Spatial-Aware Approximate Big Data Stream Processing

Luca Foschini ,

Isam Mashhour Al Jawarneh, Paolo Bellavista, Rebecca Montanari

Department of Computer Science and Engineering - DISI
University of Bologna, Italy

{luca.Foschini, isam.aljawarneh3, paolo.bellavista, antonio.corradi,}@unibo.it,

IEEE GLOBECOM 2019
9th - 13th December

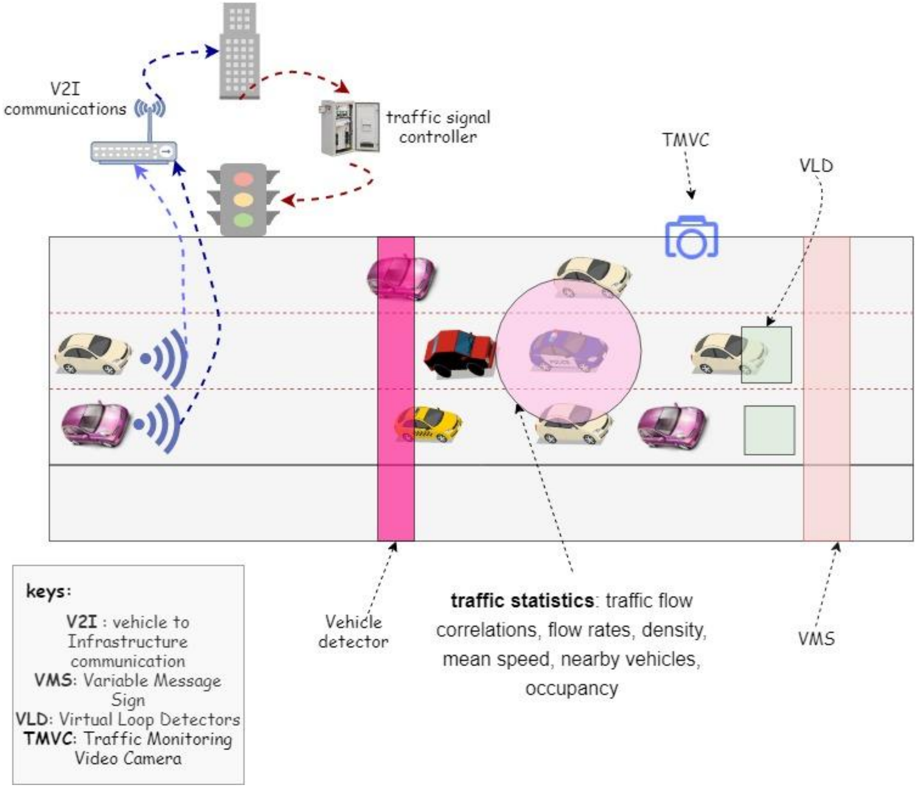
Agenda

- *Spatial Approximate Computing*: Background and Motivations
- *SpatialSPE*
  - SAOS spatial online sampling
  - Supported online queries
- *SpatialSPE* Deployment
  - Baseline system
  - Experimental setup
- Experimental Results
  - Extensive Microsoft Azure Spark cluster Test
- Conclusion

Urban planning scenario: predict the future of cities

real-time traffic control system

- Municipalities seek to cut costs of installation, repair and maintenance of detectors at junctions of streets and along freeways.
- Which are the best locations to install detectors, VMS, TMVC?
- Vehicles pass only once through the detectors; traffic statistics should be computed very fast.
- What if big number of vehicles pass through monitoring points!
- Spatial Approximate Query Processing (SAQP) is the key.



Exploiting geospatial big data for the resource management of telecommunication infrastructure

Spatial Online Sampling is expensive!

- Assigning coordinate pairs to taxi zones is one example of a ***spatial join.***
- There is the "***curse of multidimensionality***“.
- The Taxi and Limousine Commission (TLC) only provides ***pick-up and drop-off locations*** of taxi trips as longitude and latitude points. Also, “***taxi zones***” in NYC (polygons).

QoS Tension

Spatial (Approximate) Query Processing (S(A)QP)

Tension

latency

Throughput

Accuracy

Spatial Approximate Query Processing (SAQP)

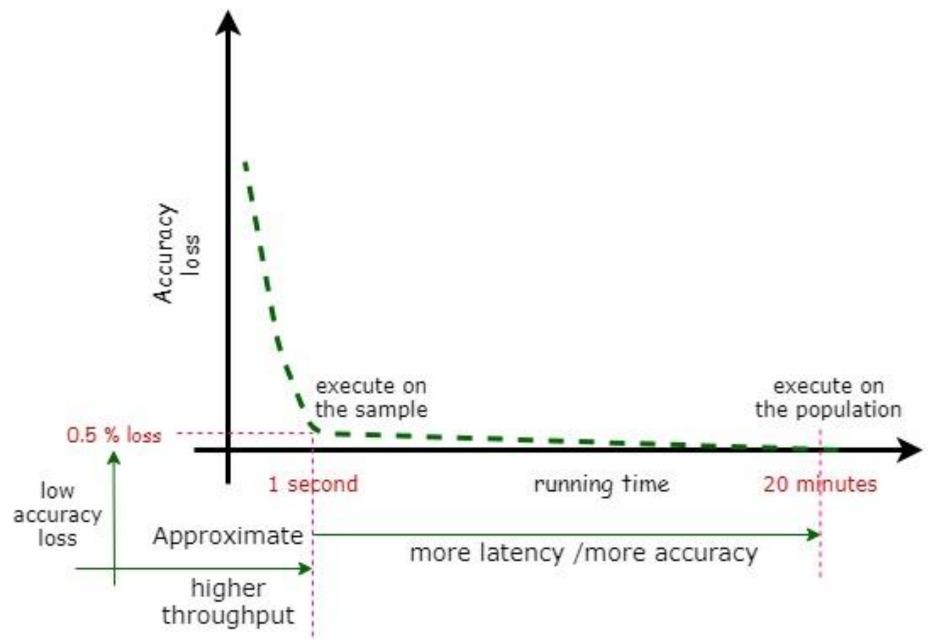
- Stream Processing Engines (SPEs) are confronted with complex challenges:

✓ fast arriving streaming workloads.

✓ Temporal arrival rate fluctuation and skewness.

- Can we do better?

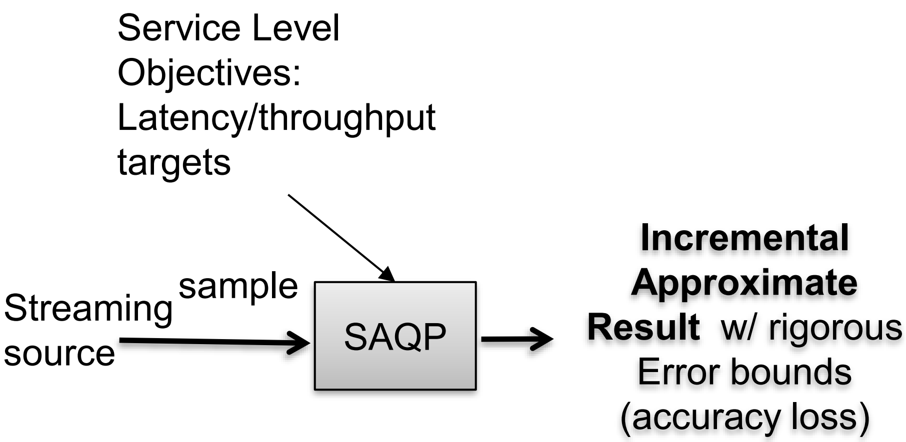
✓ After 1 second, we obtain a 99.95 accurate early result, which is satisfactory for decision making, which then makes the final exact result not needed.



Spatial Approximate Query Processing (SAQP)

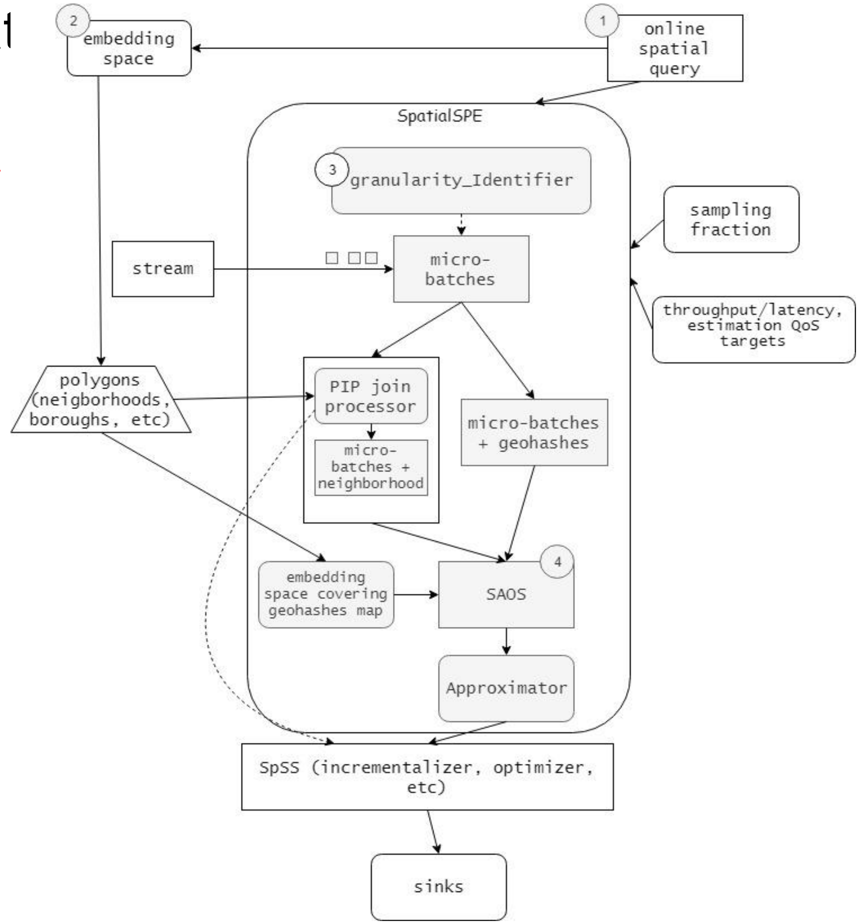
- ***Spatial Approximate Query Processing (SAQP)*** has emerged to solve part of the tension between low-latency and high-accuracy trade-offs.
- ***Sampling***. Observing a portion of the population to calculate an attribute: mean, median, range, variance.
  - Users are satisfied with approximations and are willing to trade an ***error-bounded accuracy*** for even a small ***latency gain***.
  - In streaming contexts, we do not have access to such thing like a total population.

Computing over a sample instead of the whole population



SpatialSPE

- Spatial data maintain spatial trends that affect the observed responses
  - ***spatially representative samples*** → selecting spatially well-spread out samples positively affects the accuracy of estimators (average, median, etc.).
- ***Example Continuous Query (CQ).*** “measuring the average trip distance travelled by taxis from each borough in NYC, United States”
- Sampling fractions are the same for all constituent stratum.
- CQ is ***incrementalized.***



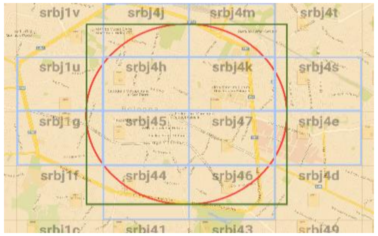
SpatialSPE overview

Spatial Aware Online Sampling (SAOS)

- A ***hybridization*** between ***z-order curves*** (geohash) and ***simple probability sampling*** (within each grid cell).
- does not require a pre-knowledge of the streaming statistics, it otherwise depends on ***incrementalization***.

Algorithm 2: Spatial-Aware Online Sampling (SAOS)

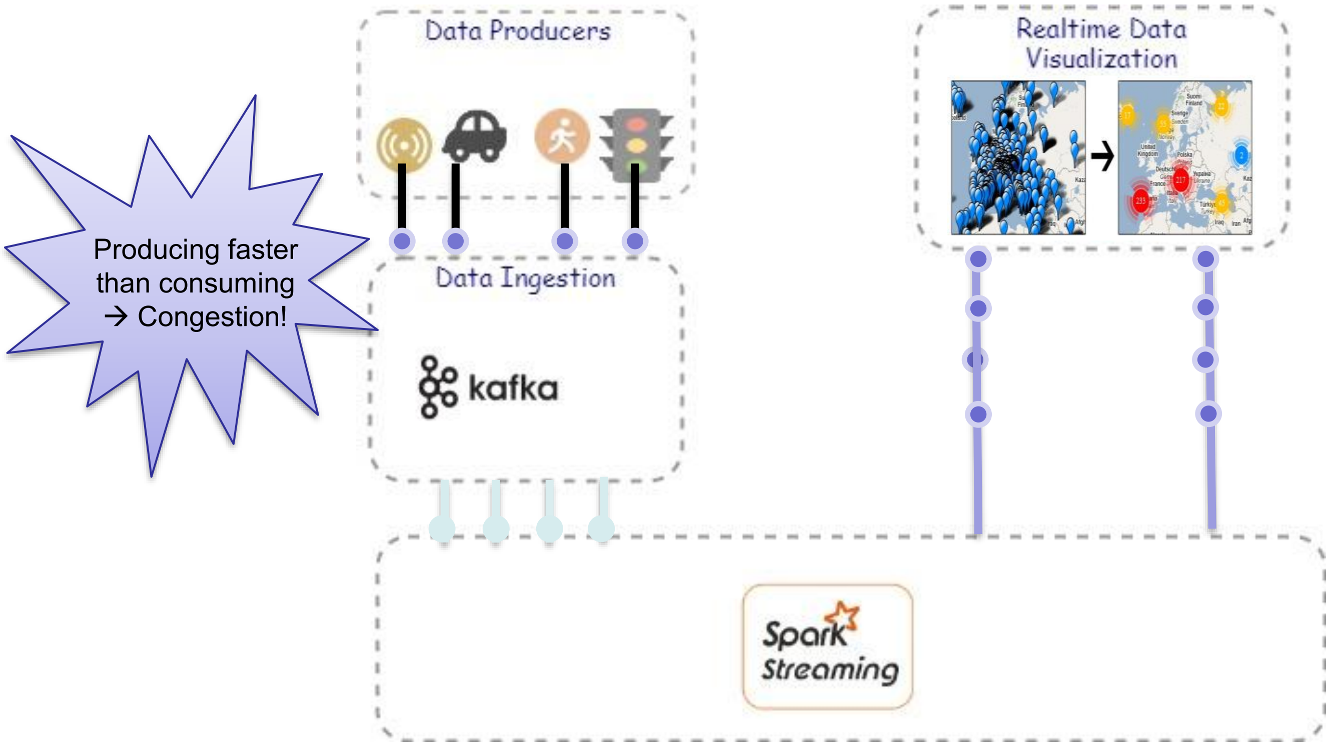
SAOS (micro-batch-tuples, samplingMap, samplingFraction, seed)
r = random(seed)
S ← ∅
Foreach tuple **in** micro-batch-tuples **do**
  geohash ← geocode (tuple)
  //get the sampling fraction for this geohash key = $fraction_i$*, or zero if not present.*
  $fraction_i$ ← samplingMap.getOrElse(geohash,0.0)
  //toss a coin for selecting items belonging to each geohash from the current batch interval
  If (P (r < $fraction_i$))
  S.put(tuple)
  End
End



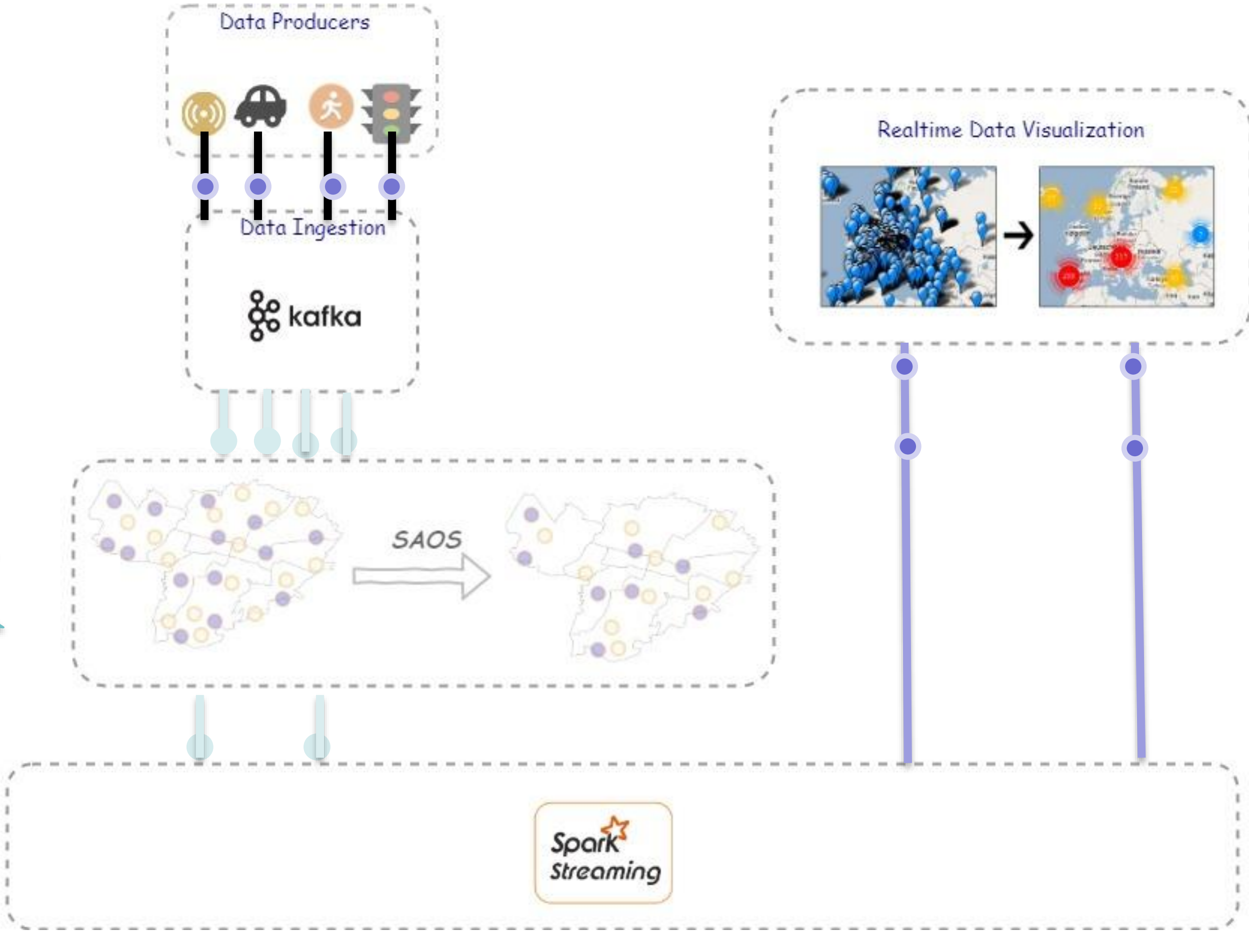
heuristic overview

- **Geohash** indexing. An ordering (string representation) imposed on grid surface earth planar representation.
- Nearby points share the same geohash prefixes, thus reducing the two-dimensional point representations to one-dimensional string ordering.

Typical pipeline architecture w/o SAOS



The improved architecture w/ SAOS



SQL-alike continuous geospatial query

Same serving speed, but serving less data!

Supported Queries

Single queries

"finding an estimate of the population average for a target variable by only using a spatial sample"

$$\bar{Y}_{SAOS} = \frac{\hat{t}_{SAOS}}{\mathrm{N}} = \sum_{i=1}^{I} \frac{N_i}{N} \bar{y}_i$$

$$RE = \frac{z_{\alpha/2} SE(\bar{Y}_{SAOS})}{\bar{Y}_{SAOS}}$$

Continuous query

Incremental estimation

Calculating error bounds (e.g., relative error RE) and spearman's rho

Stateful aggregation

"find the average trip distance travelled by taxis originating from a specific district in a metropolitan city"

spatial incremental
Top-N online aggregation function

$$\rho_{rg} = \frac{cov(rank_{nos}, rank_{samp})}{\sigma_{rank_{nos}} \sigma_{rank_{samp}}}$$

Baselines

Spark Streaming

SRS, Stratified Sampling

Spatial-awareness is not supported

Spark Structures Streaming

Sampling is not supported

In spatial patchy distributions, where spatial points are clumped into few patches, selecting a sample depending on Simple Random Sampling SRS potentially results in inaccurate results is it may tend to select disproportional quantities from each patch (area).

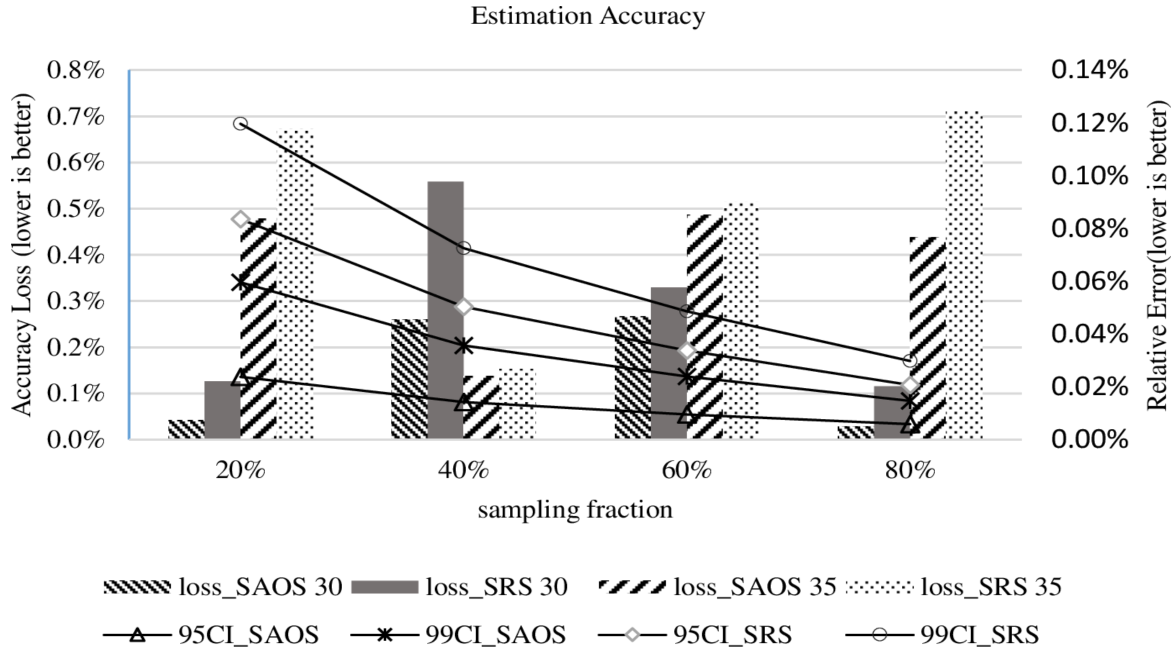
Baseline System

- We have implemented SRS on Spark Structured Streaming (we term as SpSS-based SRS baseline) and compared our new design (SAOS) with that baseline.
- SRS normally unduly overlook regions, resulting in maps that do not necessarily represent the real distributions, which does not help in assisting a correct decision making.

Experimental setup

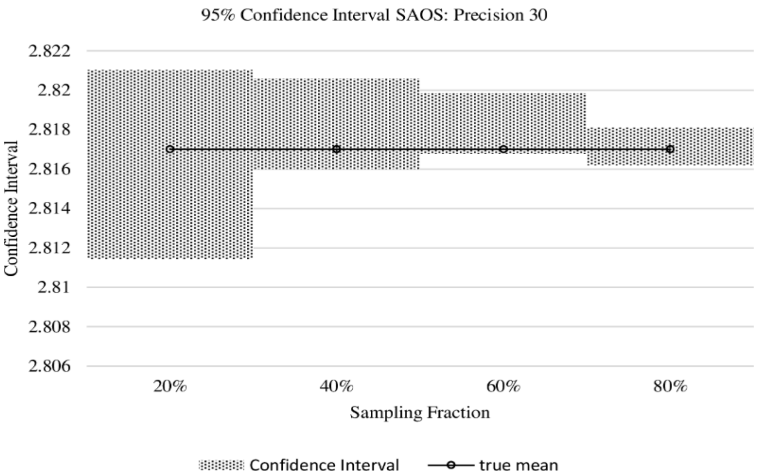
- Evaluation questions
  - Throughput vs Sampling fraction
  - Sampling fraction vs accuracy (and confidence interval)
  - Sampling fraction vs rho
- Testbed
  - Cluster: 6 nodes (Microsoft Azure HDInsight Cluster)
  - Datasets:
    - NY City taxicab trips datasets (cohort of six months dataset (around nine million units)

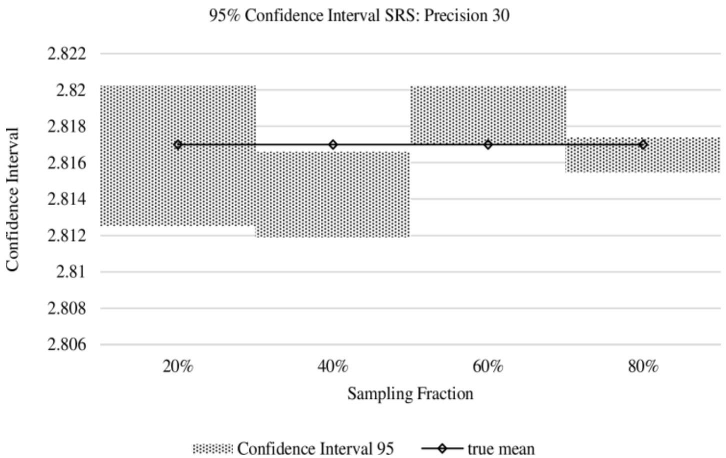
Sampling fraction vs Accuracy



- We define the accuracy loss as accLoss = |estimatedMean – trueMean| / trueMean.
- SAOS outperforms SpSS-based SRS for all precision settings (30 and 35), for both measures, accuracy loss and relative error.
- SAOS have bigger accuracy loss for geohash precision 35 , compared to SAOS accuracy loss at geohash precision 30.

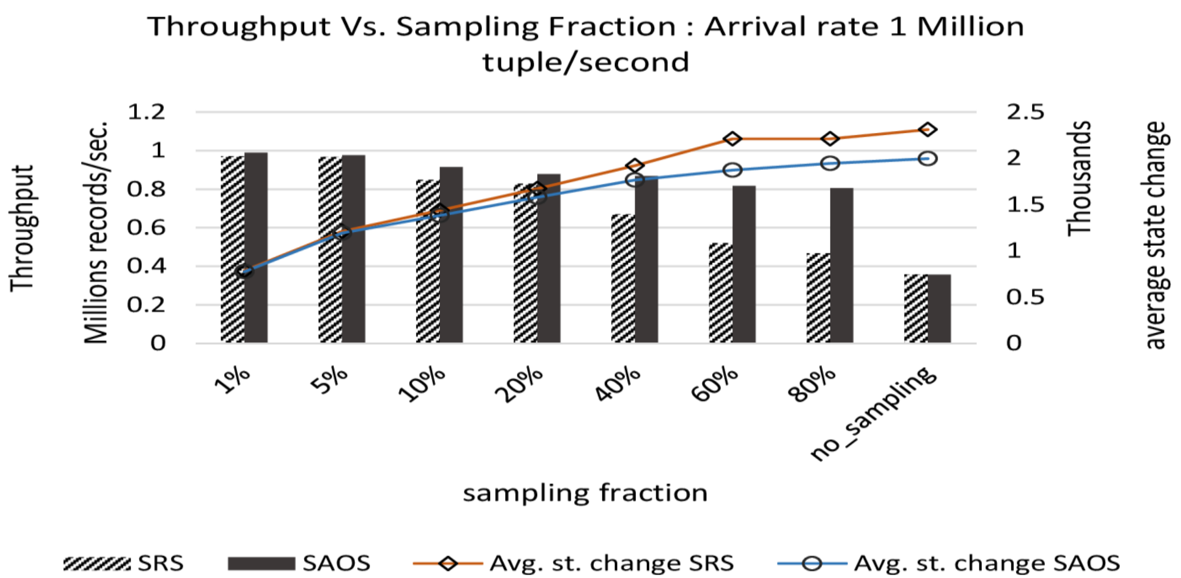
Sampling fraction vs Accuracy (Confidence Interval)





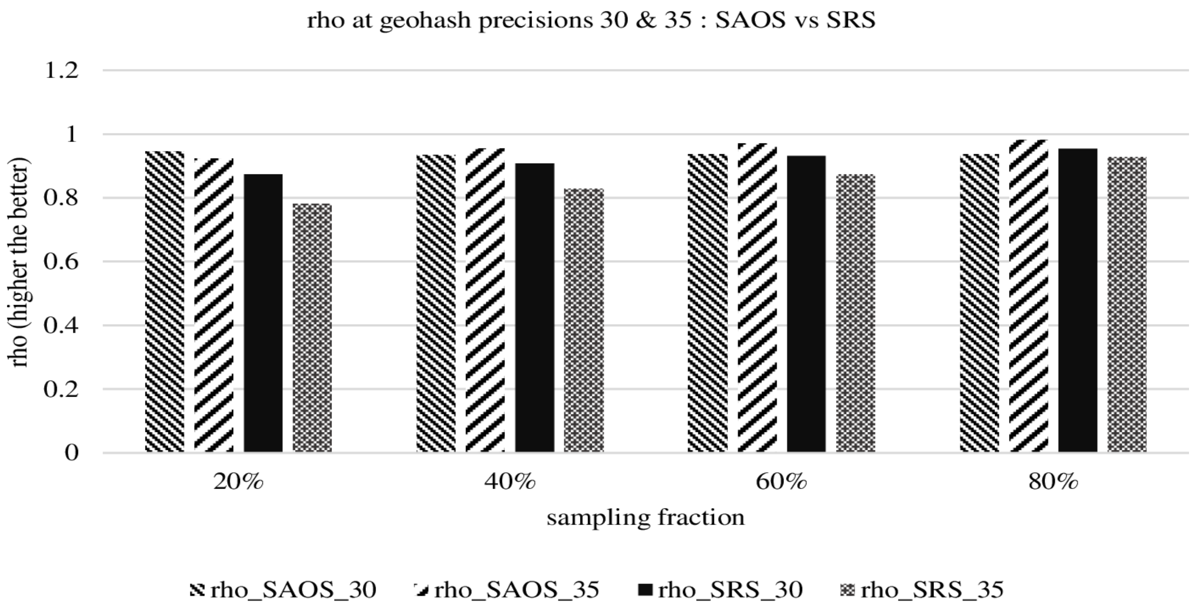
- under SAOS , for 95% of the possible samples of all fractions , the corresponding confidence intervals cover the true value of the population mean (a.k.a. average).
- SRS confidence intervals are susceptible to missing the true value.

Throughput vs Sampling fraction



- We define the throughput as the count of streaming tuples that can be processed with specific computation resources during a period (window interval in sliding window semantics).
- SAOS outperforms SpSS-based SRS.

Spearman’s rho vs Sampling fraction



- Spearman's rho is a measure for statistical dependency between the ranking of two variables in a dataset.
- Ranking precision of SAOS outperforms those for SRS.

Concluding remarks

- Most interesting analytics are required during data streams brutal spikes in arrival rates!
- We have designed an **end-to-end** QoS-aware framework for processing data coming from dynamic and scalable applications scenarios.
- Our architecture extends the trending Lambda architecture by providing QoS-awareness, Spatial support at the speed layer, thus supporting mixed spatial workloads.

Q&A and Contacts

Thanks for your attention!

Questions time...

Isam Al Jawarneh
Email: isam.aljawarneh3@unibo.it

Luca Foschini
Email: luca.foschini@unibo.it

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA
DIPARTIMENTO DI INFORMATICA - SCIENZA E INGEGNERIA