ACER: Accelerating Complex Event Recognition via Two-Phase Filtering under Range Bitmap-Based Indexes

Shizhe Liu[1], **Haipeng Dai**[1], **Shaoxu Song**[2], **Meng Li**[1], **Jingsong Dai**[1], **Rong Gu**[1], **Guihai Chen**[1]

[1]State Key Laboratory for Novel Software Technology, Nanjing University; [2]BNRist, Tsinghua University

Motivation

Complex event recognition (CER) refers to identifying specific patterns composed of several primitive events in event stores. Since full-scanning event stores to identify primitive events holding query constraint conditions will incur costly I/O overhead, a mainstream and practical approach is using index techniques to obtain these events. However, prior index-based approaches [1] suffer from *significant I/O and sorting overhead* when dealing with high predicate selectivity or long query window (common in real-world applications), which leads to high query latency.

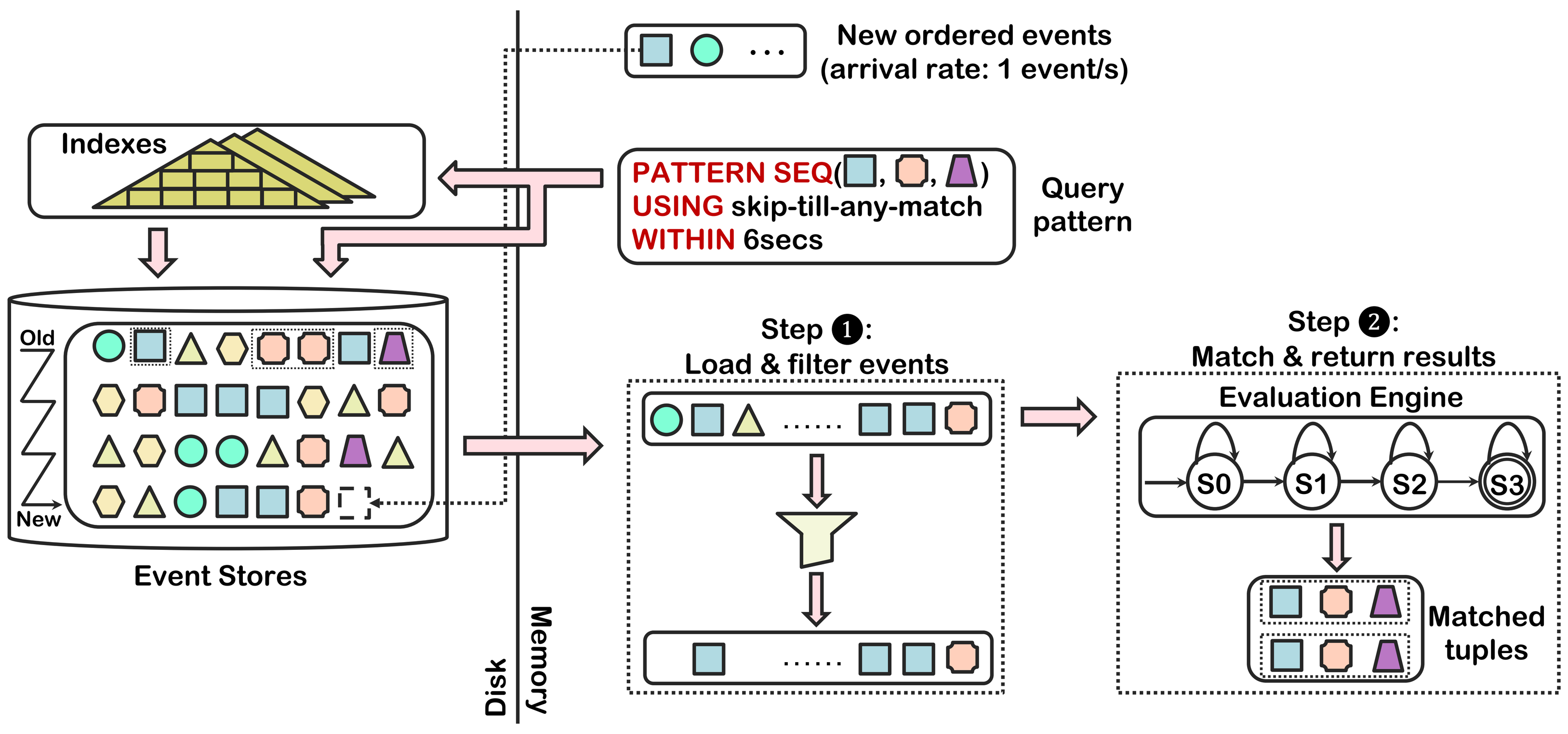

Figure 1: Process steps of complex event recognition.

Challenges

When designing index structures to accelerate CER in event stores, we face three primary challenges:

1) *alleviating the I/O overhead of reading indexes;*
2) *ensuring chronologically ordered results without sorting;*
3) *avoiding unnecessary disk access in indexes and events.*

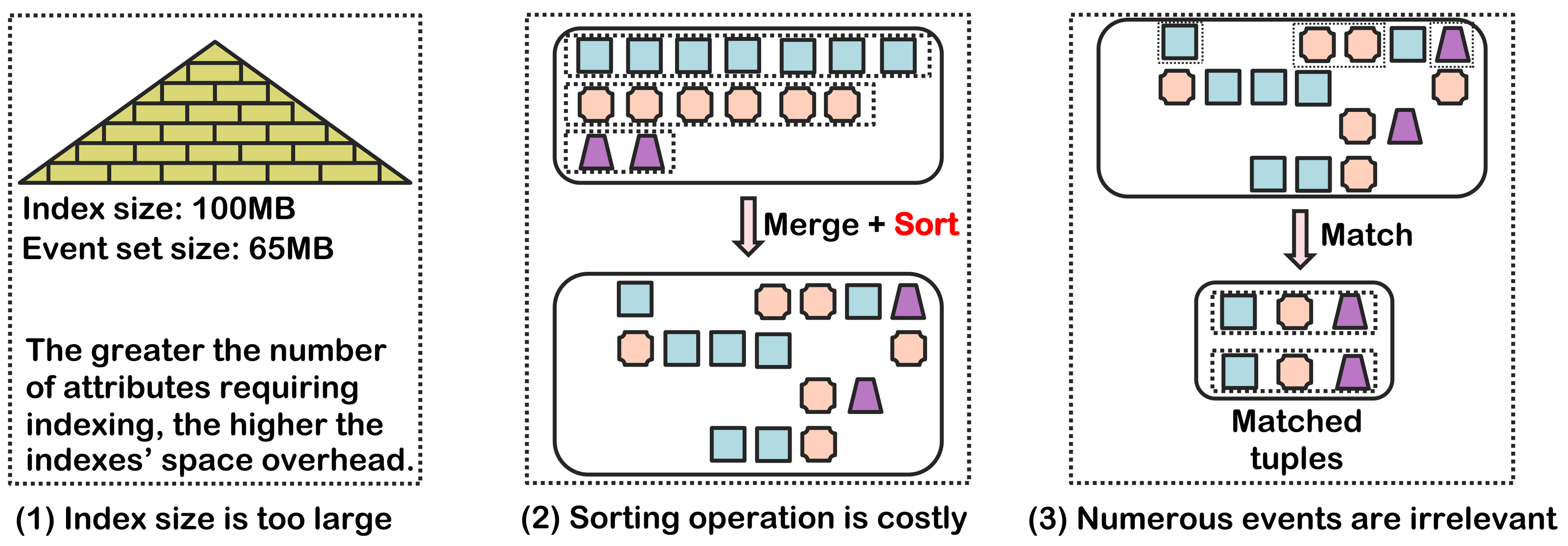

Figure 2: Three key challenges in designing index structures.

Method

We propose ACER, a Range Bitmap [2] based index (*note that bitmap-based indexes can ensure sequential disk access*), to accelerate CER. Its structure is illustrated in Figure 3. It mainly contains four components: (1) Page, (2) Buffer Pool, (3) Index Block, and (4) Synopsis Table.

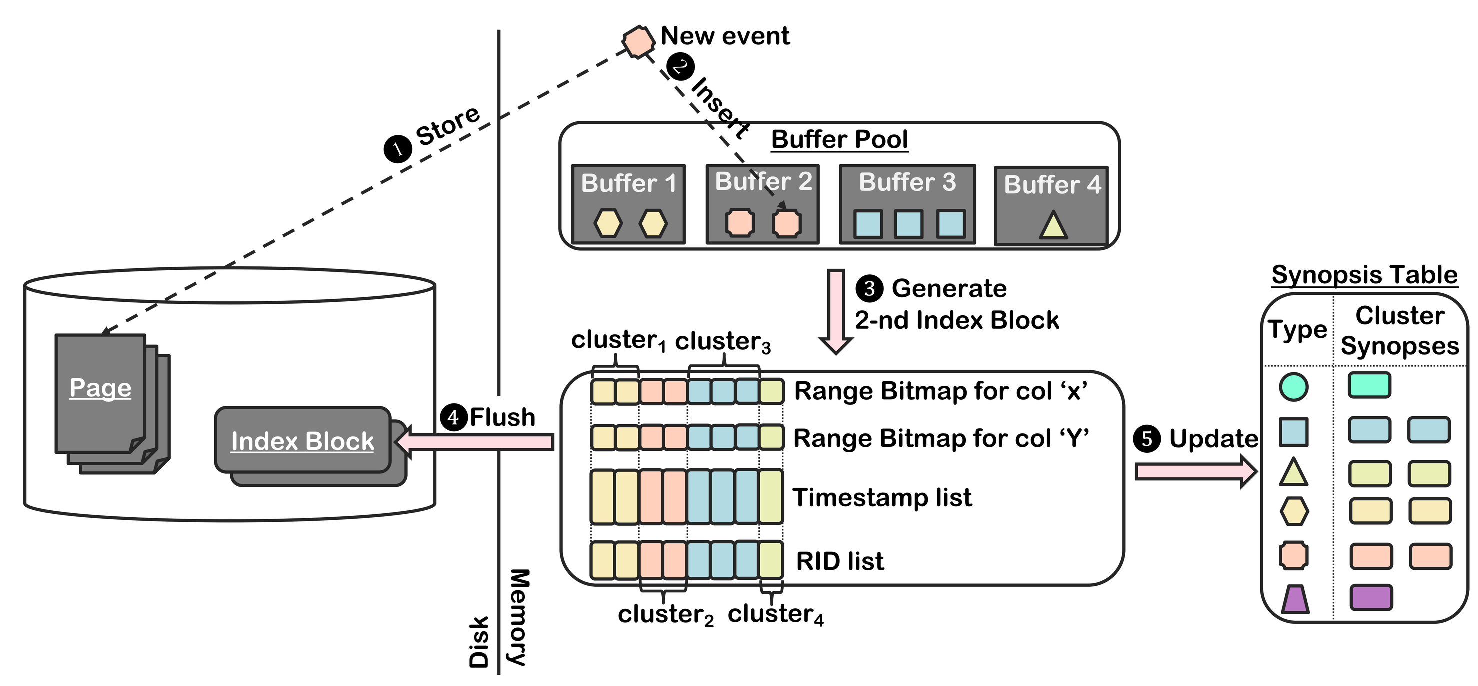

Figure 3: ACER structure overview.

WeChat QR Code

Repository QR Code

Query

To avoid unnecessary disk access in indexes and events, we propose a two-phase filtering algorithm. Its key intuition is that: (1) if an index block's time range cannot overlap with the time intervals that may contain matched results, it can be skipped accessing; (2) for the events whose timestamps are within a time interval, if none satisfy a variable's type condition and all predicate conditions, then these events cannot be combined into matched tuples, thus we can skip accessing them.

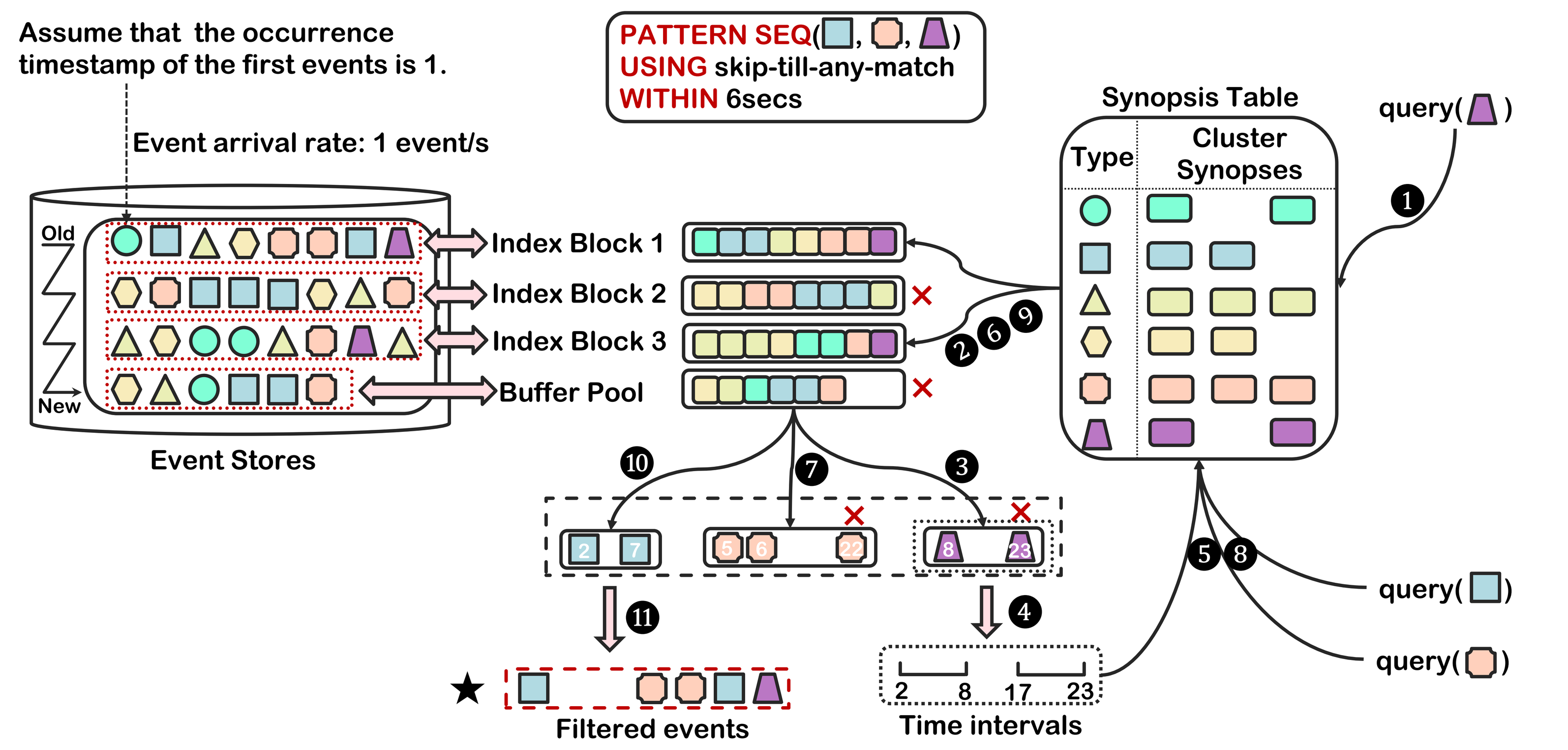

Figure 4: Two-phase filtering algorithm.

Evaluation

★ **ACER consistently outperforms FullScan across the three real-world datasets, with median speedup ranging from 6.5× to 70×.**

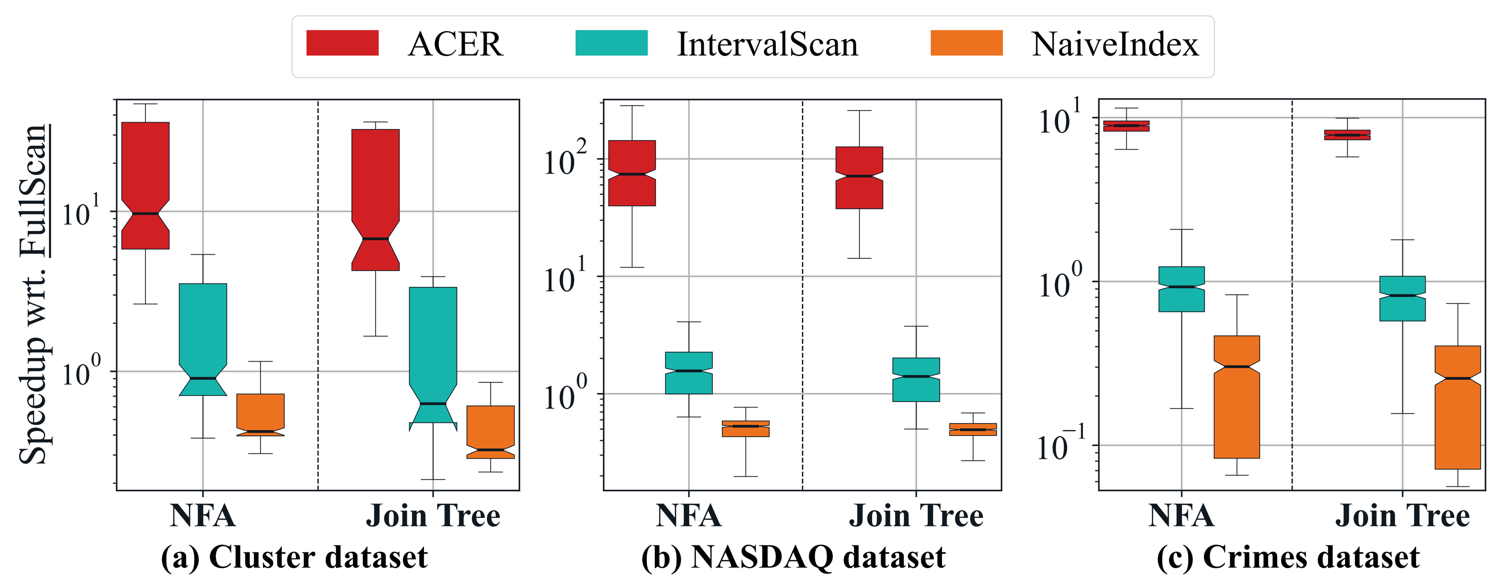

Figure 5: Speedup of index-based methods under two evaluation engines, *i.e.*, Non-deterministic Finite Automata (NFA) and Join Tree.

★ **ACER reduces insertion latency by around 10× and lowers storage overhead by 5× compared with SOTA index-based methods.**

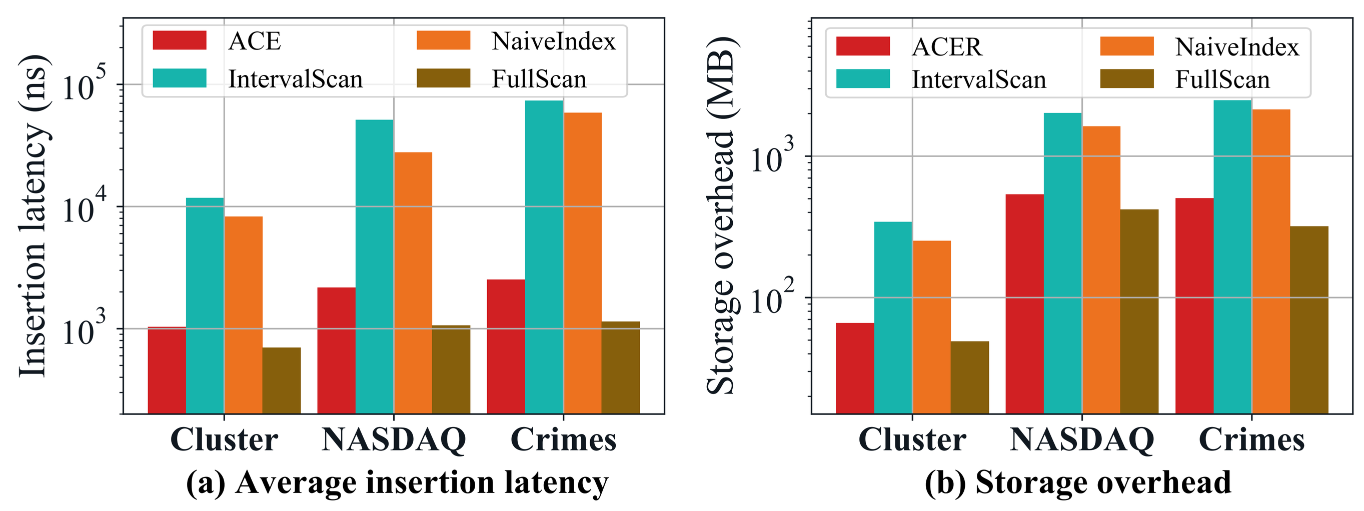

Figure 6: Comparison of insertion latency and storage overhead.

Conclusion

The core ideas of ACER are: (1) *reducing the size of index* to alleviate the I/O access overhead in indexes; (2) *constructing Range Bitmaps* for query attributes to produce ordered results without sorting; and (3) utilizing *the two-phase filtering algorithm* to avoid unnecessary disk access.

References

[1] Michael Körber, Nikolaus Glombiewski, Bernhard Seeger. Index-Accelerated Pattern Matching in Event Stores. In SIGMOD. 2021, pp. 1023-1036.

[2] Chee Yong Chan and Yannis E. Ioannidis. Bitmap Index Design and Evaluation. In SIGMOD. 1998, pp 355–366.