Semantically Partitioned Complex Event Processing

Lasaris

Filip Nguyen

October 9, 2014

Agenda

1. Complex Event Processing
2. Semantically Partitioned CEP
   - Goals of Research
   - Model
   - Implementation
   - Benefits
3. Planned Experimental Datasets

Outline

Motivation and Usage

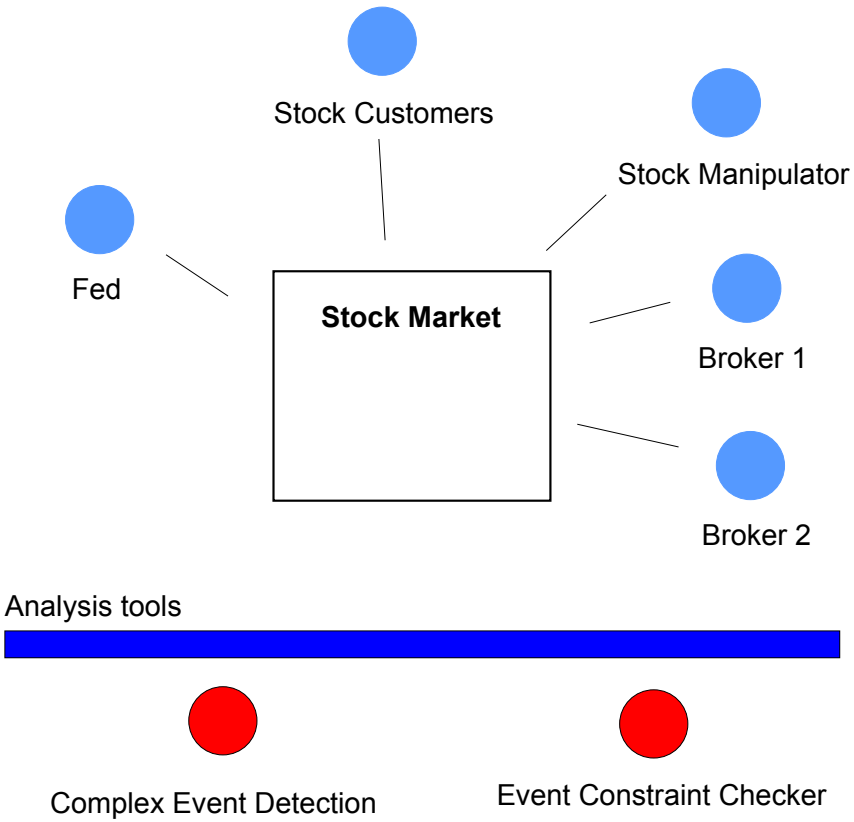


Figure 1: Illustrative Example of CEP Deployment

- Event: An object that is a record of an activity in a system.
- Figure shows a real-world concept Stock Market. From IT perspective it is a number of communicating systems
- CEP is a tool to detect so called Complex Events e.g. Economic crisis is approaching

Motivation and Usage

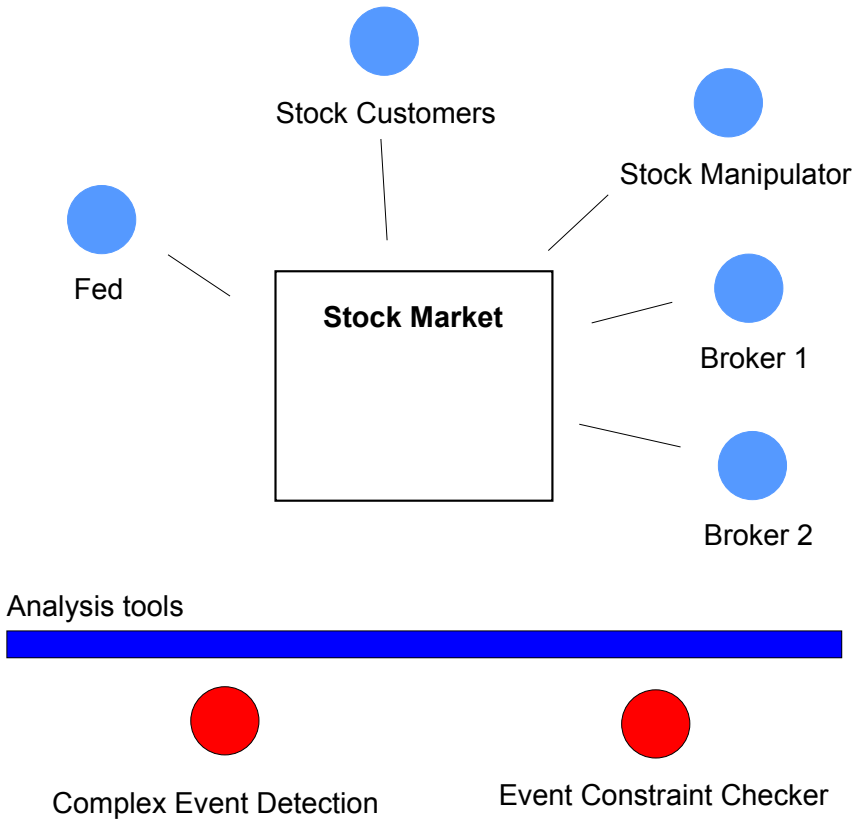


Figure 1: Illustrative Example of CEP Deployment

- Event: An object that is a record of an activity in a system.
- Figure shows a real-world concept Stock Market. From IT perspective it is a number of communicating systems
- CEP is a tool to detect so called Complex Events e.g. Economic crisis is approaching
- Using temporal operators
- Dealing with large amounts of events
- Processing of events on-line

Scaling and Performance

Why study CEP performance?

- In CEP goal of many research papers is to improve performance
- DEBS Grand Challenge. Other public CEP benchmarks such as Fincos

What is traditional approach to achieve performance?

- Carefully build Event Processing Network
- A EPN consists of EPA. The agent might filter events

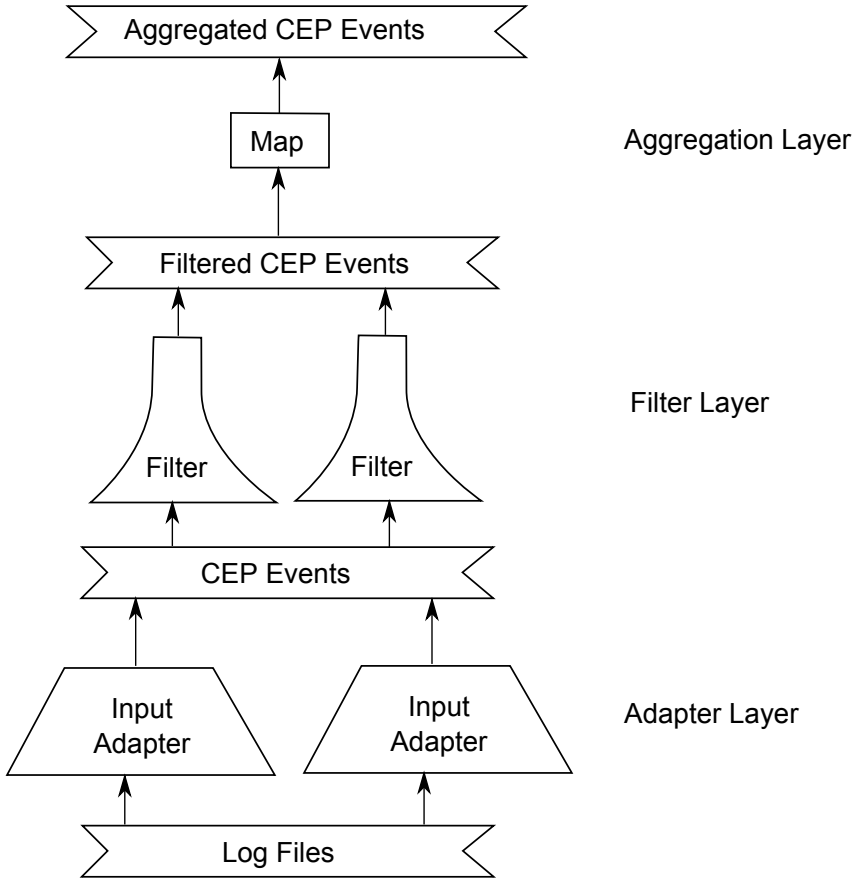


Figure 2: An EPN

Outline

Research Problems

How to horizontally scale CEP?

- Devise algorithms
- Develop a Java based ecosystem for distributed CEP

Research Problems

How to horizontally scale CEP?

- Devise algorithms
- Develop a Java based ecosystem for distributed CEP

How to capture this formally?

- Using a graph representation

Research Problems

How to horizontally scale CEP?

- Devise algorithms
- Develop a Java based ecosystem for distributed CEP

How to capture this formally?

- Using a graph representation

What if an engine wouldn't find every Complex Event? What if we prefer bigger performance to high *accuracy*?

Definition Engine Accuracy

A CEP engine A has accuracy $C_A \in (0, 1)$. C_A signifies percentage amount of Complex Events out of all Complex Events uncovered by ideal CEP engine.

Research Problems

How to horizontally scale CEP?

- Devise algorithms
- Develop a Java based ecosystem for distributed CEP

How to capture this formally?

- Using a graph representation

What if an engine wouldn't find every Complex Event? What if we prefer bigger performance to high *accuracy*?

Definition Engine Accuracy

A CEP engine A has accuracy $C_A \in (0, 1)$. C_A signifies percentage amount of Complex Events out of all Complex Events uncovered by ideal CEP engine.

How much is the accuracy/performance trade-off usable in today's EP problems?

Peer to Peer Horizontal Scaling

Trend in today's middleware is to leave centralized architecture of a solution and instead use decentralized peer-to-peer horizontal scaling.

- **Databases:** NoSQL Databases - e.g. Apache Cassandra
- **Web Servers:**Webserver session replication
- **Communication Middleware:**Messaging systems such as JBoss HornetQ
- **Caching Middleware:** in-memory caches like JBoss Data Grid

Theoretical Model

- I remove the concept of centralized CEP engine
- Graph node is so called Peer (producer, consumer and CEP engine) e.g. for a Stock Market server to become a peer it would need to have my Java daemon deployed

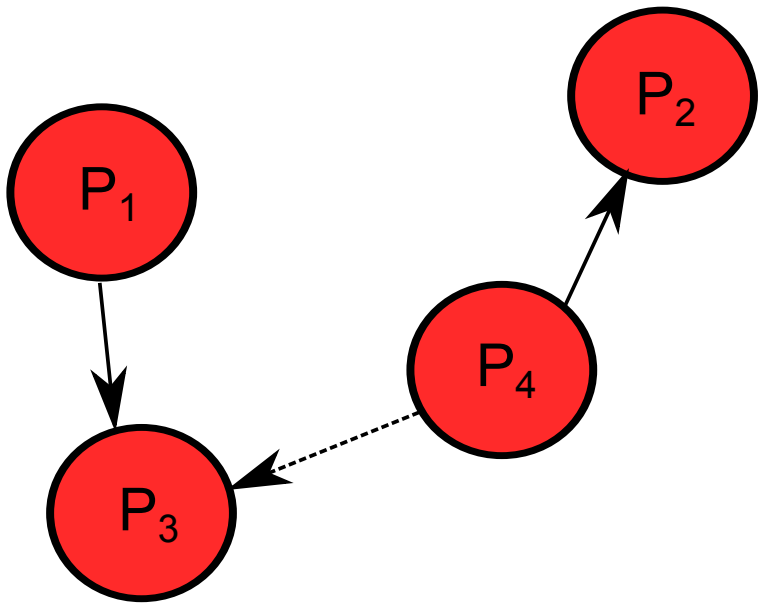


Figure 3: Peers Exchanging Events

Theoretical Model

- I remove the concept of centralized CEP engine
- Graph node is so called Peer (producer, consumer and CEP engine) e.g. for a Stock Market server to become a peer it would need to have my Java daemon deployed
- During processing of events, edges are dynamically added and removed

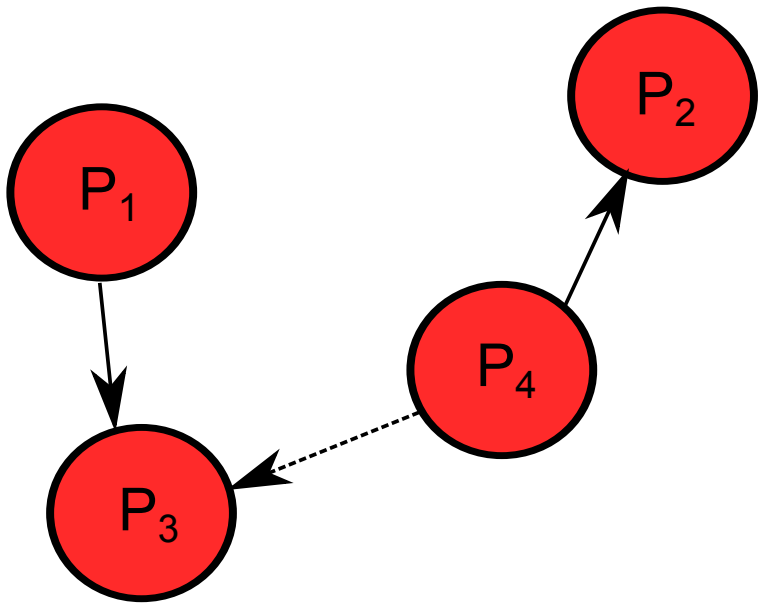


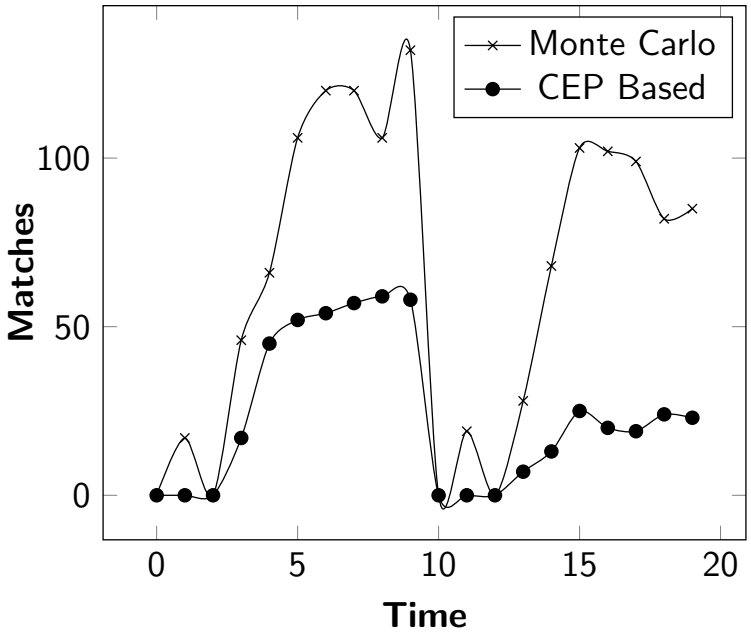
Figure 3: Peers Exchanging Events

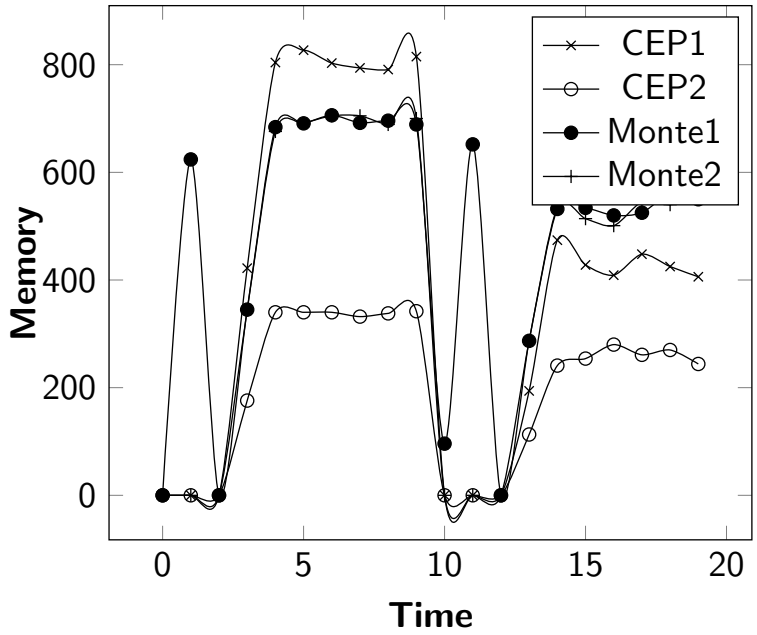
Semantic Partitioning

- The strategy that dynamically adds and removes oriented edges in the graph
- In effect it is a strategy to disseminate events between sets of peers
- Monte Carlo strategy, CEP Based strategy

Simulation Results

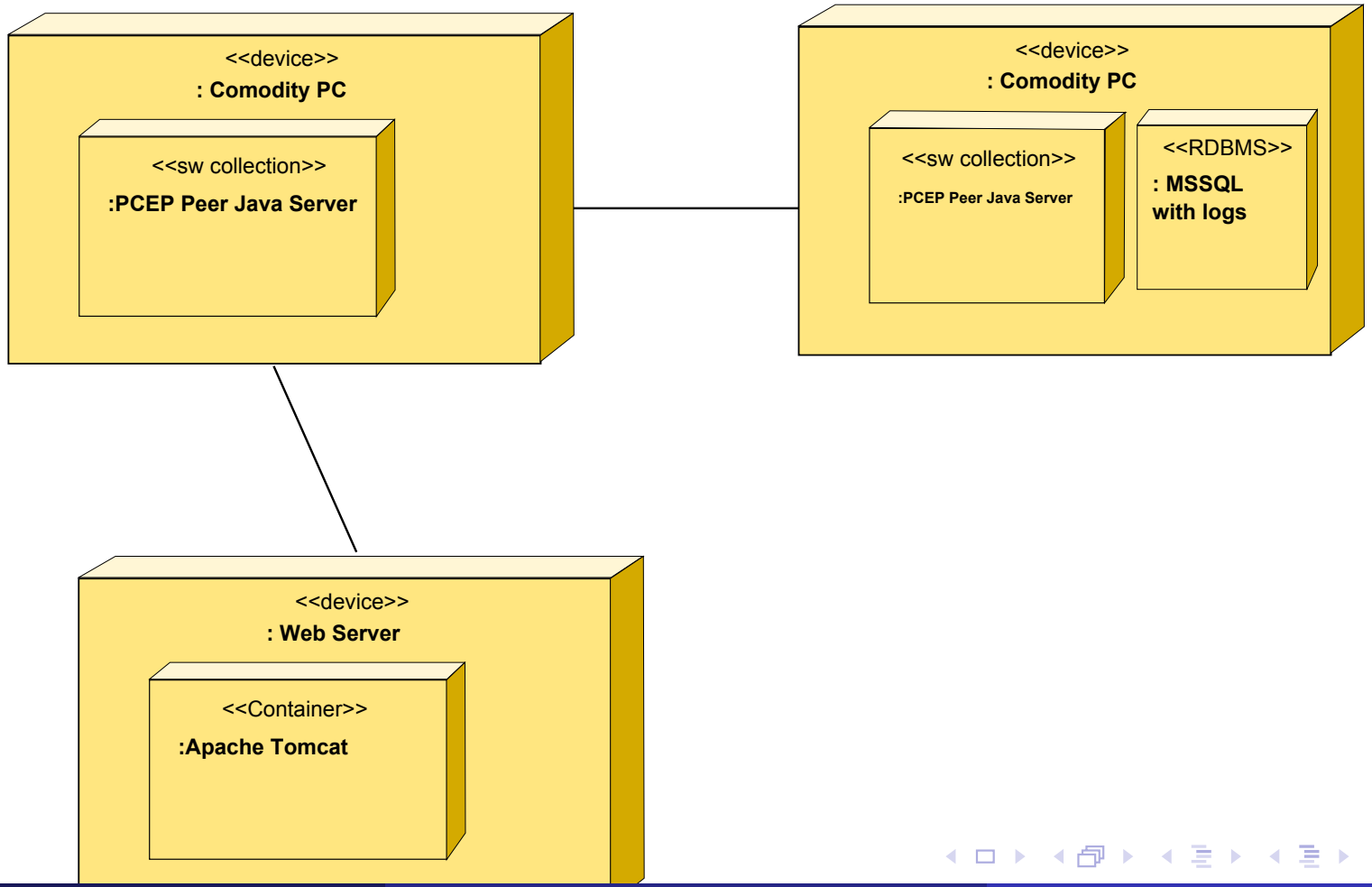
- Matching accuracy and memory consumption was monitored
- Monte Carlo performs better in regards to semantic power but suffers from memory consumption





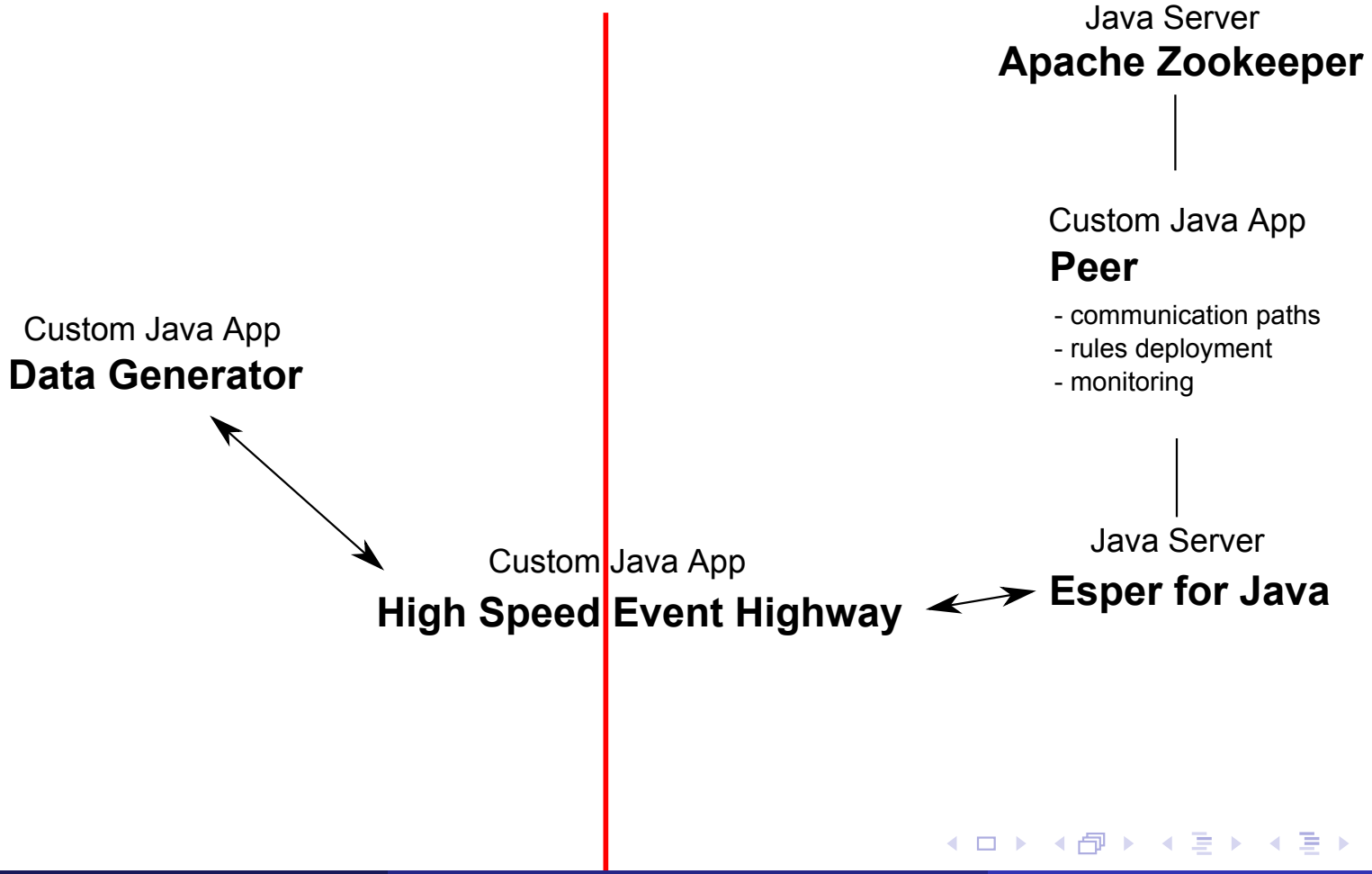
Implementation

- Peer will be a Java daemon
- One peer per one computing node
- Decentralized



PCEP Peer Java Server

- Apache Zookeeper for intragrid communication
- High speed event highway
- Completely Java based



Distributed CEP Experiments

- Virtualized environment (paravirtualization with XEN or KVM)
- Web visualization server accessible through REST API
- Generator of requests from event logs
- Experiment measurements
- Event Command Console

Benefits

- The overall goal will be performance in CEP scenarios
- Horizontal scalability will be attacked explicitly, metrics will be defined
- Insight how accuracy/performance trade-off manifests itself in real world scenarios
- Peer will use production ready CEP engine (Esper)
- System will be described formally

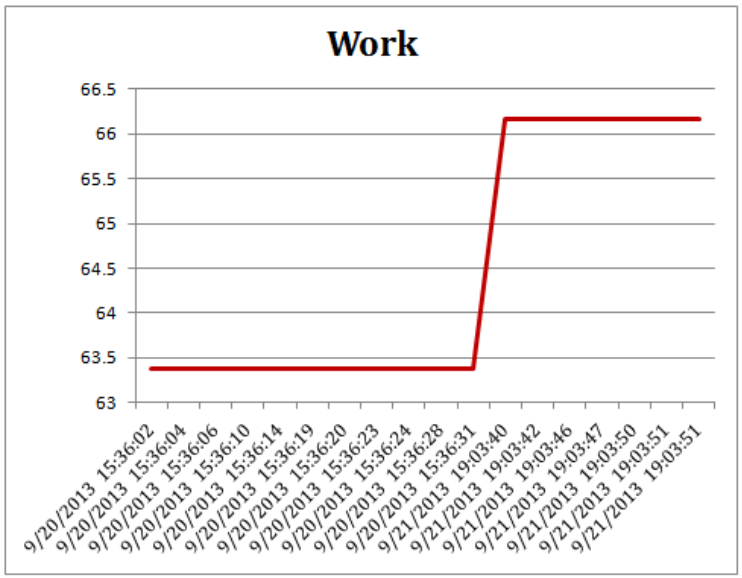
Outline

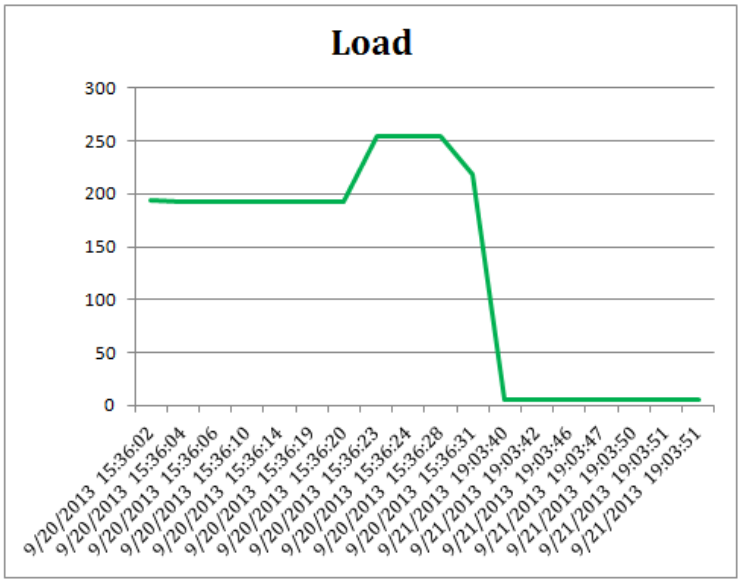
DEBS Community

- Distributed Event Based Systems since 2002
- Large research community in the area of CEP
- Industry oriented (Esper, IBM Infosphere, Drools Fusion)

DEBS Community

- Distributed Event Based Systems since 2002
- Large research community in the area of CEP
- Industry oriented (Esper, IBM Infosphere, Drools Fusion)
- Since 2011 accompanied with DEBS Grand Challenge
  - 2012 assignment is focused on high tech manufacturing where performance was the concern
  - 2014 assignment is focused to solve smart plugs queries on large real-life data set.





DEBS Grand Challenge Dataset

Recordings from 2125 plugs. 40 Houses were sampled for 1 month.

timestamp	timestamp of the measurement
value	32 bit float number
plug_id	unique (within household) id of a plug in a household
house_id	unique id of a house where the household is located
household_id	unique (in a house) identifier of a household
property	either 'work' or 'load'

Lasaris Dataset

We have datasets from smart meters but we choose to use security data or public data sets `http://ita.ee.lbl.gov/html/traces.html`

References I

Filip Nguyen and Tomáš Pitner.
Information System Monitoring and Notifications Using Complex Event Processing.
In *Proceedings of the Fifth Balkan Conference in Informatics*, BCI '12, pages 211–216, New York, NY, USA, 2012. ACM.

Filip Nguyen and Tomáš Pitner.
Scaling CEP to Infinity.
In *9th Summer School of Applied Informatics*, Brno, Czech Republic, 2012. Masaryk University.

Filip Nguyen, Daniel Tovarňák, and Tomáš Pitner.
Semantically Partitioned Peer to Peer Complex Event Processing.
In Filip Zavoral, Jason J. Jung, and Costin Badica, editors, *Intelligent Distributed Computing VII*, volume 511 of *Studies in Computational Intelligence*, pages 55–65. Springer International Publishing, 2014.

References II

Filip Nguyen and Jaroslav Škrabálek.
NotX Service Oriented Multi-platform Notification System.
In *2011 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 313–316, 2011.

Daniel Tovarňák, Filip Nguyen, and Tomáš Pitner.
Distributed Event-Driven Model for Intelligent Monitoring of Cloud Datacenters.
In Filip Zavoral, Jason J. Jung, and Costin Badica, editors, *Intelligent Distributed Computing VII*, volume 511 of *Studies in Computational Intelligence*, pages 87–92. Springer International Publishing, 2014.

Jaroslav Škrabálek, Petr Kunc, Filip Nguyen, and Tomáš Pitner.
Towards Effective Social Network System Implementation.
Control and Cybernetics, 41/2012, 2013.

References III

Jaroslav Škrabálek, Petr Kunc, Filip Nguyen, and Tomáš Pitner. Towards Effective Social Network System Implementation. In Mykola Pechenizkiy and Marek Wojciechowski, editors, *New Trends in Databases and Information Systems*, volume 185 of *Advances in Intelligent Systems and Computing*, pages 327–336. Springer Berlin Heidelberg, 2013.