

Industry:

Products and Services:

Challenge:

To learn what Complex Event Processing is - read our article titled What is Complex Event Processing?

Traditional RDBMAS cannot handle large quantities of data as efficiently as CEP.

There are several factors which make CEP a challenge:

- The evaluation of different kinds of data
- The quantity of the data needing to be processed
- The need to process this data in real-time / near real-time (expected response times range from a few milliseconds to 1-2 seconds)
- Consideration of historical data
- Consideration of current device state, which can differ from the last event received from a device / source
- Scalability
- Redundancy

Traditional systems don't cut it

Performance

For many years, traditional RDBMS were the main stay of enterprise systems, and by and large still are the mainstay in many applications. These systems cannot handle the demands of problems which involve writing large quantities of data from many different connections, many searches through large quantities of data, all concurrently, all done in less than a second for each operation. Getting into benchmarks can be a whole post in its self, and it is important to frame the circumstances correctly for fair comparison. We plan on revisiting this topic at another time.

Cogswell.io is Aviata's Complex Event Processing solution.

Architecting and building a CEP solution is no easy matter and can often take a team of developers several years.

Sharding

Sharding is the process of splitting your data up amongst several different servers. There are several reasons that this is done, some of which are:

- Too large of a data set for a single database server
- Performance reasons - your single database server is being overwhelmed, if you shard your data, then you can reduce the overall load on each database server
- Redundancy reasons

For more traditional databases, sharding is often a real pain point. They were not built from the ground up with it in mind. While it can be done, it is a very tedious process and there is a lot that can go wrong. Many newer so called NOSQL databases were built with sharding in mind from the ground up. Some make sharding incredibly simple, and there are far fewer gotchas.

Redundancy

Again, traditional systems often struggle here. They were not built with redundancy concepts in mind and it was often an afterthought. Many RDBMS are focused on the idea of ACID (Atomicity, Consistency, Isolation, Durability) and the fact of the matter is that this does not lend itself well to scaling, shading, redundancy, and frankly overall performance. Some solutions involve setting up a master / slave configuration where all of the operations are performed on the master to ensure consistency, and they are then duplicated to the slaves. But again, if you have 1 master, 3 slaves but all operations are occurring on the master - you have 3 servers which could be handling load / queries but are not used in that manner in an effort to preserve data consistency. You may have some redundancy covered, but this is a sub-optimal solution and is really not a good one for performance.

Summary

In short, traditional database systems are not well equipped to handle the kinds of loads and problems that CEP systems deal with. You may be able to cobble together a solution, but it is very time consuming and I have serious doubts that it could stand up to newer, NOSQL solutions built with these concepts in mind.

However, it's not a free lunch. Newer, non-traditional systems often focus on doing one thing very, very well at the expense of everything else. For instance, they may be very quick at inserts, but have poor read performance. They may not allow joining of data. They often place restrictions on the kinds of data and the structure of the data in order for them to excel in one area. They often give up the consistency of data, or atomicity in order to gain in performance in a distributed clustered environment. The key is picking the right backend database solution (or many solutions) which has the features and qualities you need.

These newer technologies require a fair amount of expertise to use and architect properly. Many development companies do not have the expertise with these systems or the knowledge to properly design and architect an infrastructure to properly handle and solve CEP problems. Even for those companies who have some experience with some of these solutions - architecting and building a CEP solution is not easy and can often take a team of developers several years.