Major work of Complex Event Processing

> Cugola et.al "Processing flow of information: From data stream to comple event processing", ACM computing surveys 2011

Stream databases

1. *"Monitoring Streams: A new class of data management applications"*, VLDB'02
2. *"TelegraphCQ:continuous dataflow processing"*,SIGMOD'03
3. *"Query processing, resource management, and approximation in a data stream management system"*,CIDR'03
4. *"Sequence query processing"*,SIGMOD RECORD'94

Those systems still used a storage centred approach with expressive query language, yet were not optimized for processing sequential patterns.

> e.g. TelegraphCQ was built by modifying the existing architecture of open source PostgreSQL database to support continuous queries.

Publish-subscribe system

1. *"Matching events in a content-based subscription system"*, PODC'99
2. *"Filtering algorithms and implementation for very fast publish/subscribe systems"*, SIGMOD RECORD'01

System like those are charcterized by query language having very limited expression power and only allowing simple selection predicates applicable to individual events in a data stream.

> System like these trade expressiveness for performance. Data flow through them (often as an acyclic graph) where data stream are filtered, combined, or transformed on the way. Those system are often distributed, highly parallel, and support sharding. Other representative in this category: **Aurora, PIPES, STREAM, Borealis, S4, Storm**.
>
> those system cannot be treated as complex event processing engine as it does not even have the basic temporal event processing capabilities over time or length windows, and it cannot handle complex events.

Full complex Event processing system

1. ODE:*"Composite event specification in active database"*, VLDB'92

   > Stopped.

2. SASE(UMASS):*"[High-Performance Complex Event Processing over Streams]*, sigmod'06

   > still active.
   >
   > Their motivated application domain is RFID.

3. Cayuga(cornell):*"Cayuga:A General Purpose Event Monitoring System"*, CIDR'07

   > Their original system is a publish-subscribe system, which Cayuga is implementing based on.
   >
   > Their motivated application domain is online tradding.
   >
   > Their reseach on this topic had stopped, because of the graduation of core phd student.

4. Esper(commercial, open-source)

   > The above CEP use some variants of NFA architecture. And interestingly, all of those system rely on sequential event processing mechanism, except Cayuga uses *priority queues* and it also tries to perform multi query optimization techniques to merge manifestly equivalent states events having the same time stamps to be processed together, but the core is still single threaded.

5. Siddhi:*""Siddhi:A Second Look at Complex Event Processing Architectures"* - GCE'11

   The author try to improve the architecture of CEP system from four decision:

   ```
   - Multi-threading
   ```

```
- Queues and use of pipelining
- Nested queries and chaining streams
- Query optimization and common sub-query elimination
```

The architecture are still rely on NFA.

6. HP-Chaos (WPI, boss: Elke A. Rundensteiner): **I don't know which paper is their first publication of this series.. there're just too-many and their engine is close source**

   > still active. Close sourced, the detail of their architecture is not available.
   >
   > They are very active, almost two paper in top conference each year.

7. Zstream(MIT):*"zstream a cost-based query processor for adaptively detecting composite events"*, SIGMOD'09

   > Interestingly, their system is rely on **query-plan-tree** architecture.
   >
   > Stopped, maybe because of the graduation of core phd student.

8. Borealis: *"rip run-based intra-query parallelism for scalable complex event processing"*-DEBS'14.

   > Very recently, Borealis team also participate in the game of "Full Complex Event Processing System".