Technical Perspective: Diva: Dynamic Range Filter for Var-Length Keys and Queries

Viktor Leis
Technical University of Munich
leis@in.tum.de

To accelerate data access, database systems have long relied on probabilistic filters, most notably Bloom filters, to avoid unnecessary I/O. These compact structures are highly effective for point queries: they answer "is key x present?" with one-sided error, i.e., with false positives but not false negatives.

This success does not carry over naturally to range predicates. Hashing destroys order, and once the workload shifts from membership to range emptiness – "is there any key in $[l,r]$?" – traditional hash-based filters cease to be a satisfactory solution. Range filters fill this gap by allowing systems to rule out empty intervals and skip disk pages, SSTables, or cloud objects. This is particularly valuable in LSM-based engines, where pruning across many runs or levels is essential to performance.

What makes this problem especially difficult is that the properties required in practice pull in different directions. A useful range filter should operate under a tight DRAM budget, support arbitrary query lengths, handle variable-length keys, admit updates and growth, and still remain fast both to construct and to query. The paper's first contribution is a particularly clear articulation of this design space. It explains why all desirable guarantees cannot simultaneously hold in their strongest form for worst-case datasets, and uses this impossibility result not as a limitation but as guidance. Diva is explicitly designed for the common case that matters in real systems: realistic key distributions, variable-length keys and queries, and dynamic workloads. Thus, the work aims for completeness, but it does so with a careful understanding of what theory permits and what systems need.

The central idea behind Diva is elegant. It picks roughly one in a thousand keys from the sorted set and builds a cache-efficient trie from them that captures the coarse structure of the key space. Diva then compresses the keys between adjacent samples in a way that preserves order information without resorting to hashing. For each such group, Diva removes the longest common prefix that the trie already explains, and then truncates suffix bits while keeping a fixed-size middle fragment, or infix, that is still sufficient to distinguish keys in sorted order with low false-positive probability. This turns the sampled trie into a lightweight distribution model and the infixes into a compact, order-aware representation of local detail. The effect is exactly what one would want: dense regions are represented at finer effective granularity, while sparse regions are represented more coarsely, allowing the filter to adapt naturally to both short and long range queries.

The engineering of the Infix Stores, which make up most of the overall space, is impressive. Diva stores these infixes using quotienting, compact bitmaps, and rank/select-based navigation, so that queries examine only a small number of slots. Diva supports insertions, deletions, and growth by overprovisioning Infix Stores, spacing runs evenly, shifting locally when needed, and splitting or merging stores as new keys are added to or removed from the trie. This combination of conceptual simplicity and systems-minded detail is rare.

The empirical results reinforce the paper's claims. Across the evaluated datasets and workloads, Diva delivers a strong balance between false-positive rate, latency, construction cost, and update overhead. In end-to-end experiments inside WiredTiger, Diva speeds up empty range-query processing by as much as three orders of magnitude.

Overall, Diva is an excellent example of database systems research at its best. It starts from an important practical problem, identifies precisely why existing approaches fall short, and then introduces a design whose power comes not from unnecessary complexity, but from the right idea. The paper combines theory, algorithmic insight, and careful implementation to produce a range filter with strong real-world tradeoffs.