

Online Replication Strategies for Distributed Data Stores

Niklas Semmler
SAP SE

Georgios Smaragdakis
TU Berlin

Anja Feldmann
MPI for Informatics

Industrial IoT

Transport/Distribution



Energy/Utilities

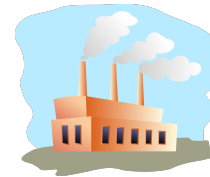


Manufacturing



Industrial IoT: Characteristics

High number of co-located machines



Factories, wind farms, airports

Each machine is equipped with increasing number of sensors.



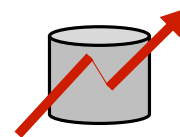
Industrial robots, wind wheels, airplanes, etc.

Continuously creates immutable device records.

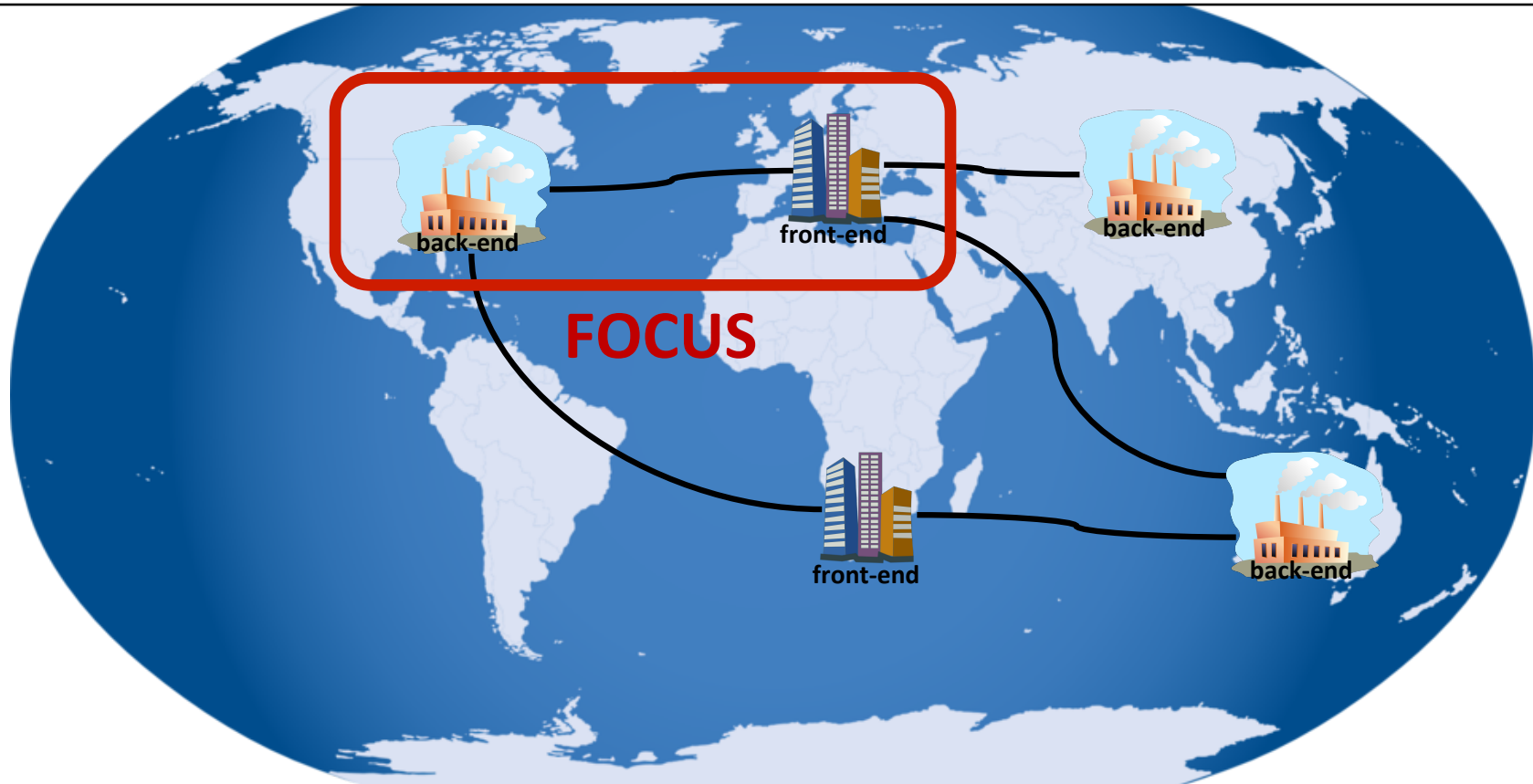


Position, movement, temperature, etc.

Mega-datasets emerge.

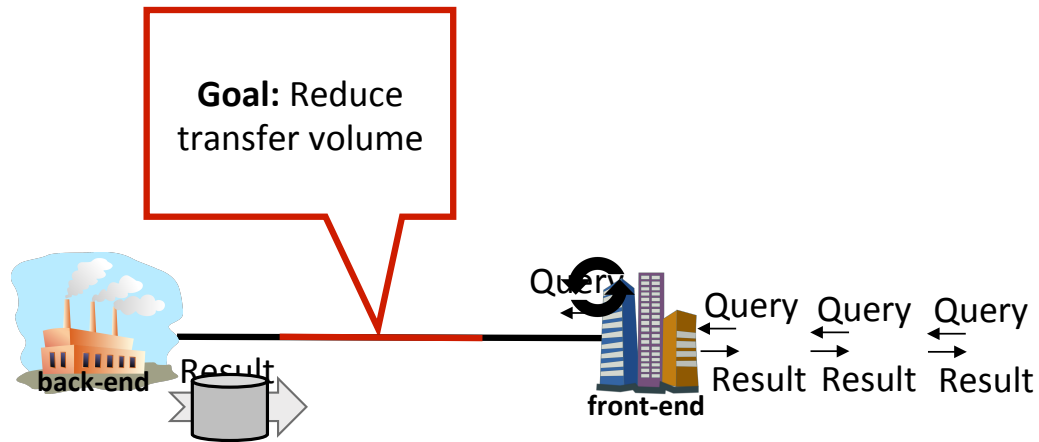


Local silos of IoT data.



1. Back-ends collect & aggregate data.
2. Front-ends access & join data.
3. Links between sites become a bottleneck.

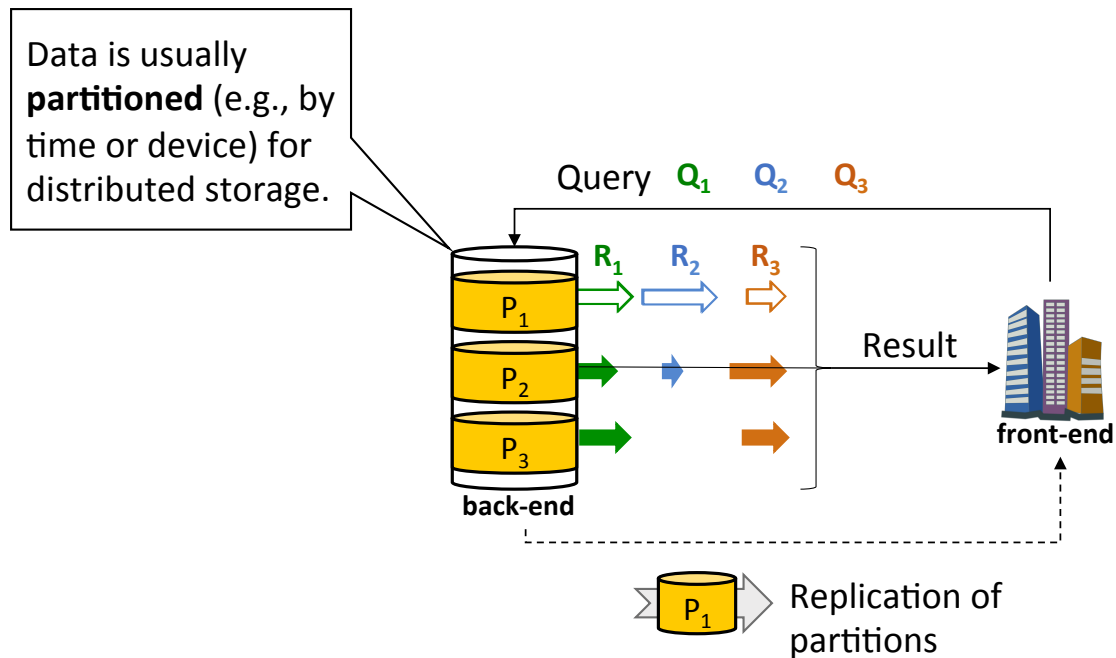
Minimizing data transfers



<i>Data Exchange Mechanisms</i>	Only send relevant data	No redundant transfers
<u>Option 1</u> Ship results	✓	✗
<u>Option 2</u> Replicate everything	✗	✓

Can we get the benefits of both?

Minimizing data transfers

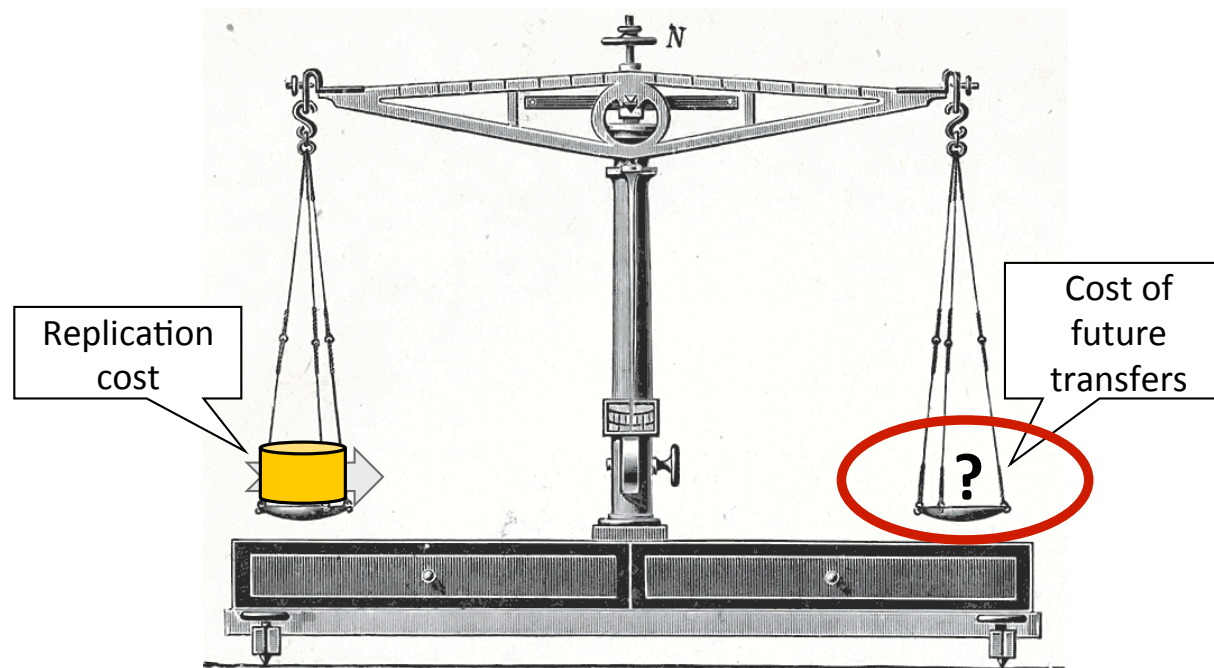


Skewed Popularity of IoT Data

*Most IoT data → rarely used
Some data → highly popular*

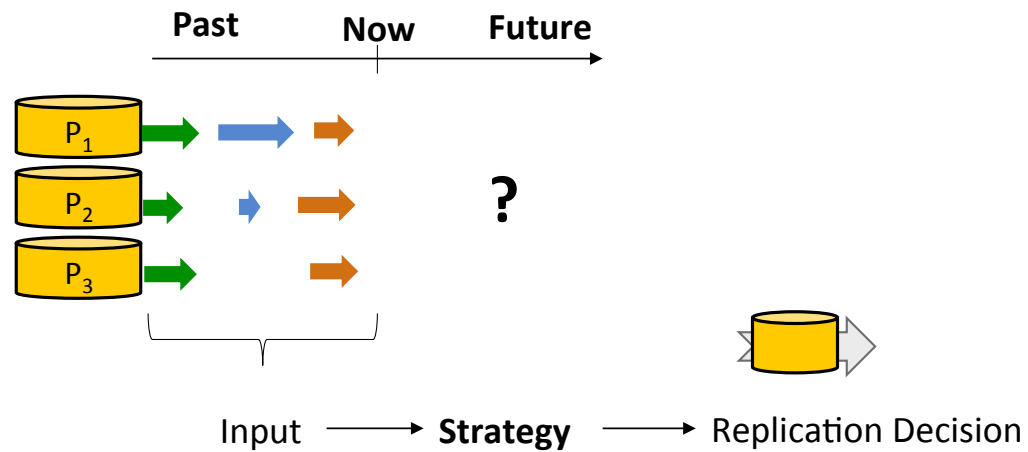
Can we replicate only popular partitions?

Problem: When should we replicate?

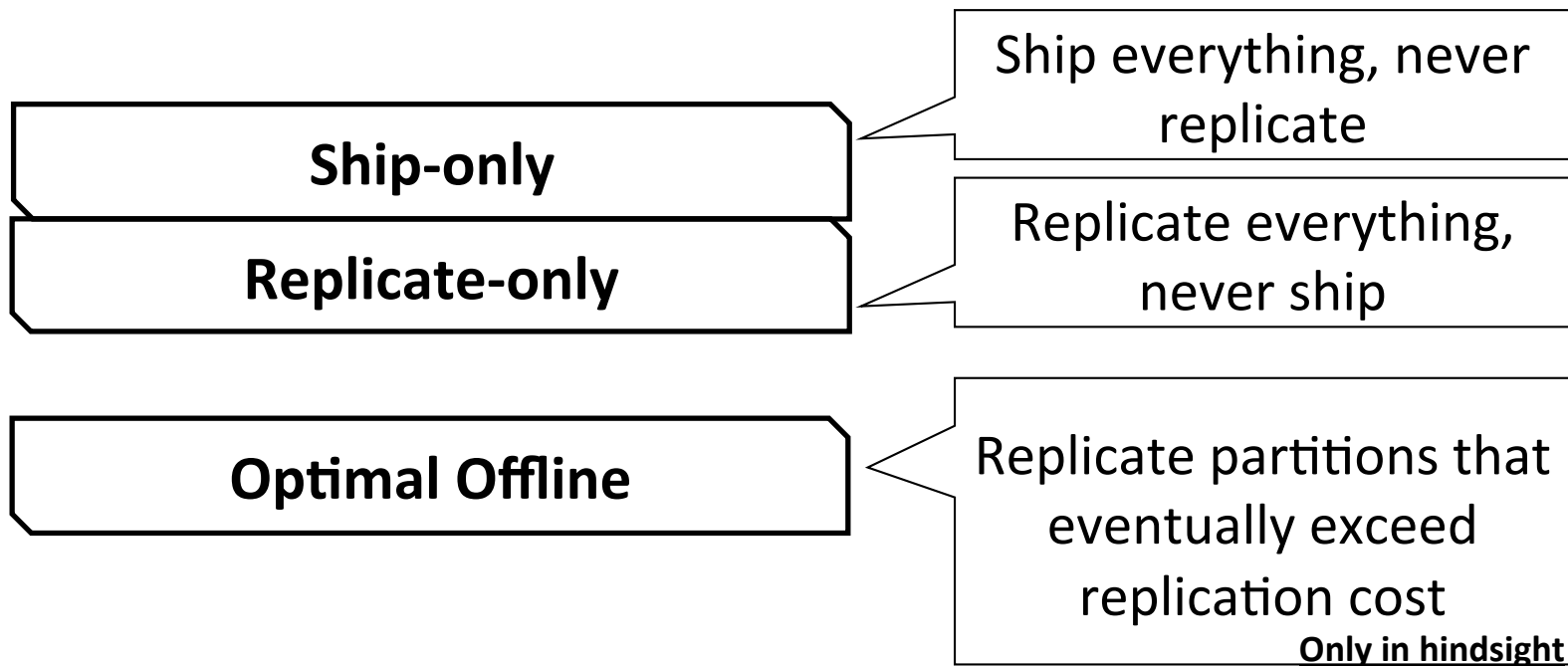


No knowledge of the future!

Online Replication Strategies

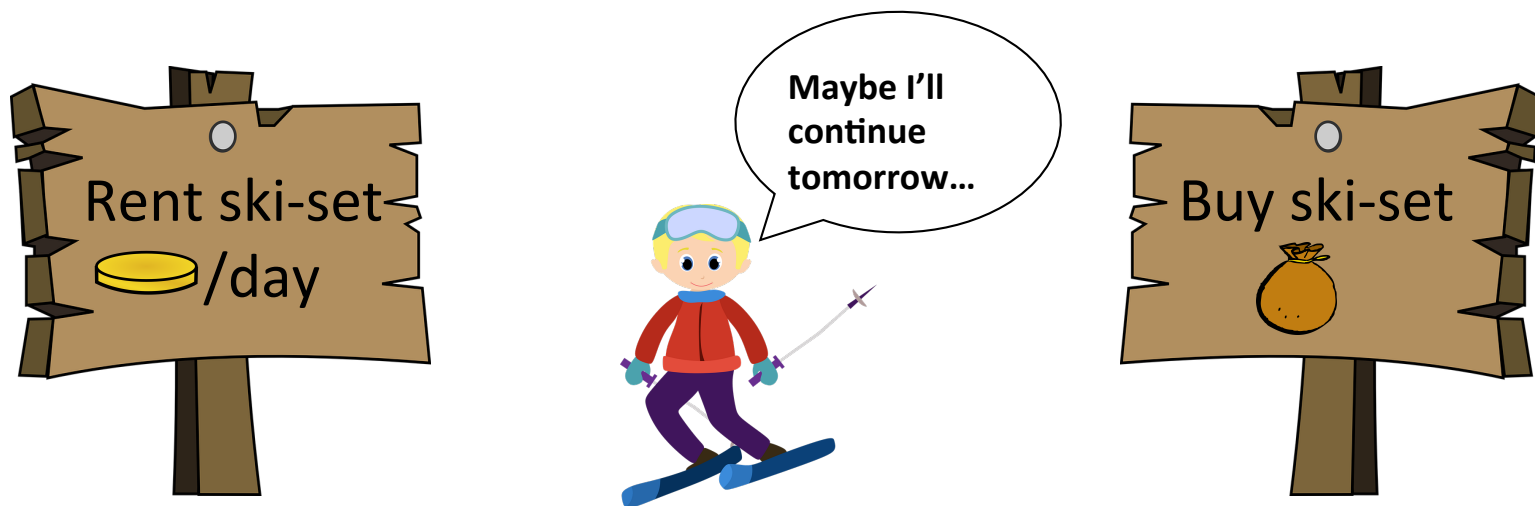


Online replication Strategies



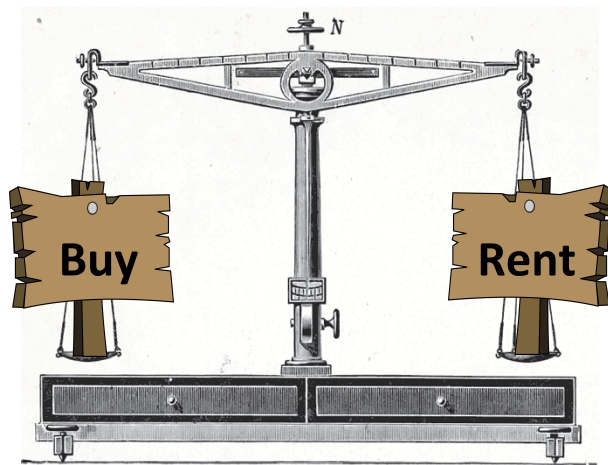
Can we find a better strategy?

Inspiration: Ski rental problem



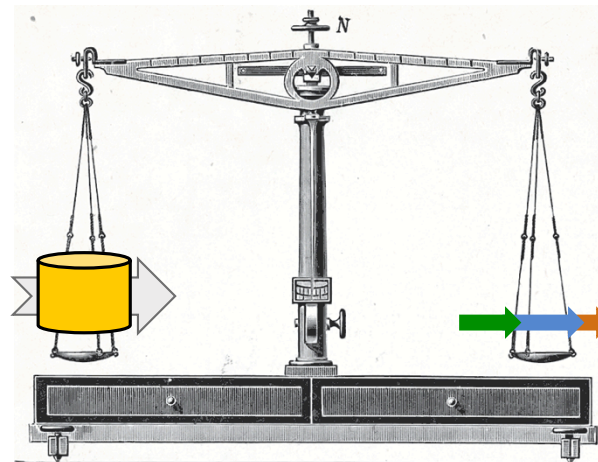
„Competitive snoopy caching“ by Karlin et al. (Algorithmica'88)

Inspiration: Ski rental problem



High one-time Low continuous

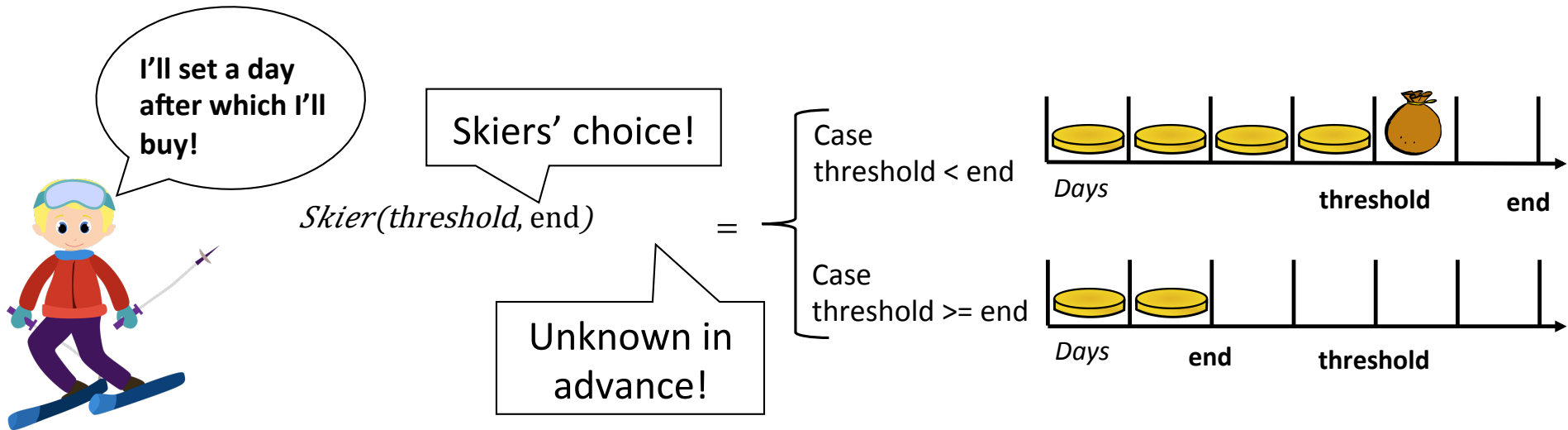
=



High one-time Low continuous

Skiing Days \neq Transfer Volume

Inspiration: Ski rental problem



- Optimal threshold?
 - No single best threshold.
 - Solutions for competitive ratio and expected cost.

Ski rental problem: Competitive ratio

I'll set a day after which I'll buy!



$$\text{Competitive Ratio} = \frac{\max_{end} \text{Skier}(\text{threshold}, \text{end})}{\text{Oracle}(\text{end})}$$

$$\text{Oracle}(\text{end}) = \min \left(\begin{array}{l} \text{Rent forever} \\ \text{or} \\ \text{Buy directly} \end{array} \right)$$

Rent till the cost of buying is reached.

Best threshold =



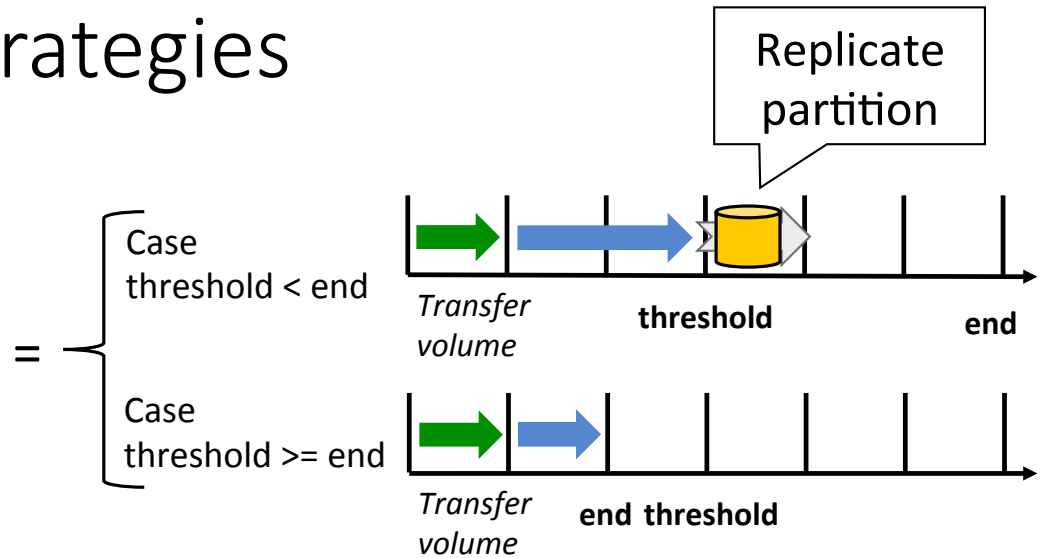
Only 2x worse than oracle.

Online replication strategies

ThresholdStrategy(threshold, end)



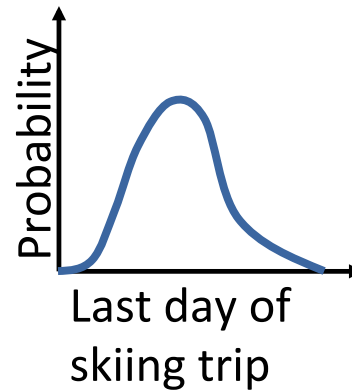
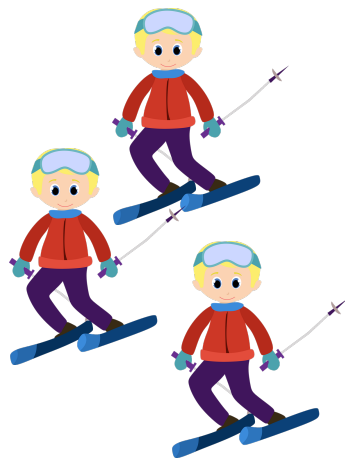
Ski-Rental



ThresholdStrategy(replicationCost, end)

Ski rental problem: Expected Cost

- Probability distribution of the last day of the skiing trip



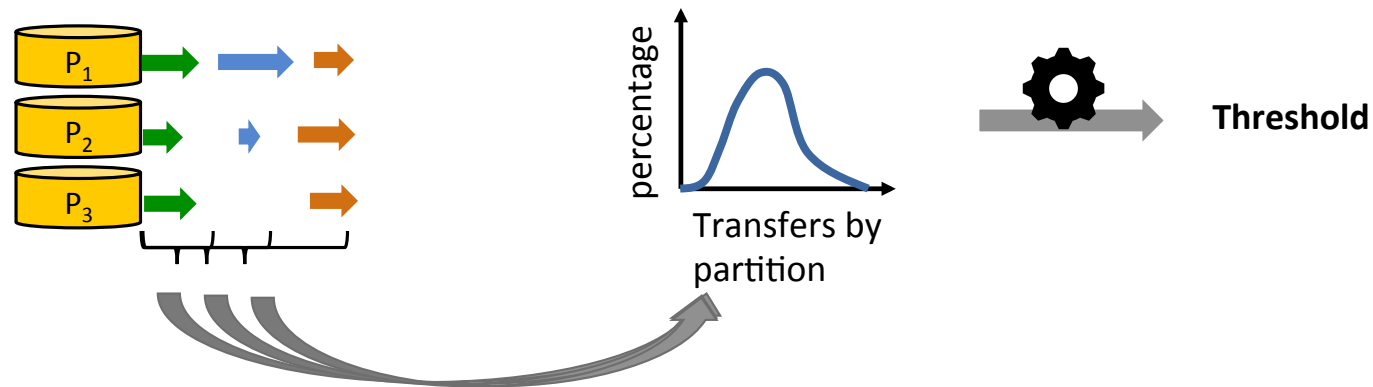
Probability of skiing trip length t

Expected Cost

$$= \sum_{end} \uparrow \text{Skier}(threshold, end) * p[end]$$

Online replication strategies

- We do not have a probability distribution.
 - But, we can iteratively approximate the distribution.



Reactive-Threshold

Online replication strategies

Ski-Rental

Static: Single threshold over time

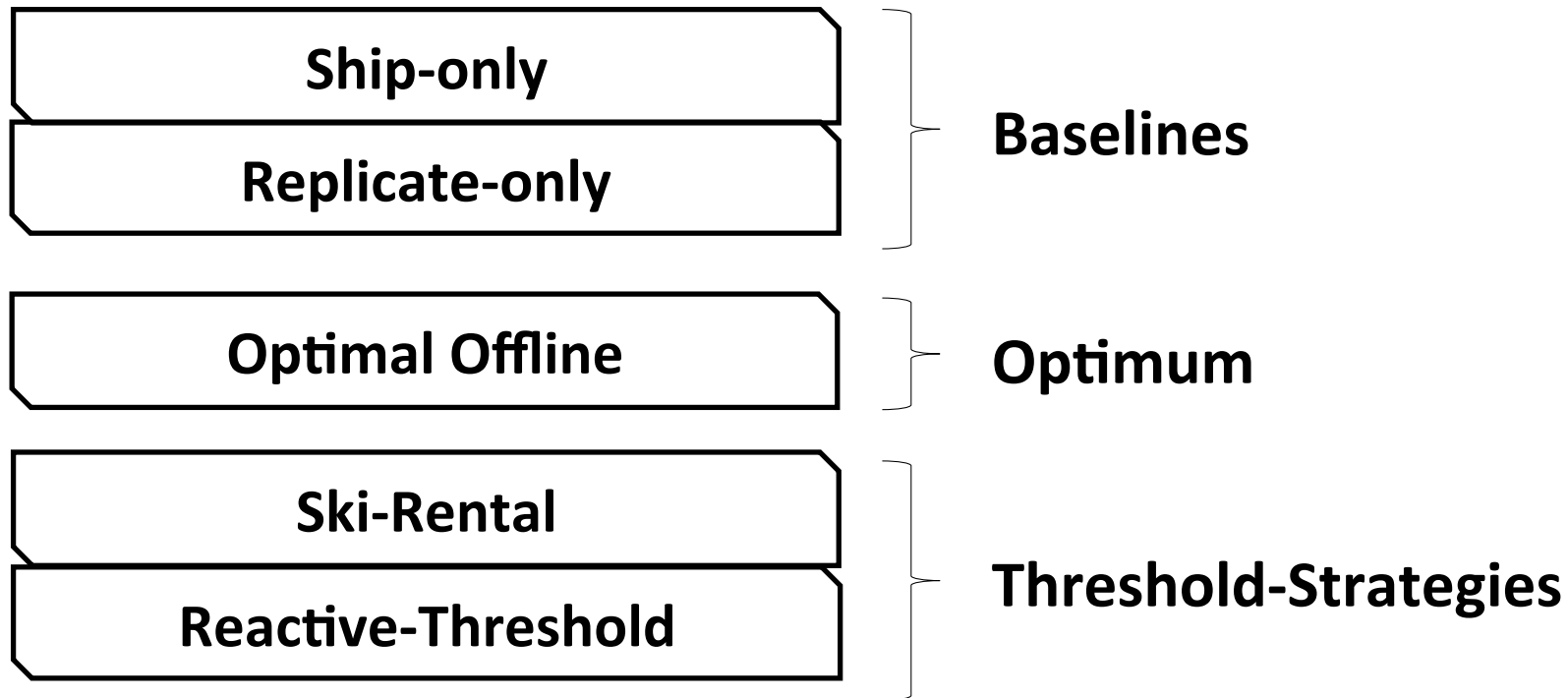
Expectation

Dynamic should work better!

Reactive-Threshold

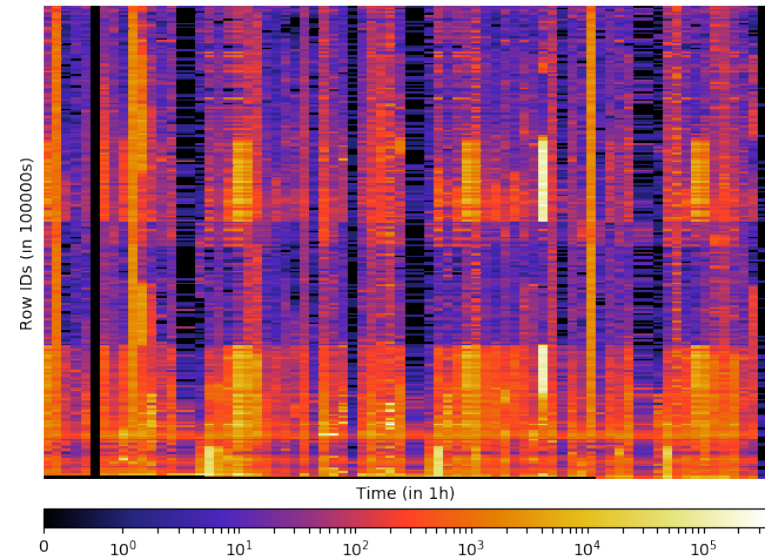
Dynamic: Threshold adapts over time

Online replication Strategies



Dataset

- Two query traces from an ERP database of a Globe 2000 company.
- Both traces contain accesses to rows of relational tables over time.
- Duration: 3 days / 72 hours of trace

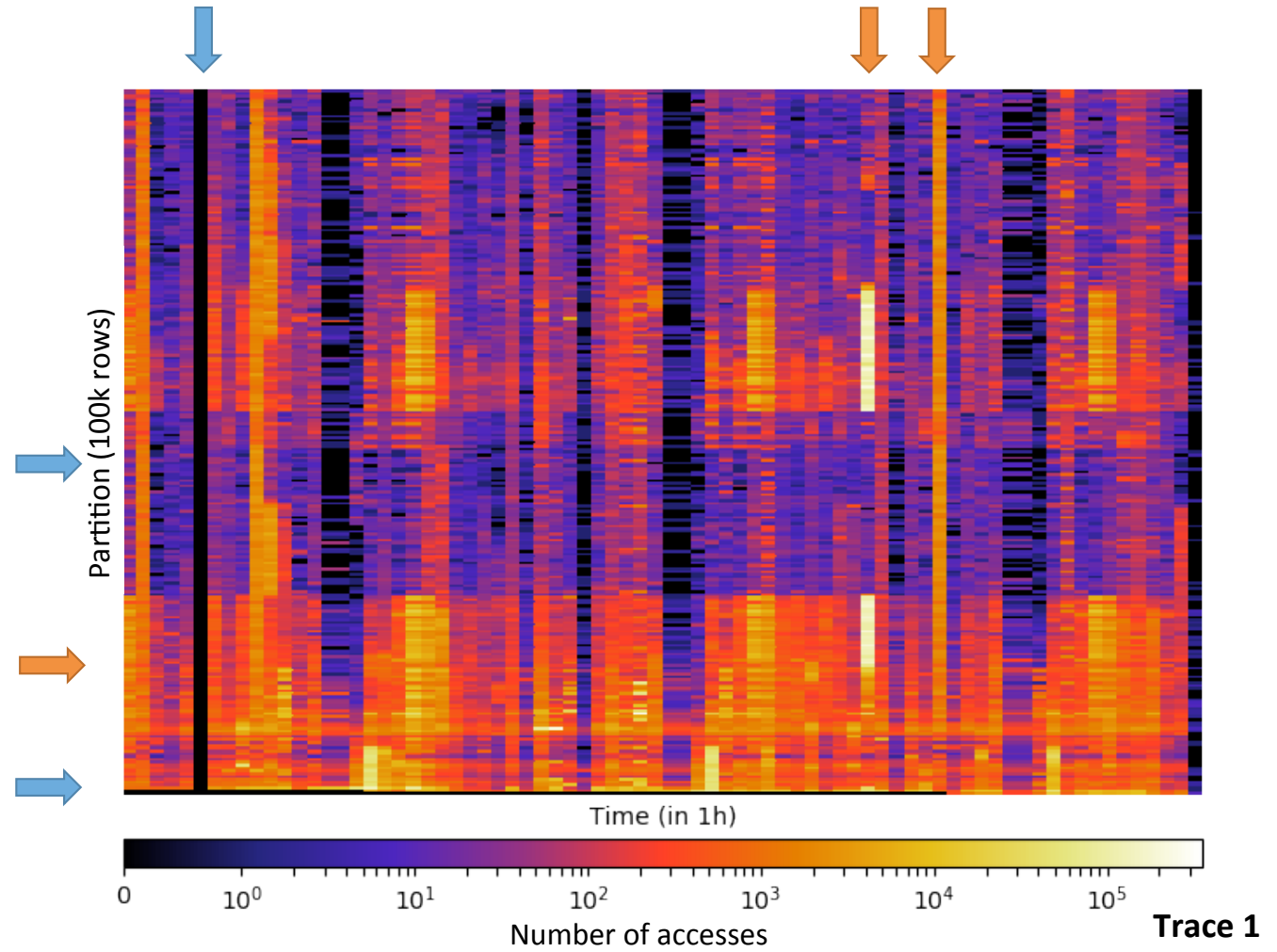


“Analyzing data relevance and access patterns of live production database systems” by Martin Boissier et al. (CIKM’16)

Dataset

Regions of very **LOW** interest

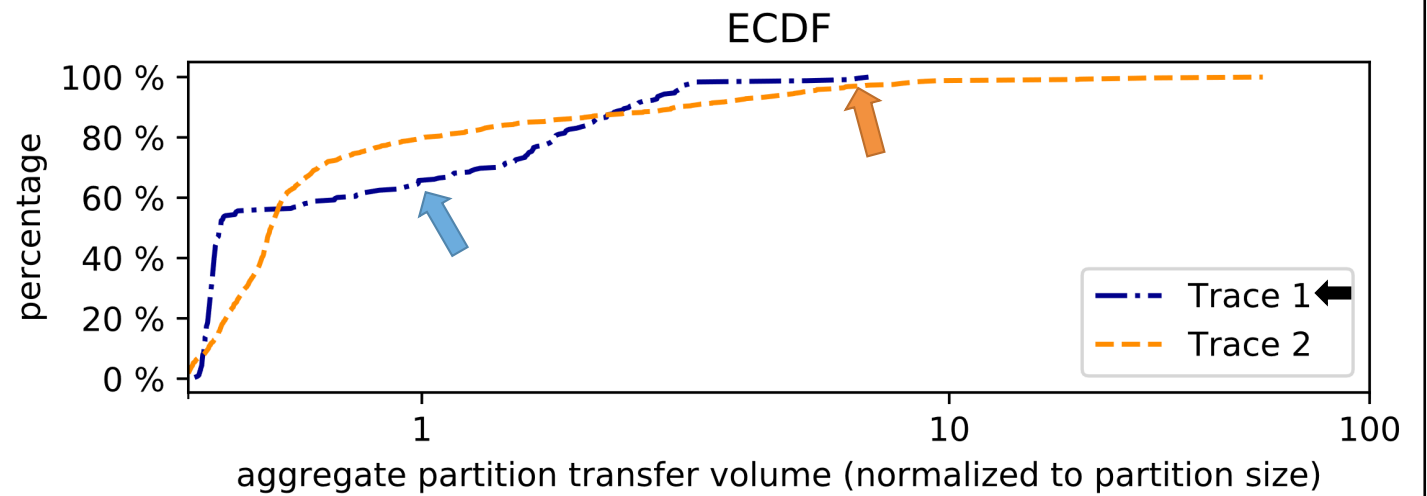
Regions of very **HIGH** interest



Dataset

~60 % transfer a volume lower than partition size

~1 % transfer a volume greater than 5x their size



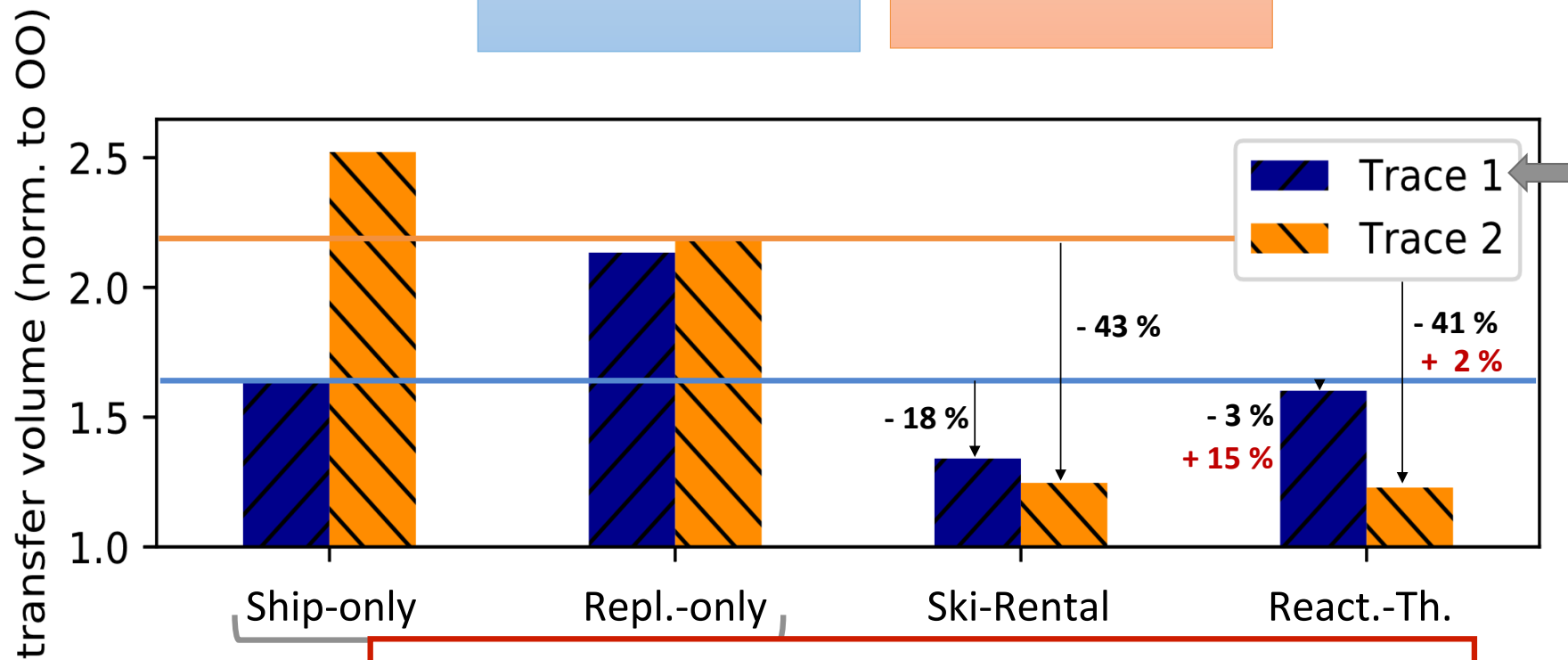
Skewed data access across the two traces.

Evaluation

Moderate improvement for Trace 1

Strong improvement for Trace 2

Lower is better!



Why does the dynamic strategy perform worse?

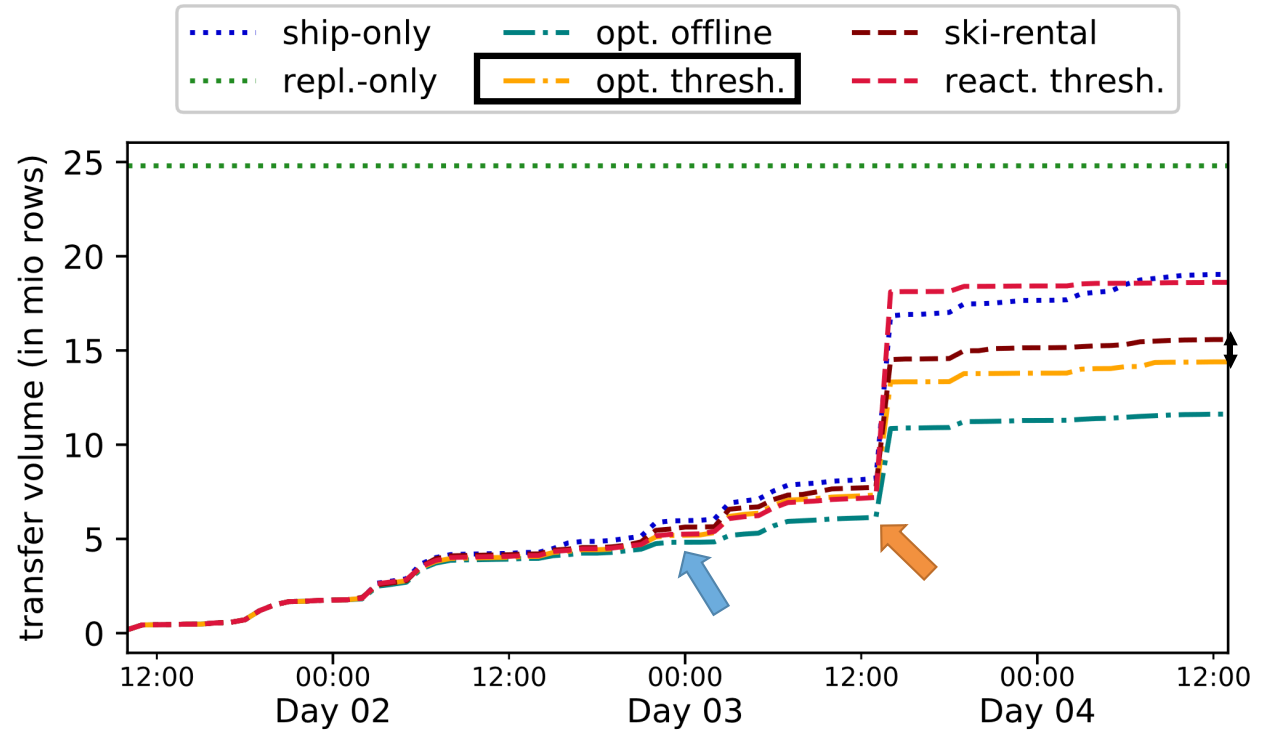
Baselines

Evaluation over time

Lower is better!

Up to half-way strategies perform equally well!

45% of all transfers occur at single time window!
Current Work



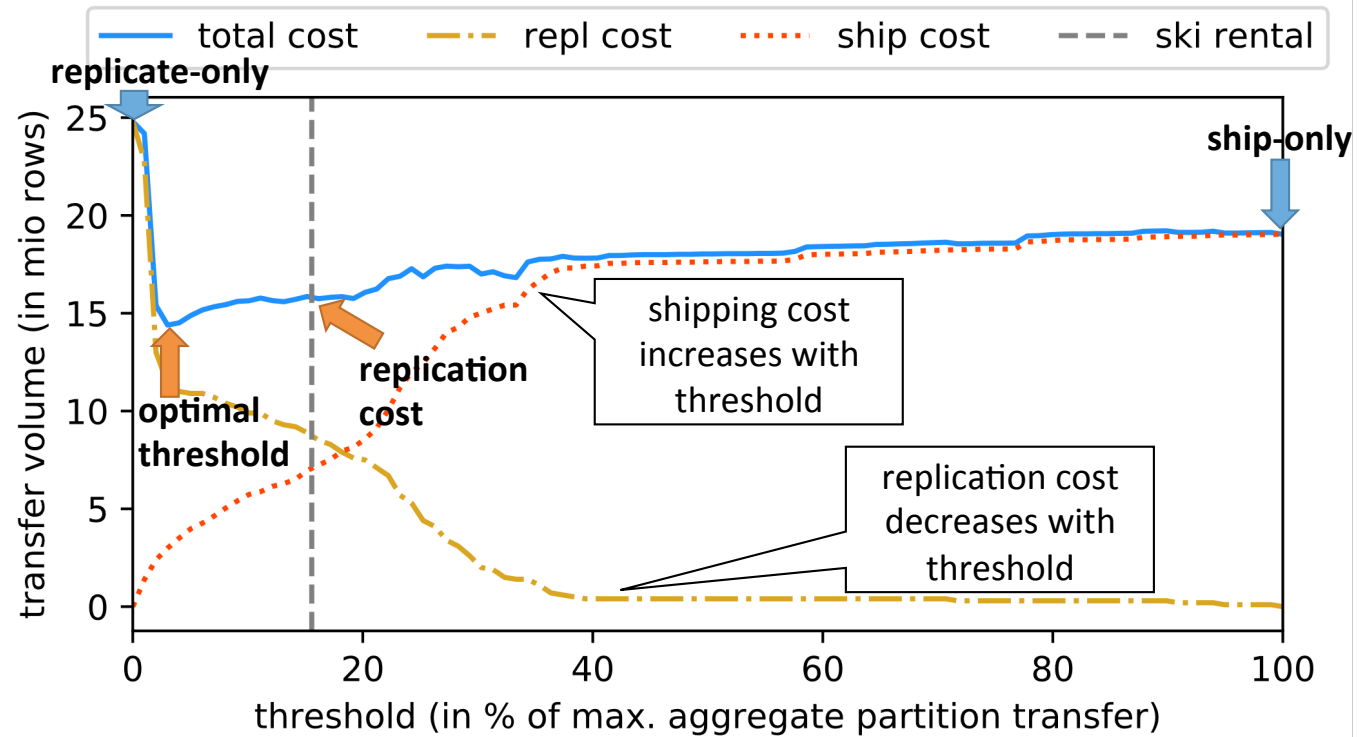
Trace 1

Evaluation of varying thresholds

↓ Lower is better!

Baselines at extremes

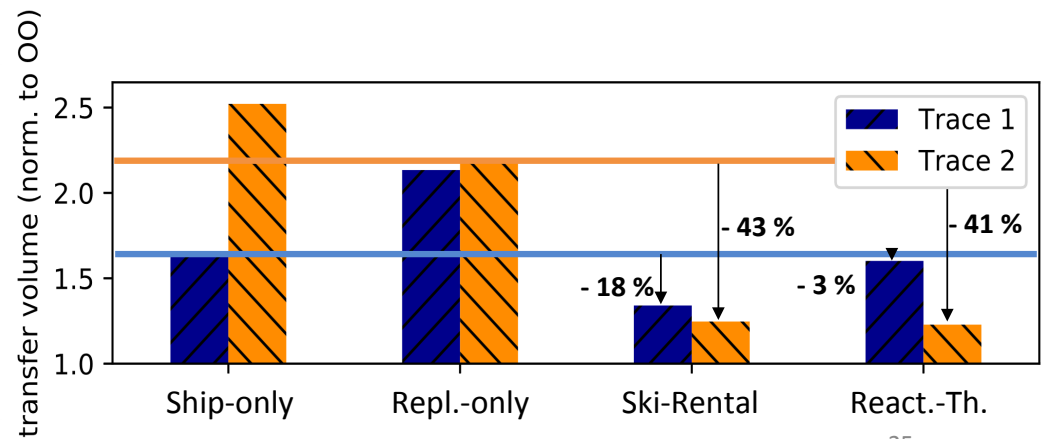
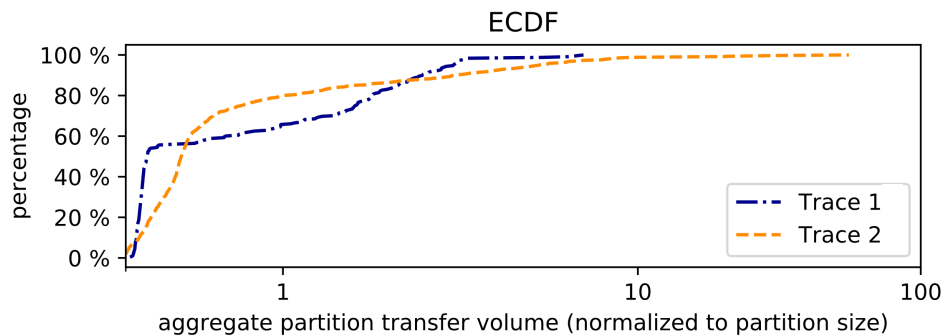
Ski-Rental strategy performs close to optimum.



Trace 1

Evaluation of the effect of skew

- Trace 2 is clearly more skewed than Trace 1.
- Both threshold strategies are more successful for Trace 1.
- We expect a real IoT trace to be even more skewed!



Summary of results

- Improvement of 18% to 43%
- Performance of Ski-rental strategy is only 1% and 11% away from optimal threshold.
- Reactive-threshold strategy outperforms baselines
- Future work: Make reactive-threshold strategy more robust

Conclusion

- We use data replication to reduce transfer volume.
- We introduce two online replication strategies.
- Our static strategy shows a reduction of transfer cost between 18% to 43%.

Thank you!

niklas.semmler@sap.com

