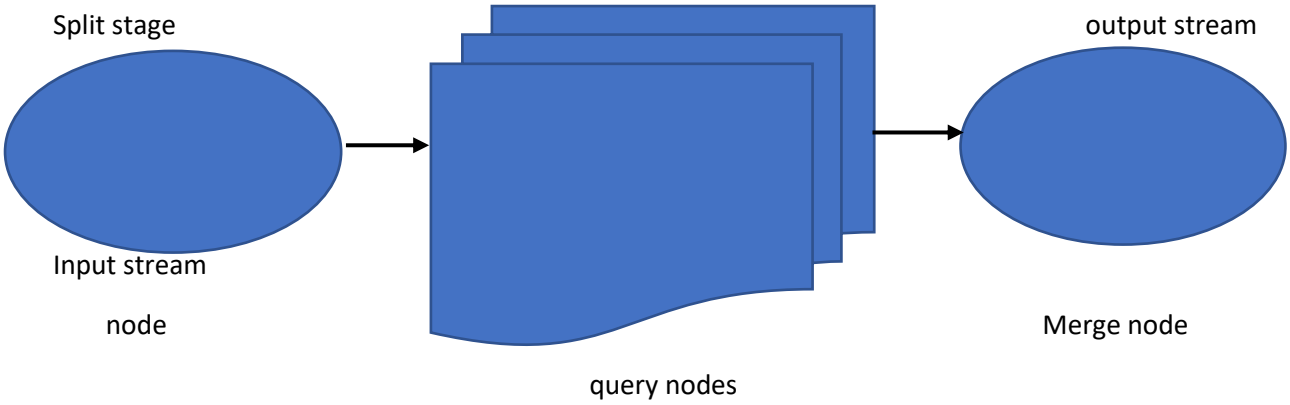
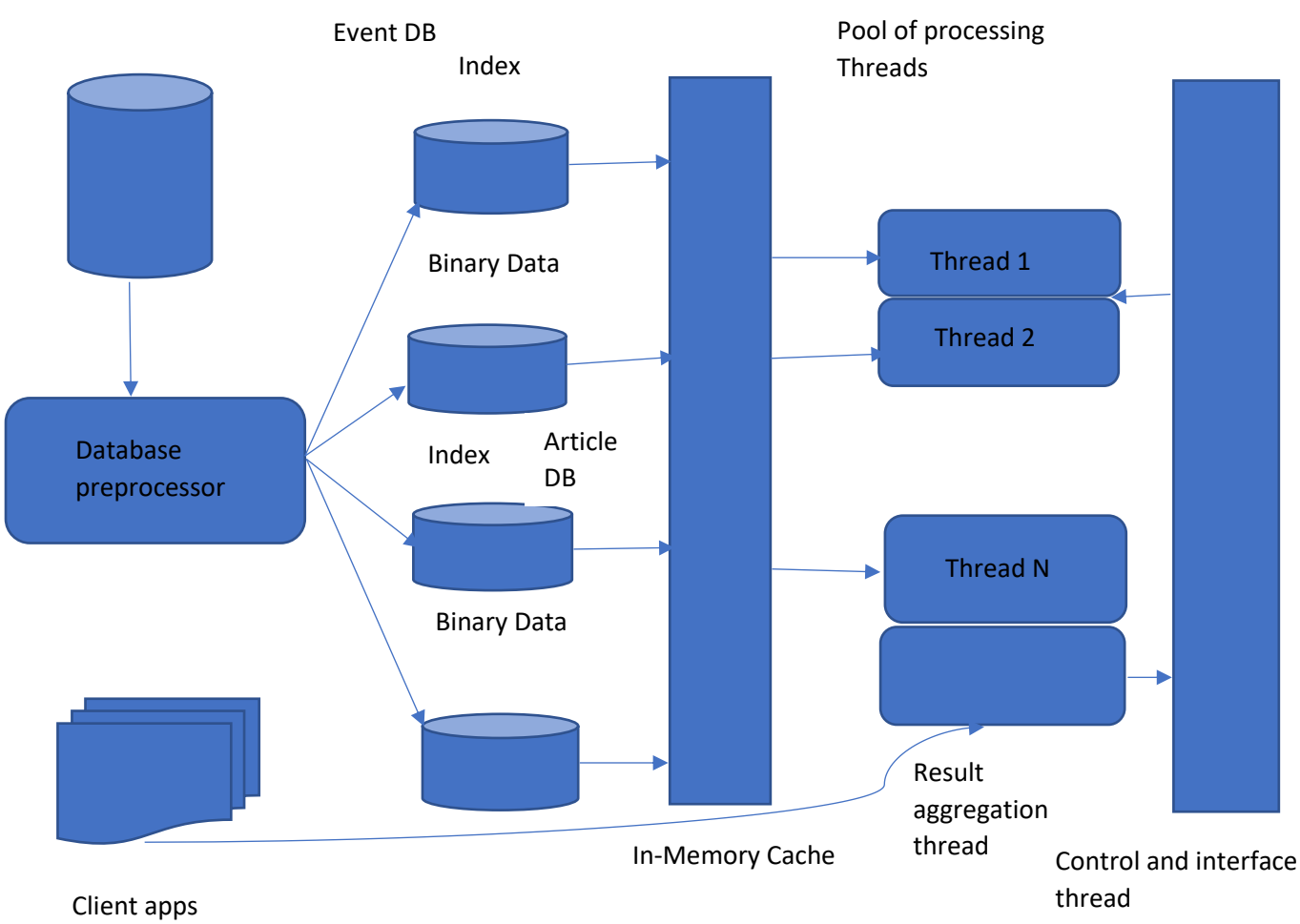


Parallel stream processing framework overview



GDELT ANALYSIS SYSTEM

The low number of active sites shows that many of the sources tracked by GDELT are periodical publications rather than daily newspapers. our data shows a slight but noticeable deviation from the power law around the centre of the graph. Note that unlike Lu et al., we take all sources and all articles into account. A crucial component of our efficient handling of GDELT data is its conversion to a binary format. While doing so is straightforward, it requires cleaning and checking the data. Doing so, we found a small number of problems with the GDELT source data which are listed in Table II. We present the development of key statistics over time. The number of sources is shown in Figure 3, events in Figure 4, and articles in Figure 5. For readability reasons we aggregated time into quarters. Note that the first entry begins on the 18th of February 2015, and thus does not represent a full quarter. The numbers are relatively stable over time, with a slight decrease in the years 2018 and 2019. Interestingly, while the number of sources is relatively stable, only about one third of the sources are active in any given quarter. 0 per page at the time they register.

The current version 2.0 of GDELT monitors both English and non-English news sources, with archives going back to 2015. Non-English articles in 65 languages are translated to English for further analysis using what is believed to be the largest real-time streaming news machine translation deployment in the world. It has the capacity to monitor news of the entire world. 98.4% of the monitored content is translated in real time. Thus, it is most likely the system with the widest reach w.r.t. media in the non-western world, although its reach in these areas is limited compared to the western world.

As a result, we can query large amounts of data much faster. The system is written in C++ using OpenMP for parallelization. It is designed to run on large memory nodes. The overall system structure is depicted in Figure 1. Before working with the data, we once convert GDELT database files with our pre-processing tool in order to build indexed version of the database which contains data fields in machine-readable binary format. User-defined queries to the database are processed via a query execution engine optimized for in-memory handling of previously converted GDELT data. We implemented parallel version of the most intensive aggregated queries