Universidade Federal
de Campina Grande

TECHNISCHE
UNIVERSITÄT
DRESDEN

*Stream*MapReduce

When Stream Processing crosses MapReduce

***André Martin**[*], **Andrey Brito**[#] and **Christof Fetzer**[*]*

andre.martin@tu-dresden.de, andrey@dsc.ufcg.edu.br, christof.fetzer@tu-dresden.de

[*]Technische Universität Dresden - Dresden, Germany

[#]Universidade Federal de Campina Grande - Campina Grande, Brazil

STREAM 2015 @ October, 28th 2015, Indianapolis, IN

Outline

- Why StreamMapReduce?
  - Programming Model
  - Application Examples
- Fault Tolerance & Elasticity in SMR
- Conclusion & Summary
- Ongoing & Future Work

Why StreamMapReduce?

- Massive data growth
  - RFDI, click streams, net flow data, credit card transactions
  - user tracking, recommendations, intrusion and fraud det.
- Large scale data processing got very popular
  - Google’s MapReduce is **very simple**
  - Mature open source implementation **hadoop**
- Batch processing ⇨ **Real time data processing**

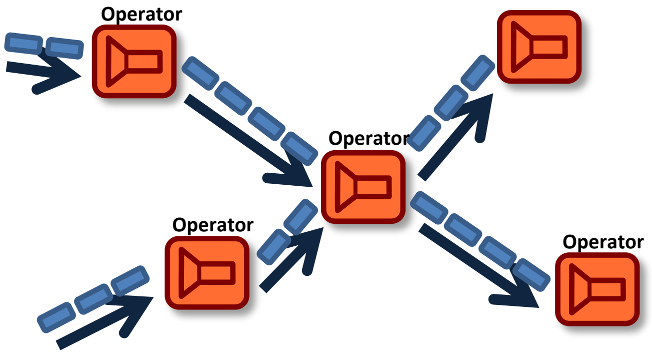
STORM samza S4 distributed stream computing platform StreamMine3G

StreamMapReduce

PROGRAMING MODEL

Programming Model

- Inspired by MapReduce (UDFs)
- ***Stateful*** event processing



Event

State

```
public class SomeOperator extends OperatorImpl
{
    public void process(int routingKey, byte[] event, Object state,
          Collector collector)
    {

          /* some computation */
          collector.emitEvent(someKey, timestamp, event.getBytes());
    }
    ...
```

Collector

Programming Model - Properties

- We consider everything as a **black box**
  - State (simple ptr to some arbitrary data structure)
  - Operator Code = User Defined Function
  - Event (except timestamp & routing key)

StreamMapReduce

APPLICATION EXAMPLES

Application Examples

- ACM DEBS’15 Challenge: Geo-spatial data context / taxi rides NYC area:
  - **Query #1:** Provide top-10 most frequently driven routes using a 30mins sliding window
  - **Query #2:** Provide top-10 most profitable areas using a 15 & 30 mins sliding window

Application Examples - Properties

- Access to specific **historic data**
- Complex computation, e.g. **profitable areas**:
  - accumulated fares / # of empty taxis
  - # of empty taxis = # taxis w/o follow-up pickup
- "Conditional" output
  - Only generate output if top-k changes

Very complicated to express in CQL!

StreamMapReduce

FAULT TOLERANCE & ELASTICITY

Fault Tolerance & Elasticity

- **State persistence:**
  - User provides serialize & de-serialize methods
- **Event logging** (in memory)**:**
  - Timestamps associated w/ events
- *Repeatability:*
  - Deterministic merge of multiple up-streams

$\Rightarrow$ Provide **rollback recovery** in a precise manner

$\Rightarrow$ **Active replication** w/o costly atomic broadcast (state machine replication)

- Re-use FT mechanisms for **elasticity** [1]

[1] Raul Castro Fernandez, MatteoMigliavacca, Evangelia Kalyvianaki, and Peter Pietzuch. **Integrating scale out and fault tolerance in stream processing using operator statemanagement.** In *Proceedings of the 2013 ACMSIGMOD International Conference on Management of Data*, SIGMOD ’13, pages 725–736, New York, NY, USA, 2013. ACM

Black box vs. Performance

- Use flat data structures for state [1]

- User annotations for operators, e.g.
  - stateful vs. stateless
  - commutativity [2] or
  - window lengths [3]

[1] YongChul Kwon,Magdalena Balazinska, and Albert Greenberg. **Fault-tolerant streamprocessing using a distributed, replicated file system***, VLDB Endow.*, 1(1):574–585, August 2008
[2] AndréMartin, Thomas Knauth, Stephan Creutz, Diogo Becker, StefanWeigert, Christof Fetzer, and Andrey Brito. **Low-overhead fault tolerance for highthroughput data processing systems.** In *Proceedings of the 2011 31st International Conference on Distributed Computing Systems*, ICDCS '11, pages 689–699, Washington, DC, USA, 2011. IEEE Computer Society.
[3] Zoe Sebepou and Kostas Magoutis. **Cec: Continuous eventual checkpointing for data stream processing operators.** In *Proceedings of the 2011 IEEE/IFIP 41st International Conference on Dependable Systems&Networks*, DSN '11, pages 145–156,Washington, DC, USA, 2011. IEEE Computer Society.

Summary & Conclusions

- SMR allows a **quick and easy transition** from batch to online processing of MapReduce applications
- Simplifies the development of complex queries and operators
- Good foundation for higher-level abstractions such as CQL/SQL [1] or K3
- Fault tolerance through check-pointing and logging
- Re-use of fault tolerance mechanisms to achieve elasticity

[1] Thomas Heinze, Zbigniew Jerzak, AndréMartin, Lenar Yazdanov, and Christof Fetzer. **Fault-tolerant complex event processing using customizable state machine-based operators.** In *Proceedings of the 15th International Conference on Extending Database Technology*, EDBT ’12, pages 590–593, New York, NY, USA, 2012. ACM.

On-Going & Future Work

- Privacy preserving ESP using **Intel SGX**
  - Putting the whole system vs. operators into the enclave

- Integration of StreamMine3G into the cloud stack using **OpenStack Sahara**
  - Apache Storm integration already existing

Thank you for your attention – Q&A

andre.martin@se.inf.tu-dresden.de