The LIEBRE Stream Processing Engine

Since I started by Ph.D. back in 2008, I spent a considerable amount of time digging into the internals of stream processing engines. Pioneer engines like Aurora and Borealis were not really easy to work with. During the Ph.D., I had to “fight” a lot with Borealis in order to design and develop StreamCloud, one of the first parallel and distributed stream processing engines.

Once the Ph.D. was over, I continued my research in data streaming and learned how to use new systems, like Storm, Flink or Spark. Things got way easier than they were before, and a simple “hello world” example is actually a matter of minutes. Still, I always had (and still have) mixed feelings with these systems. On the positive side, they are very powerful and supported by a large community. On the less positive side, they force you to design and implementation decisions that you might find sub-optimal (remember my perspective is the one of a researcher in data streaming).

Of course, one could always make changes to the internal architecture of such systems, but the engineering effort is considerable and might be needed each time a new streaming platform becomes popular and every time existing ones get updated.

Because of this, I decided to put together a minimal stream processing engine (Liebre) and make it available in GitHub. You can find the documentation here and the actual code here.

The idea is to keep adding to it operators I develop for research (you can already find standard operators in it) and to use it for research in data streaming, especially when using IoT devices (the footprint of streaming engines like Storm and Flink is considerable, since they are designed for large data centers, rather than cheap single-board devices). I decided to make the code available and put together some documentation in case someone is interested in trying it or wishes to provide feedback.