# MERA:
# Trading Precision for Performance


Niklas Semmler (PhD student)

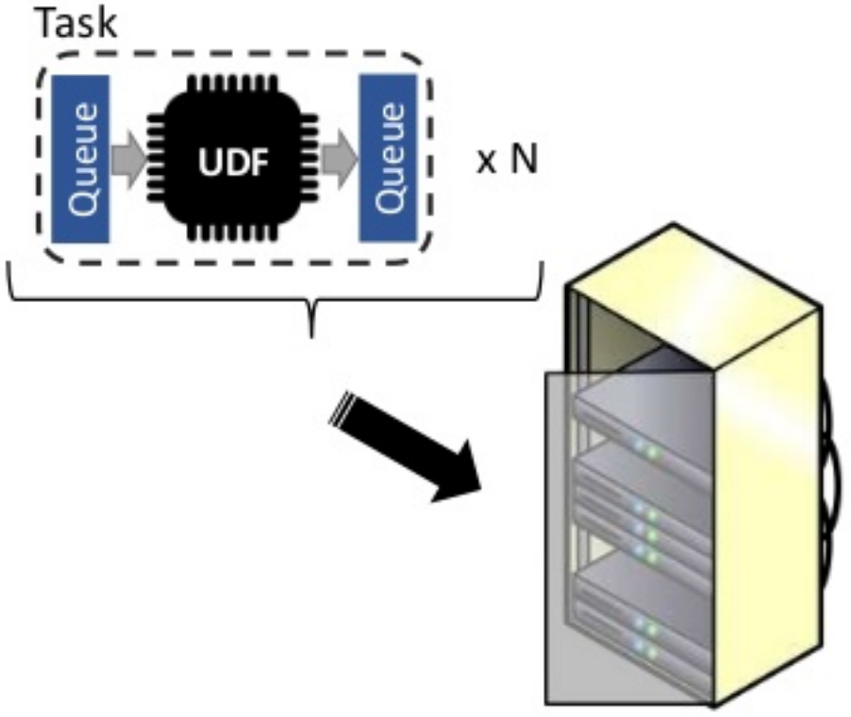
Anja Feldmann (Supervisor)

*The Why:* Compute function chains over large datasets.

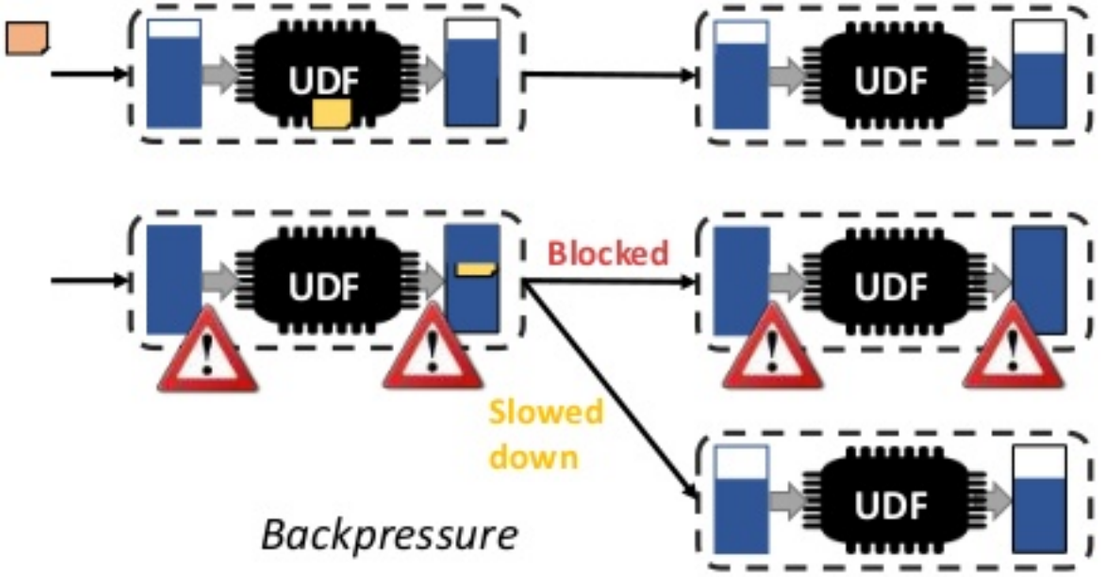
Fast


Cost-efficient

Accurate

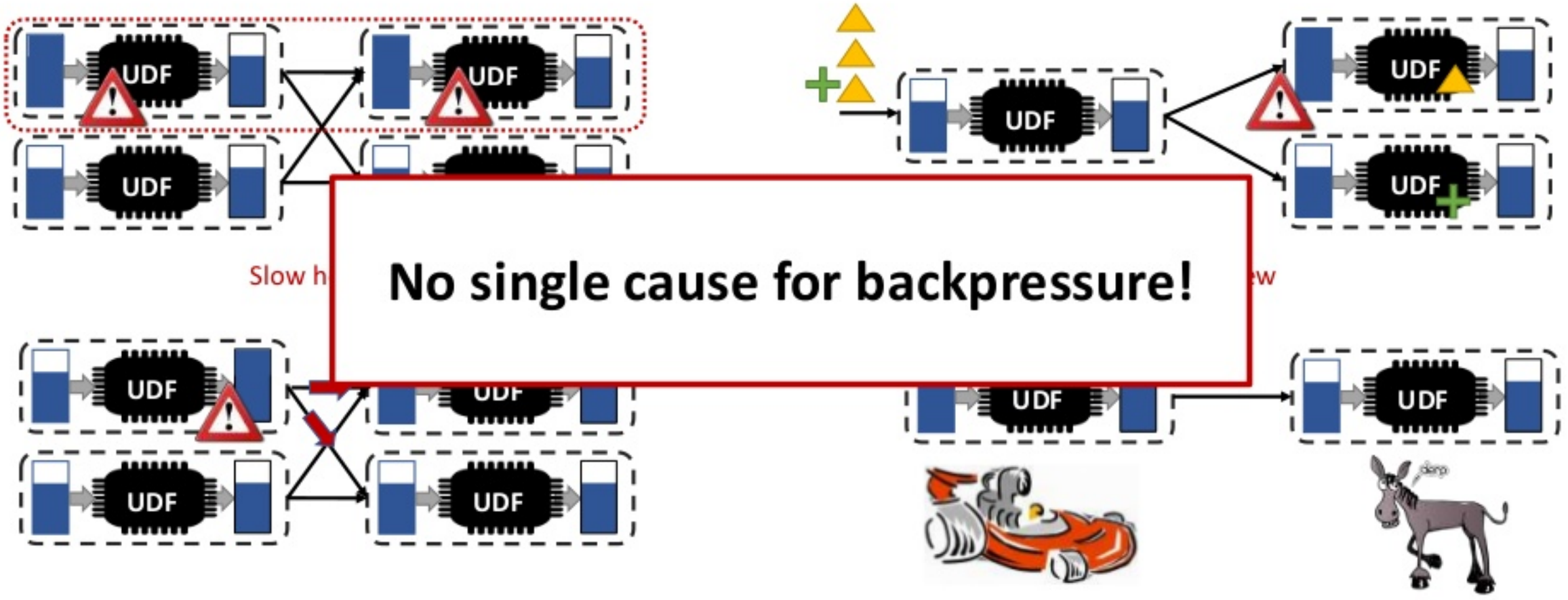



Jobs are deployed on commodity hardware




Commodity hardware is failure-prone

# Causes for Backpressure



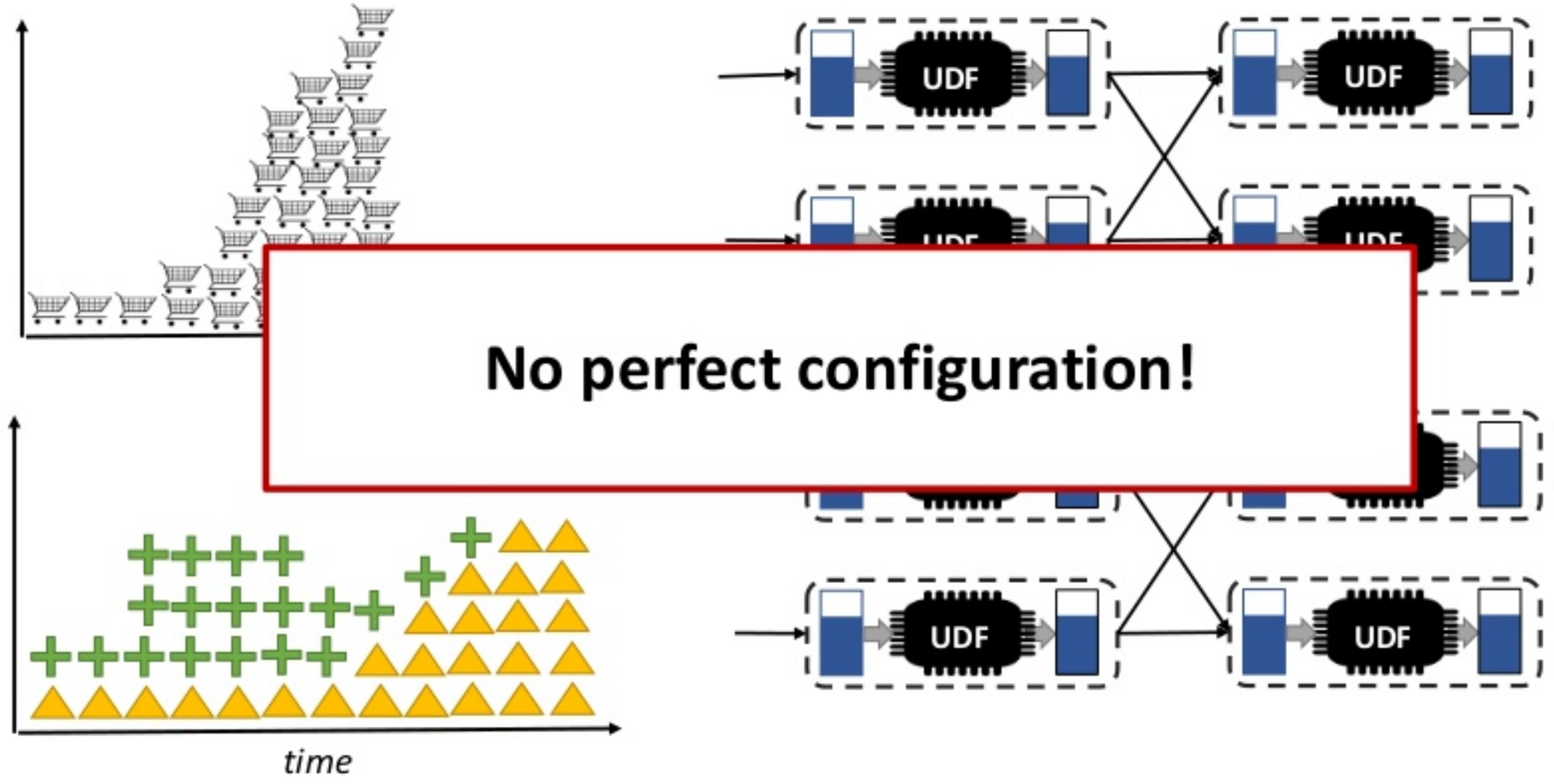
# Prior Knowledge: Straggler Mitigation

- Learn over iterations to identify the best configuration options?
  - *Herodotou, Herodotos, et al. "Starfish: a self-tuning system for big data analytics." Cidr. Vol. 11. No. 2011. 2011.*

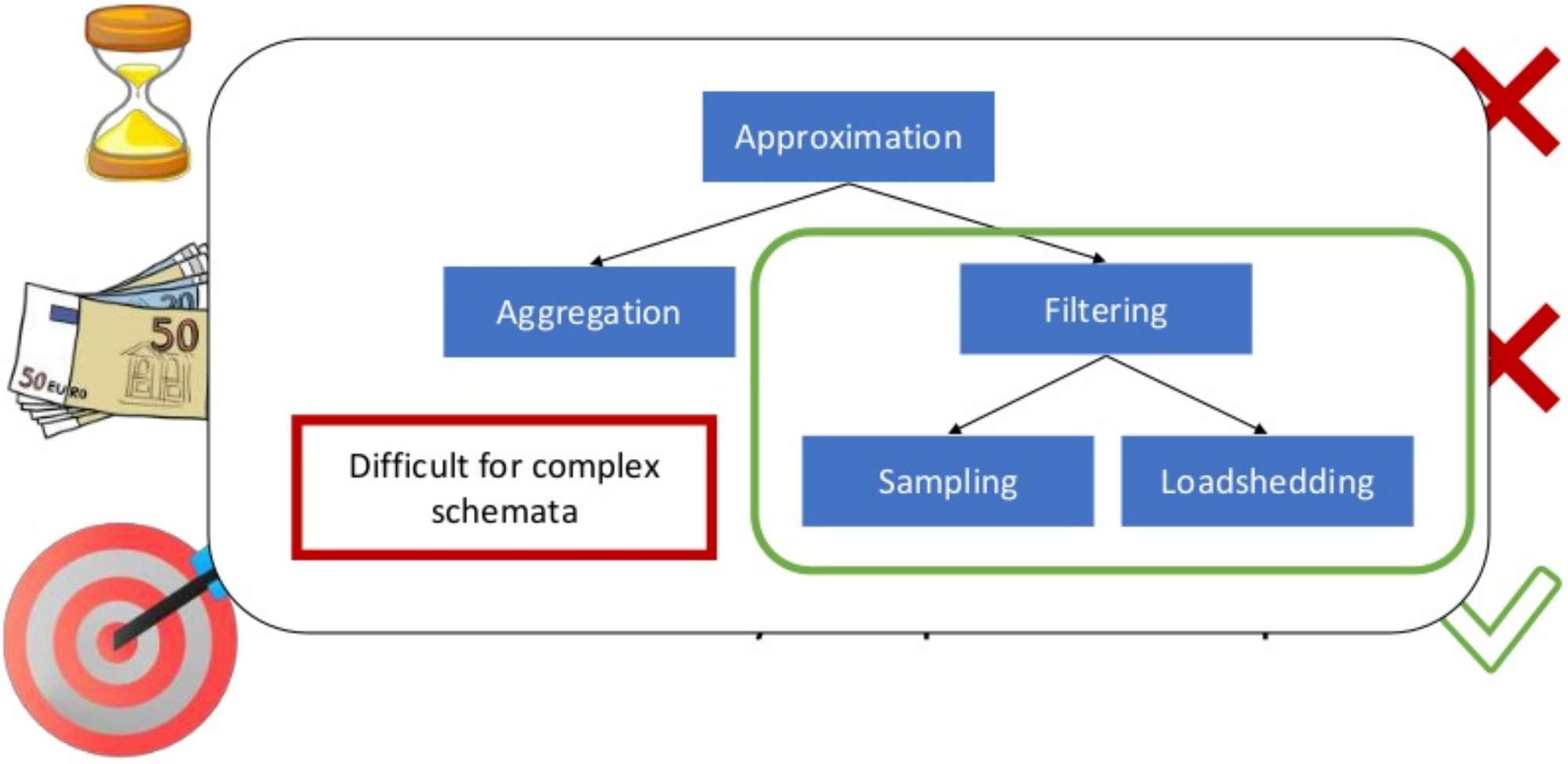
- Use speculative execution, i.e. execute the same task multiple times in parallel, and use the result from the fastest tasks.
  - *Ananthanarayanan, Ganesh, et al. "Effective Straggler Mitigation: Attack of the Clones." NSDI. Vol. 13. 2013.*

**Does not work for streaming!**

# Data Streams are different



# Methods to deal with Backpressure



# Filter - Sampling



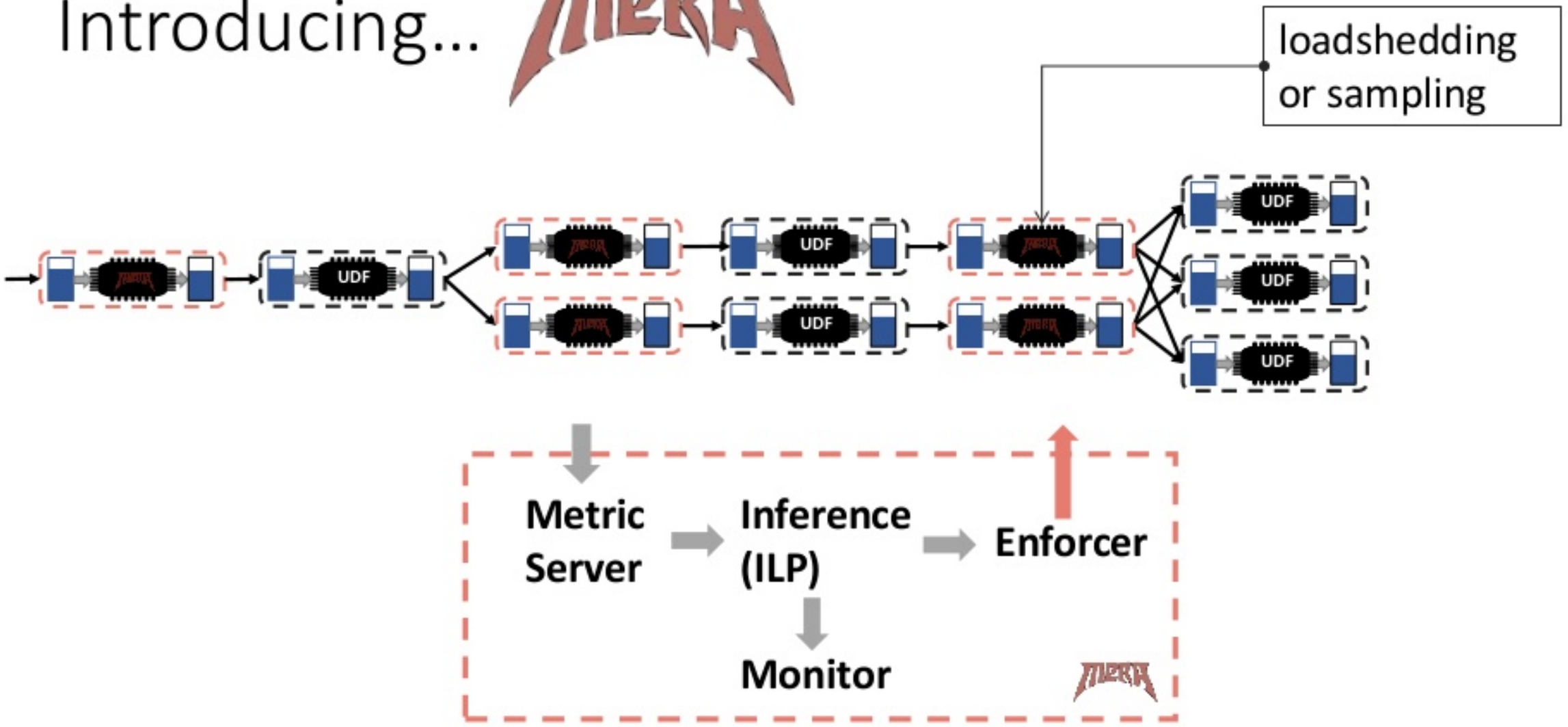
StreamApprox presented by *Pramod Bhatotia* and *Do Le Quoc* at FlinkForward'17

Generate a representative subset

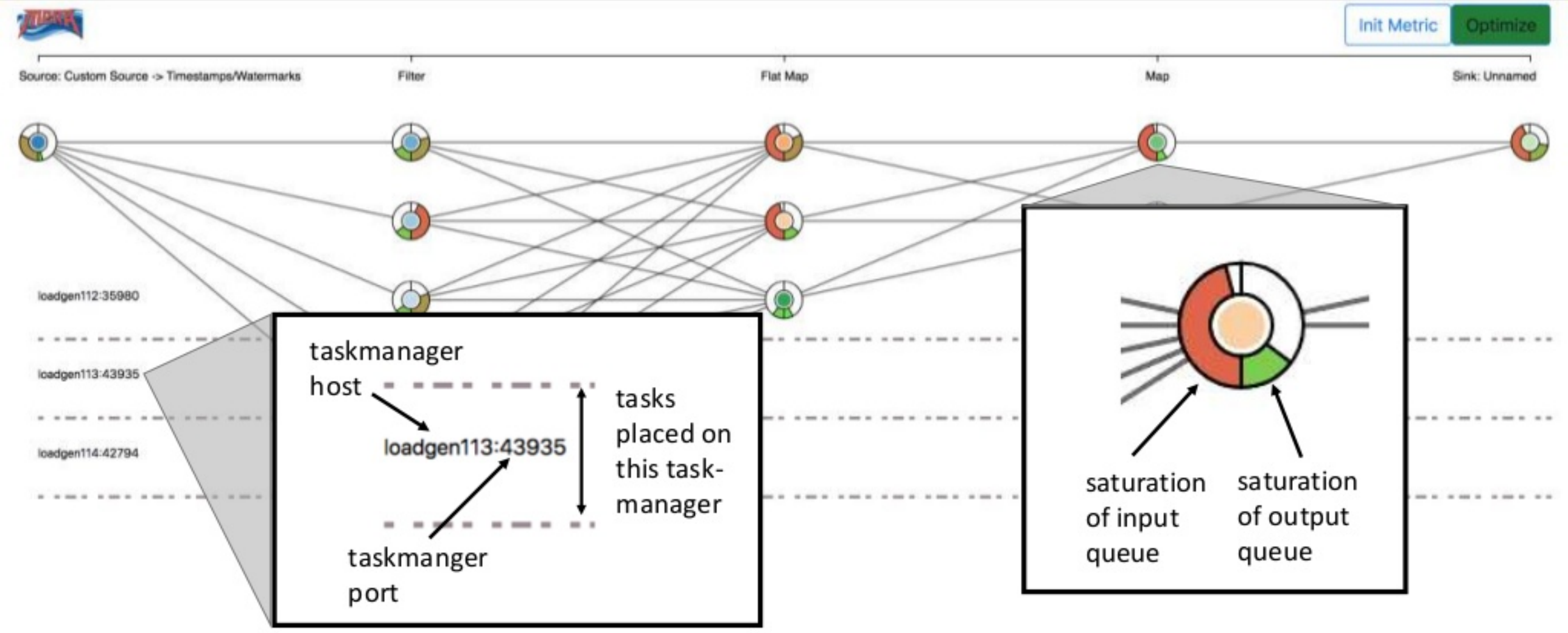
# Filter - Loadshedding


Generate a subset fulfilling a given property

# Introducing… MERA

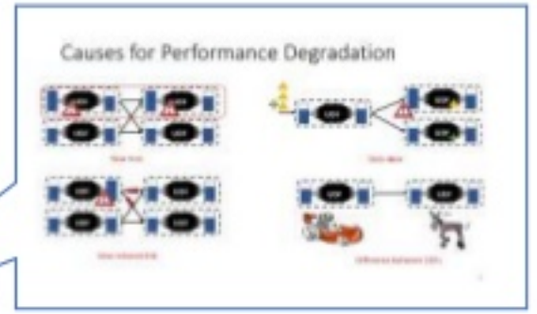


# MERA - Monitor

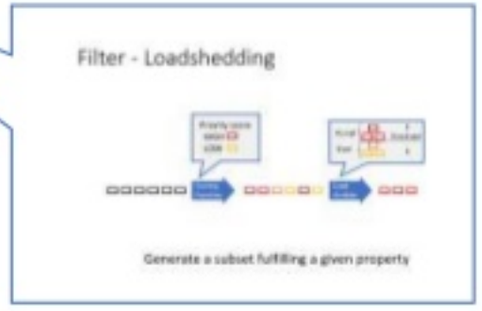


# Insights

No provisioning & state migration!




- Dynamic approximation is faster than scaling
- Backpressure needs to be handled everywhere at the job!
  - Filtering at the source is sub optimal!
- There is more to approximation than sampling!

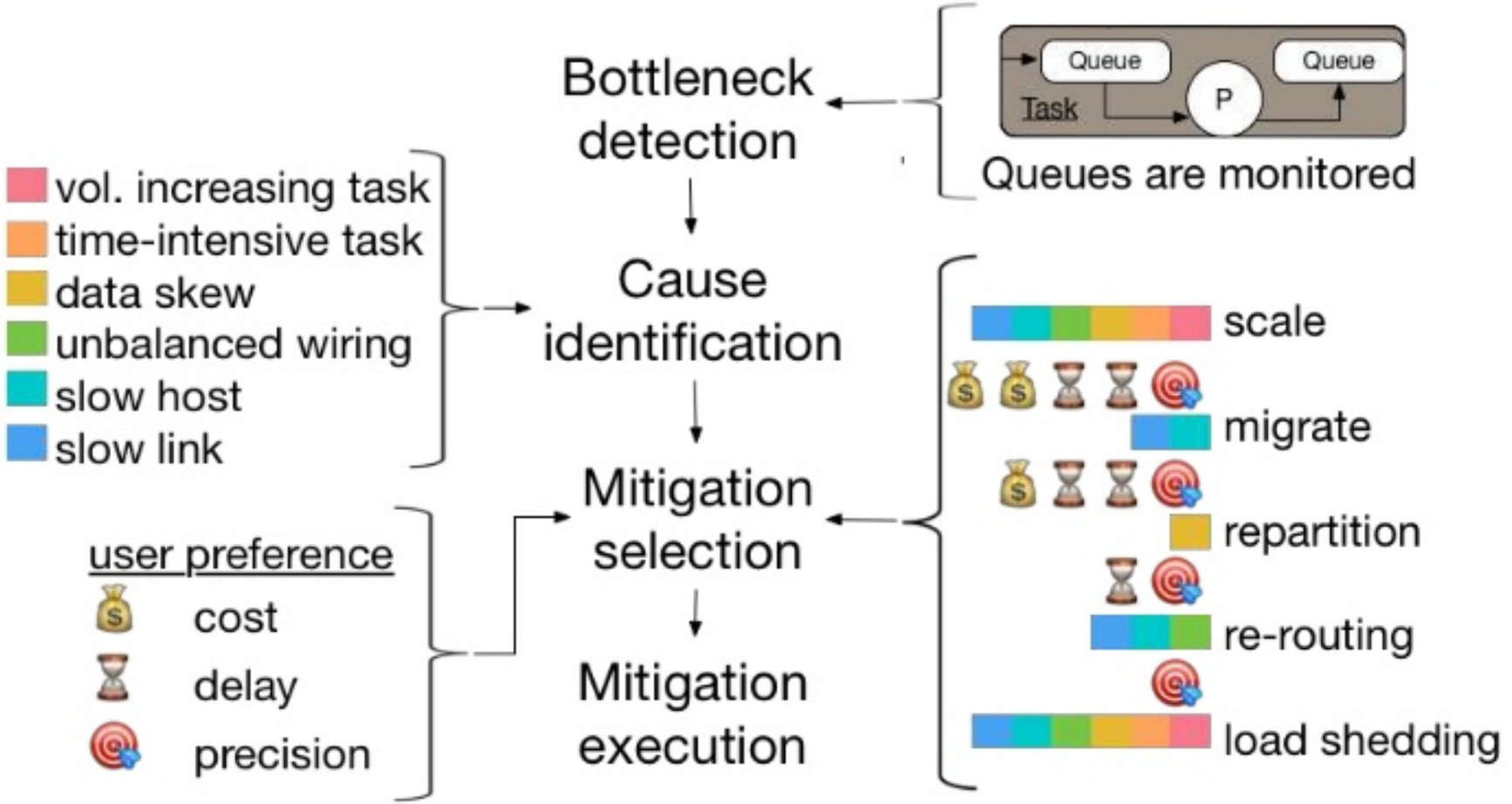



**Contact:** niklas@inet.tu-berlin.de

# Results

- No optimization vs optimization
  - Bottleneck latency
  - Input rate
  - Output rate
  - Drop rate
  - Latency?
- Drop only at source or everywhere
- Bar charts
  - Bottleneck Latency
  - Latency Mean
  - Without Opt/Only at Source/Full

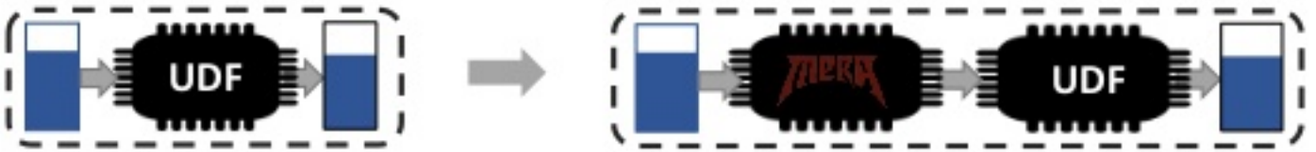
# Future


Bottleneck detection
Queue
Task
P
Queue
Queues are monitored
vol. increasing task
time-intensive task
data skew
unbalanced wiring
slow host
slow link
Cause identification
Mitigation selection
Mitigation execution
user preference
cost
delay
precision
scale
migrate
repartition
re-routing
load shedding

# Re-Read

- Load Shedding Techniques for Data Stream Systems
- STREAMAPPROX: APPROXIMATE STREAM ANALYTICS IN APACHE FLINK
- Approximation papers

**Future**




*Combine aggregation with task*

# Open issues

- Problem: The ILP will select only a single or only few places to shed items, leading to a loss of high-scoring items.
  - Fix: Include the distribution factor into the ILP.
- Problem: By shedding at reducers, the distribution of data will change in one direction.
  - Fix: Shed only at mappers