

Milestone 3
CS165, Fall 2016
Kevin Zhang

1 Goals

The goal of this milestone is to introduce query parallelization by allowing select queries to share the same scanned data and return multiple sets of results using only one select scan. In addition, we add scan parallelization by making use of multiple cores, vectorizing the data and scanning chunks of each column at a time, on each core.

2 Technical Description

To split queries up into batches, we need to separate the scan and qualifying check components of a select query. To do this, we divide each select query into the column which it scans, along with the logic that selects qualifying values. We create a new struct, `QuerySet`, that contains a pointer to a `Column` object, along with a `Comparator*` array. Batches then result in a `QuerySet*` array of shared scans that we need to perform; each shared scan will be performed while executing each `Comparator`. Each `QuerySet` gives a `Result*` array, which we then store in the corresponding variable in our pool.

To split scans across cores, we vectorize access to our `Column` objects. Upon starting the server, we perform system calls to determine the number of cores available, then fork that many processes and divide our `Column` objects into as many components (broken into multiples of a page size). One core will be a coordinator; it creates a `Result*` array of size N , where N is the number of cores usable for scanning. Each of these N cores then scans their corresponding chunk of data, storing the results in the corresponding `Result` object. This assumes that each core has its own L_1 cache, in order to ensure that pages are not kicked out by other cores as the scan progresses. This is a reasonable assumption on most modern machines. If this is not true, we have to worry about pages of data being kicked out by separate cores; in this case, we should assign a `QuerySet` to each core instead, and have all cores process one chunk of data at a time.

3 Evaluation

Not implemented yet!

4 Challenges & Open Items

Haven't begun implementation yet.