

## ABSTRACT

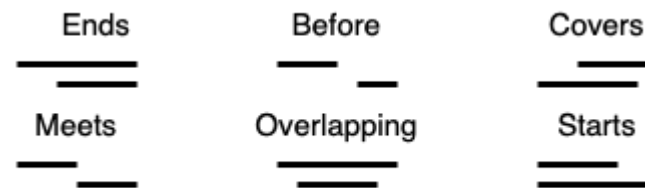
The implementation of *interval time sweep join* aims to replace *interval sort merge join* in AsterixDB, which should increase the speed of interval joins in the database. These joins operate on temporal data (intervals), which represent a span with a starting and ending point. Temporal data and interval joins have a range of applications from the medicine to business. This project also completes the set of Allen's relations in AsterixDB by adding six new relations, and it also adds an active sweep manager unit test to ensures memory is handled correctly in time sweep.

## ABOUT ASTERIX DATABASE

AsterixDB is a highly scalable NoSQL database co-developed by the University of California Irvine and the University of California Riverside. It is now managed by the Apache Software Foundation, and used by several software companies including Couchbase and Yardi Systems as part of their services. AsterixDB continues to be primarily developed and supported by those companies, and by U.C. Irvine and U.C. Riverside who, together, have received over two million dollars in grant money from the National Science Foundation.

## ALLEN'S RELATIONS

Allen's relations describe thirteen different ways to process intervals. For example, assume there is a database that stores the first and last day of every student and staff member that has attended or worked at your university. If you wanted to know who started working at the same time you started college, you might choose the relation *starts*; the figure below depicts *starts* and five other Allen's relations. Below the figure is a sample query for this join.



```
SELECT * FROM Staff AS f, Students AS d
WHERE interval_starts(f.employment, d.attendance);
```

## SETTING UP TIME SWEEP

The implementation of interval time sweep join includes three major pieces: the physical operator, the operator descriptor, and the join algorithm. When an interval join is queried, there are a set of conditions that must be met to call time sweep. If the conditions are correct, there are specific properties including the type of Allen's relation that are set up and passed to the physical operator. Based on this information, the physical operator sets up a partitioning scheme to describe the data. Then, the operator descriptor is called to set up and run *time sweep's* activities.

## THE TIME SWEEP ALGORITHM

The *interval time sweep join* algorithm works by switching between input streams. The algorithm always chooses the stream with the interval that starts earliest in time. Once the stream that the join is processing is no longer first, the algorithm switches to the other stream and processing continues on that side. *Time sweep* continues working this way until all necessary tuples have been processed.

## CONCLUSION

*Time sweep* is now fully implemented and passing its runtime tests. Currently, it reads from disk on both input streams, but eventually, one of the streams should be managed in memory. In the future, Dr. Carman, a professor at Walla Walla University, will perform tests against *interval time sweep join* and *interval sort merge join* to verify that *time sweep* is faster. Based on these results, we will be able to merge *time sweep* into the active Asterix repository. I plan to continue working on Asterix in the future for Google Summer of Code, and I value all of the time I've spent learning about interval joins throughout this year.

## REFERENCE

- <https://doi.org/10.1007/s00778-020-00650-5>
- [https://github.com/prestoncarman/asterixdb/tree/ecarm002/interval\\_io\\_in\\_symmetric](https://github.com/prestoncarman/asterixdb/tree/ecarm002/interval_io_in_symmetric)