Elastic Complex Event Processing under Varying Query Load

Thomas Heinze, Yinying Pan, Yuanzhen Ji, Franz Josef Grüneberger, Zbigniew Jerzak, Christof Fetzer
Agenda

1) Elasticity for Complex Event Processing
2) Concept
3) Evaluation
4) Conclusion & Future Work
Elasticity

Elasticity is a capability to handle computing resources which can be rapidly: "[...] provisioned, in some cases automatically, to quickly scale out and rapidly released to quickly scale in“ [1]
Elasticity

query load / event rate

Load

Static Provisioning

Elastic Provisioning

Underprovisioning

Overprovisioning

time

8:00 18:00 1:00 8:00 18:00
Elasticity for Complex Event Processing

- System S elastically scale thread count per operator [2]
- Scale out solution with increasing event rate for StreamCloud [3]
- Recently research on state management for efficient scale out [4]

Contributions of this work: Elastic Scaling with varying Query Load
System Architecture

Queries

Operator Placement
Processing Coordination
FUGU

Distributed CEP Engine
Load Model

- Modelling CPU, memory and network consumption
- Measured constantly for individual hosts and operators
- Estimation for newly added operators based on Viglas [5] and work from Borealis [6]
Operator Placement using Bin Packing
Rebalancing Heuristics

- **observation:** fragmentation due to adding and removing operators without global reoptimization

- **idea:** detect situations where by simple rebalancing less hosts could be allocated new host or used hosts could be released

  - **Heuristic A:** Release less loaded hosts on query removal
  
  - **Heuristic B:** Avoid adding additional hosts on query addition
Rebalancing On Query Removal

- Check if minimal number of hosts is used
  \[
  host_{\text{min}} = \left\lfloor \frac{\sum_{V_{\text{op}}} \text{load}_{\text{CPU}}(op)}{\text{thres}} \right\rfloor
  \]

- Release all not needed hosts by reassigning their operators
Rebalancing On Query Addition

Host 1
- f4
- f2
- f1
- a1

Host 2
- a3
- f3
- a2

Host 3
- a5
- a4
Evaluation Setup

- implemented approach on top of state of art commercial CEP engine
- private cloud environment with up to 10 VM’s
- event input from Frankfurt Stock Exchange with fixed/variable data rate
- used non-incremental aggregation queries with public available query work load pattern
Elastic Scaling with Varying Query Load
Rebalancing Heuristics

![Graph showing the comparison of Avg. Utilization with different Utilization Thresholds. The graph compares the performance of 'Without rebalancing', 'Ideal', and 'With Rebalancing'.]
Latency for Rebalancing Heuristics

![Graph showing Latency Ratio vs Utilization Threshold]

- **Latency Ratio**
- **Utilization Threshold**

- **Without rebalancing**
- **Ideal**
- **With Rebalancing**
Variable Event Rate

- Fixed Data Rate 750 Event/s
- Var. Data Rate (Speedup 5)
- Var. Data Rate (Speedup 10)
Summary

- Elasticity allows handling varying load more cost efficient
- Designing an elastic CEP system always comes with trade off between utilization and end to end latency
- Heuristics can be used to increase utilization in case of varying query load
Future Work

- Extend our approach to allow handling varying data rate
- Improve trade-off between latency and reached utilization
- Support user in configuring the elastic CEP system by automatic parameter optimization


