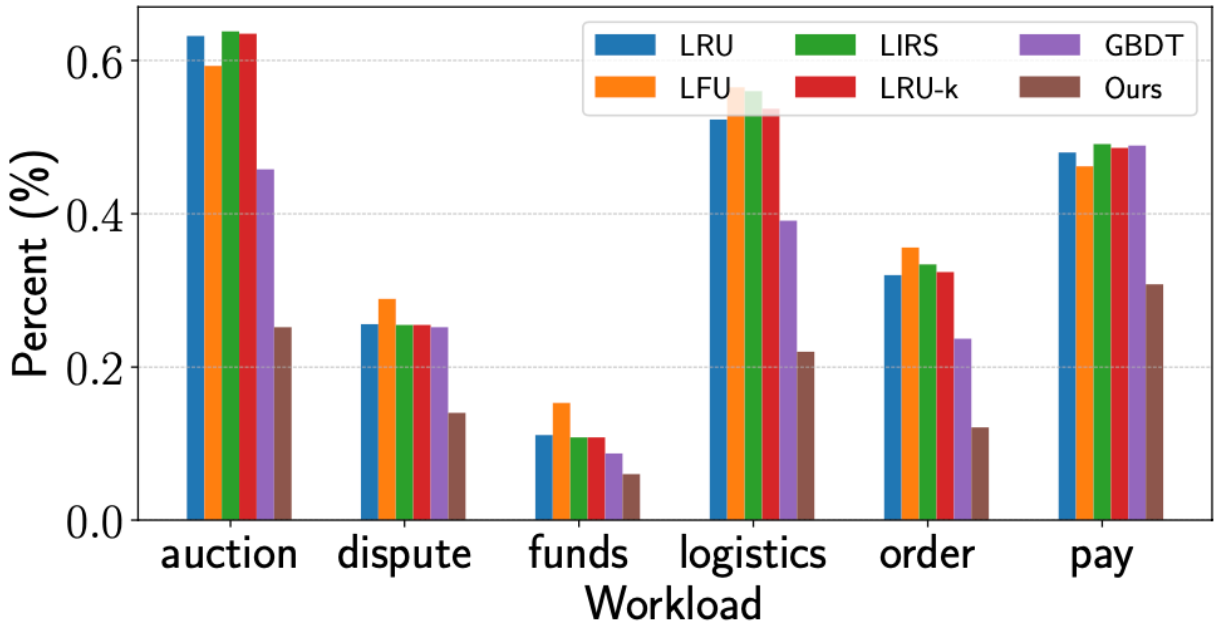
(a) *Logistics* workload



(b) *Order* workload

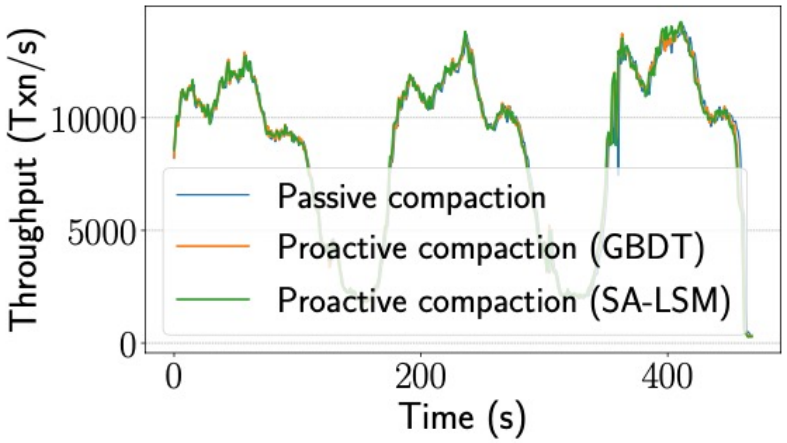
Experiments

- Algorithms performance
  - Predict the τ-event within the next coming 30 days.
  - SA-LSM can effectively decrease the $cold\_fir$ metric by ranging from **31.0%** to **48.9%** compared with the best baseline.

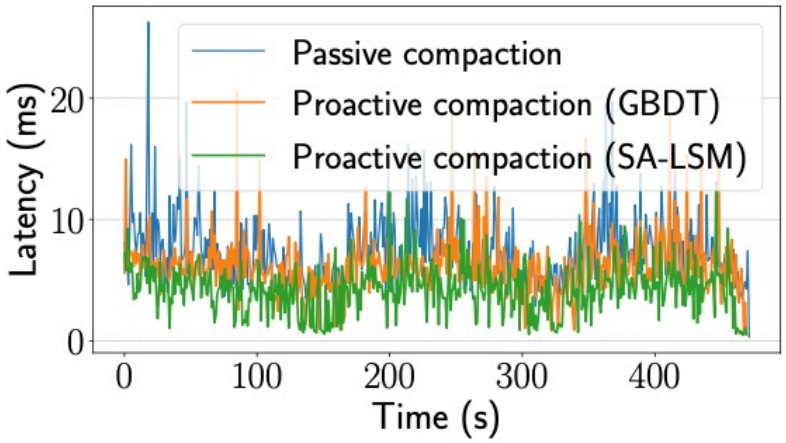


Experiments

- System performance
  - SA-LSM reduces the tail latencies by **31.5%** for *logistics* workload



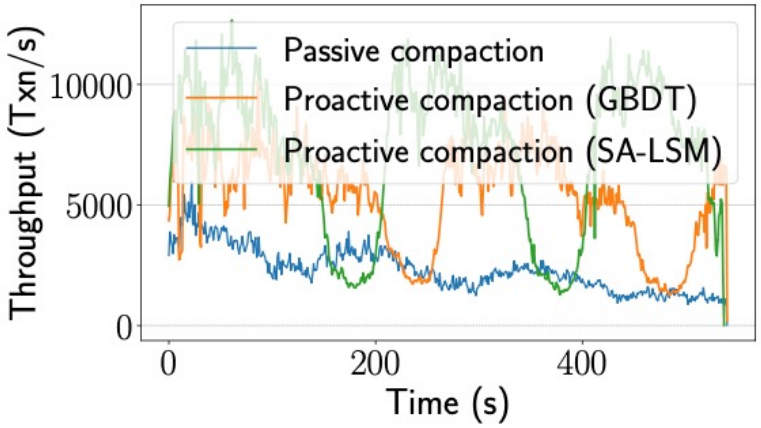
(a) Throughput



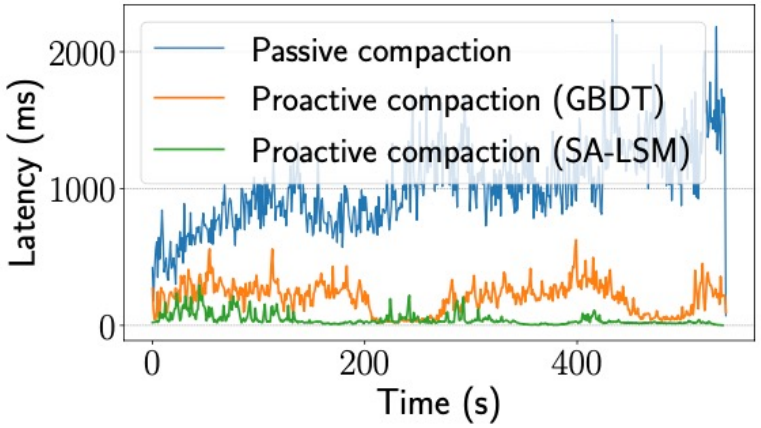
(b) Tail latency

Experiments

- System performance
  - SA-LSM reduces the tail latencies by **78.9%** for *order* workload



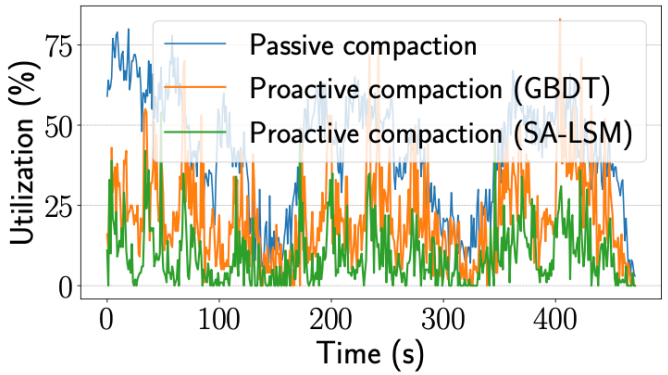
(a) Throughput



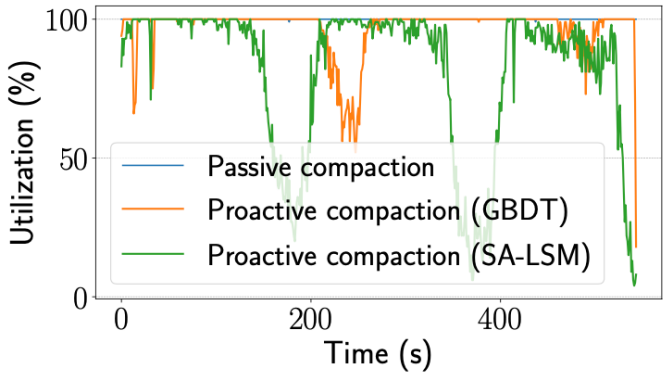
(b) Tail latency

Experiments

- System performance
  - The performance improvement is more substantial for order workload
    - The fractions of the cold data migrated (**51.7%** v.s. **28.1%**) , the I/O utilization of *order* workload reaches the bottleneck of HDD
    - The *cold_fir* metrics improvement (**48.9%** v.s. **43.7%**)



(a) *Logistics* workload



(b) *Order* workload

Conclusion

- We propose to utilize survival analysis, a statistical learning algorithm commonly used in biostatistics, to effectively compact cold data for LSM-tree based KV stores

- We design an external service in conjunction with a lightweight protocol to offload the heavy training and inference operations from the database kernel.

- The tail latency of the system is decreased by ranging from **31.5%** to **78.9%** on typical real-world workloads