1. Name(s) of the author(s): **Ajit Mathew, Changwoo Min**
2. Title of article: “**HydraList:** **A Scalable In-Memory Index Using Asynchronous Updates and Partial Replication.”**
3. Title of journal, volume number, date, month, and page numbers: “**Proceedings of the VLDB Endowment. Vol. 13, No. 9”**
4. Statement of the problem or issue discussed: **An index should ideally have time complexity that is independent of the size of the key set-in order to be performant with large data sets. Critical sections should be reduced, and synchronization mechanisms should be carefully designed to decrease cache coherence traffic to ensure scalability. Furthermore, because servers have a complex memory hierarchy, data placement and memory access patterns are critical for high performance across all workload types.**
5. The author’s purpose, approach, or method: **HydraList is a new concurrent, scalable, and high-performance in-memory index structure for massive multi-core machines, which they present in this paper. The fundamental insight behind HydraList's design is that an index structure can be split into two components (search and data layers), each of which can be modified separately, resulting in lower synchronization overhead.**
6. Primary (evaluation) result: **In comparison to Wormhole, Wormhole and HydraList take different approaches to design. Wormhole tries to boost performance by employing a highly effective search layer. However, other factors play a role in performance in massive multicores, as shown in their analysis. Wormhole's and HydraList's designs vary in three keyways: 1) Search layer changes, 2) data layout updates, and 3) concurrency control updates HydraList allows search layers to be replicated per-NUMA. As a result, HydraList's remote cache line accesses are greatly reduced**.