Designing Distributed Geospatial Data-Intensive Applications

Ph.D. Course, **2022**

Instructors:

Prof. **Luca Foschini**, Associate Professor &

Dr. **Isam Mashhour Al Jawarneh**, Postdoctoral Research Fellow

{isam.aljawarneh3, Luca.foschini}@unibo.it

Department of Computer Science and Engineering (DISI), Università di Bologna

Part 3

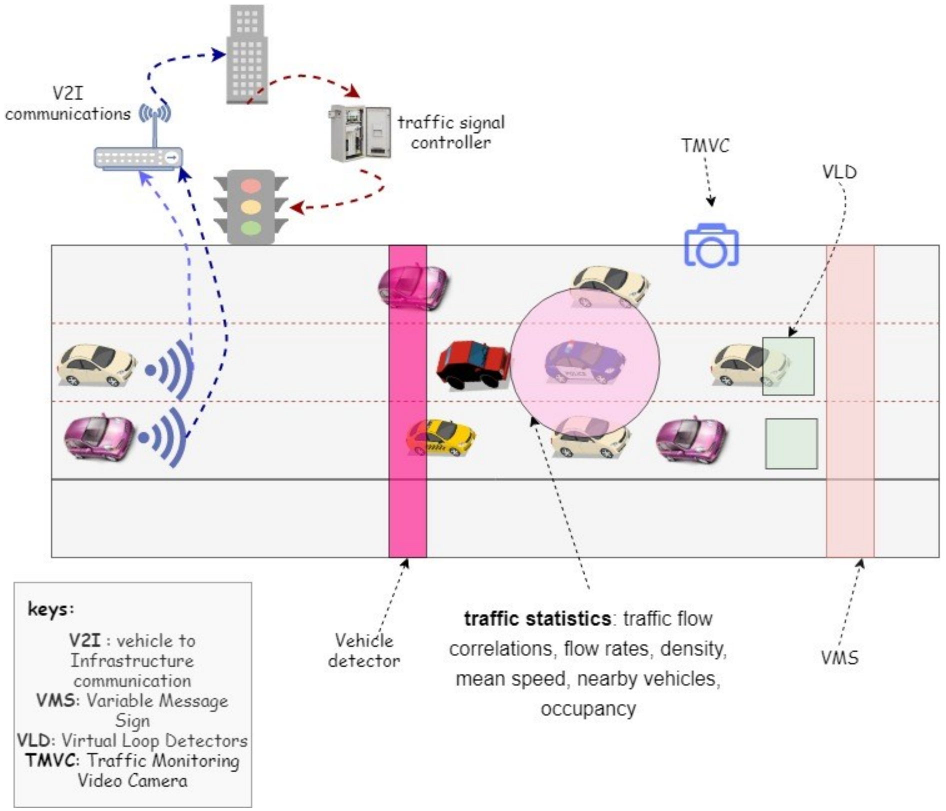
Designing QoS-aware **approximate** solutions for distributed geo-spatial data-intensive applications

27th July 2022

Urban planning scenario: short-term predictions for smart resource management

real-time traffic control system

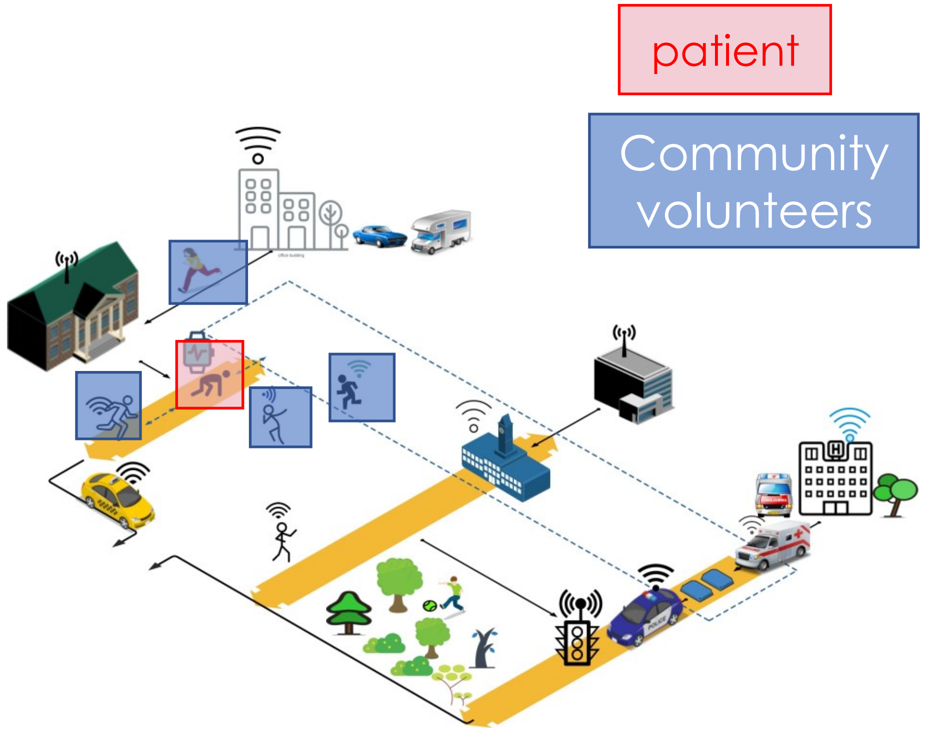
- **Problem:**
  - Municipalities need to ***install*** a set of new monitoring stations. Such as traffic cameras and sensors to study traffic trends in a metropolitan city.
  - They seek to cut costs of installation, repair and maintenance of detectors at junctions of streets and along freeways.
  - Equipping all traffic points with such tools would be expensive.
- **Goal:**
  - To choose representative locations, that are well spread out.
  - Which are the best locations to install detectors, VMS, TMVC?
  - Need to study the trend, ***but*** vehicles pass only once through the detectors; traffic statistics should be computed very fast.
  - Computing statistics for all arriving GPS signal could turn prohibitive during rush hours!
- **Solution:**
  - Spatial Approximate Query Processing **(SAQP)** is the key.
  - **Sampling** and choosing portions of GPS signals from every potential location.



Exploiting geospatial big data for the resource management of telecommunication infrastructure

Motivating Application Scenario

participatory healthcare



A mixed-workload scenario requiring at least:

- **Traffic Light Controller**. Actuator decides to change lights consistently for ambulance to pass
- **Smart Real-time Pathfinder**. Interactive navigation map for ambulances and other vehicles
- **Real-time Community Detector**. Identify volunteers' communities in the surroundings of the patient

➢ **Primitive geospatial queries (expensive!)**
  - Proximity queries
  - Spatial join
  - Spatial clustering
  - Spatial geo-statistics.
  - *k*-Nearest Neighborhoods)

➢ Data arrives fast during peak hours

➢ Exceeds the capacity of ingestion and processing systems

- Spatial Approximate Query Processing (**SAQP**) is the key.

Original work source

Sampling

- the procedure of selecting a **representative portion** (could be **miniatures**) of a **population** for **estimating** an unknown population **quantity**, such as an '***average***' or '***count***' of a **target variable**
- **Population** represents all units in a specific **study area**
  - all persons in a city, where the **target** of sampling is, for instance, estimating the **average age** of persons
  - Those estimators are normally associated with a **variance** measuring their **accuracy**
- Sampling is pivotal for most **statistical** studies for various reasons

  (1) obtaining a **total population** could be purely **fictional**
    - For instance, heights of all people in a country

  (2) **processing** a whole **population** census is **computationally challenging**
    - data arrives in **streams,** where updating results regularly based on newcomers is pivotal for correct time-dependent **estimators**
    - we usually base our estimates on **observations** arrived **so-far** and **extrapolate** our results to future times

  (3) it's **not** even **practical** to **visually plot** a summary of **billions** of **observations** on **boards**, such as those cases where we generate **heat-maps** of a **natural phenomenon**

Sampling (cont.)

- A method is a **good** or **bad** sampling method depends on various **factors** including the **sampling design** and size
  - The **sampling design** is the procedure by which a **sample** of units or sites is **selected**
- the sample should be a good **representative** for the **population**
  - **sample** constitutes a **scaled-down** ('**microcosm**') of a population mirroring characteristics of the **population** it is representing
  - no "**perfectly-representative** sample", at least a sample good enough to yield **characteristic's estimations** with a known degree of **accuracy** or **confidence**,
    - then the sample is **representative**
- some sampling designs are bad because if the **selection biasedness**
  - sampling method **overlooks** some **parts** of the **population** by design
  - E.g., estimating a percentage of possible voters in the United States who potentially will vote for the democratic party in an upcoming election cycle,
    - selection biasedness may render estimates invalid
- sampling causes sampling **errors** (**Standard Errors** (SE))
  - basing estimates on a sample rather than the population

Sampling (cont.)

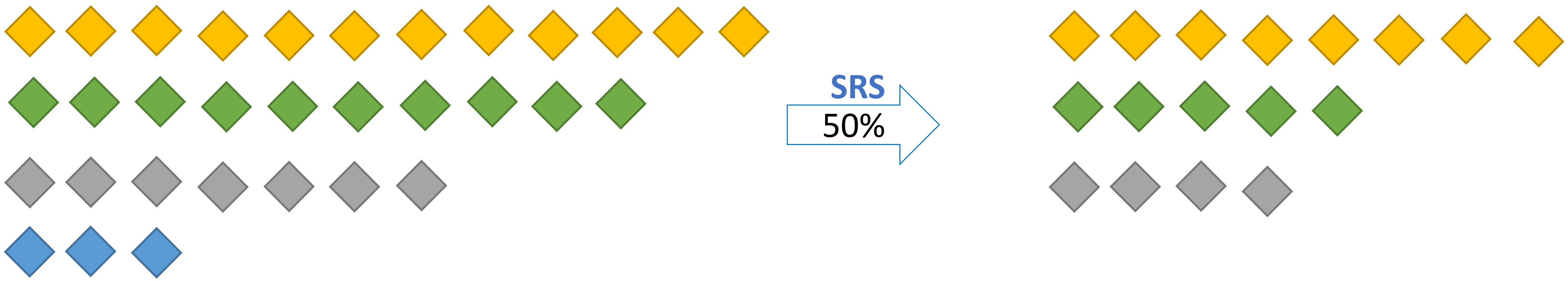
- **Modeling uncertainty** has strong ties with selecting proper **sampling designs**
  - A design that minimizes **uncertainty** (e.g., standard errors) is plausible
  - values estimated using a **sample** are close to the **real values** (i.e., estimated from the population with no sampling) for some arbitrary number of sampling permutations, the method is considered **good**, otherwise not
- two most widely used
  - simple random sampling (**SRS**) , which is a probability design (a.k.a. random sampling without replacement)
  - and Simple Stratified Sampling (SSS).
- **SRS**
  - assigning an equal **selection probability** to each **unit** in the **population**,
  - thereafter, assigning **labels** to each **unit** and **selecting labels randomly** until a **specific number** of **distinct units** that is equal to the **sample size** is selected
  - all possible **permutations** have equal **probabilities** of being considered as a sample

Sampling (cont.)

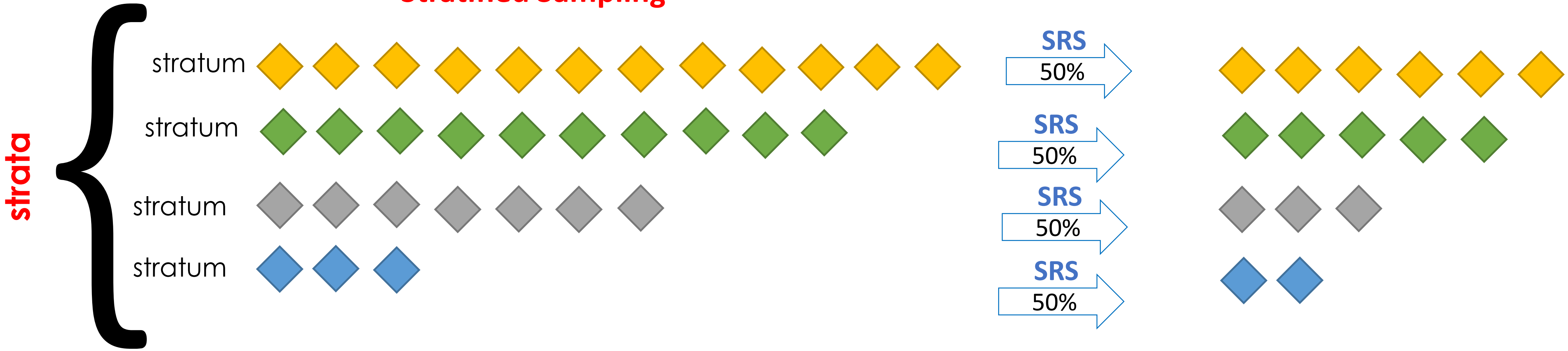
- SSS
  - selects fractional **portions** from **population units** depending on the **group** they belong to
  - Sampling students from schools, we take **50%** boys and **50%** girls, where boys and girls are **stratum** in this case.
- The distinction
  - SSS may assign **equal inclusion probabilities** to each **unit** in the same **stratum**, but this may **differ** from other **units** in **other stratum** as **each stratum is treated independently**

Sampling methods

Random Sampling



Stratified Sampling



QoS Tension

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

latency

Tension

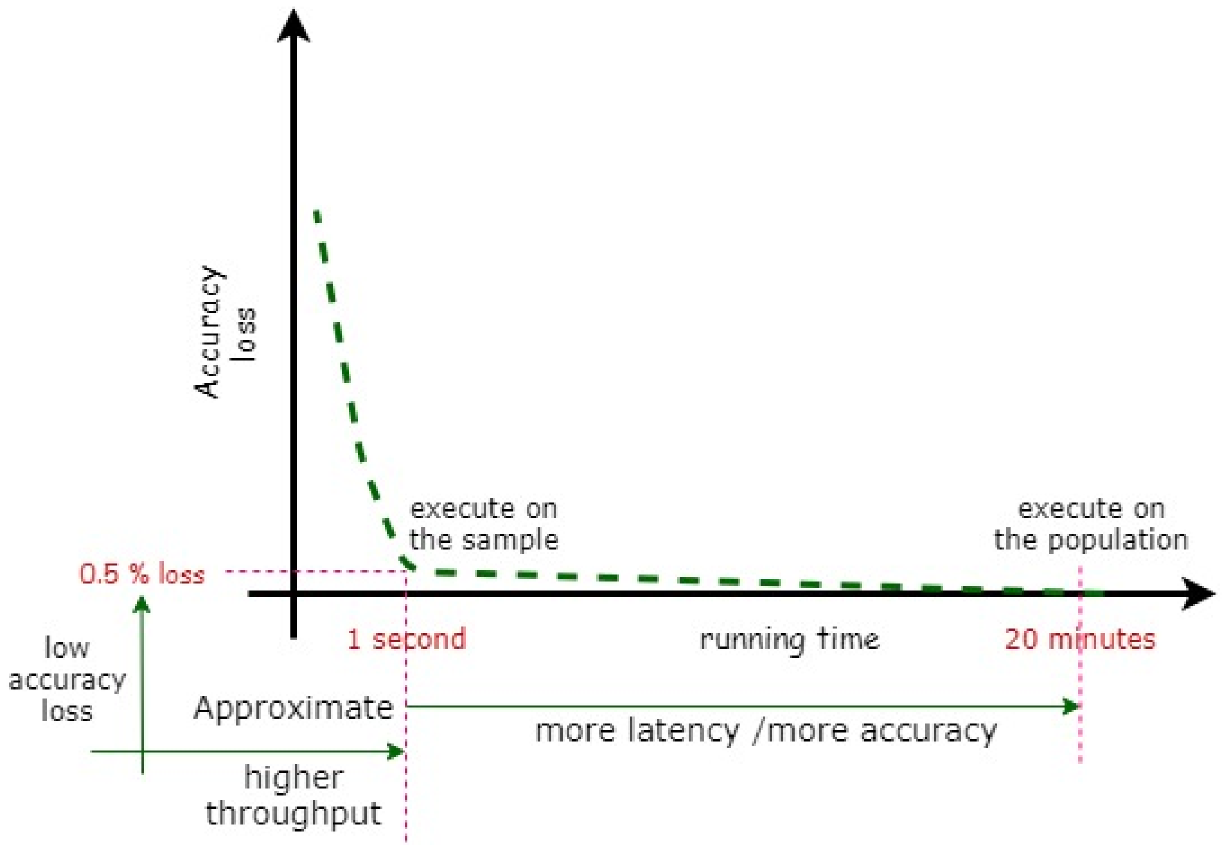
Throughput

Accuracy

Original work source

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.



[Original work source]

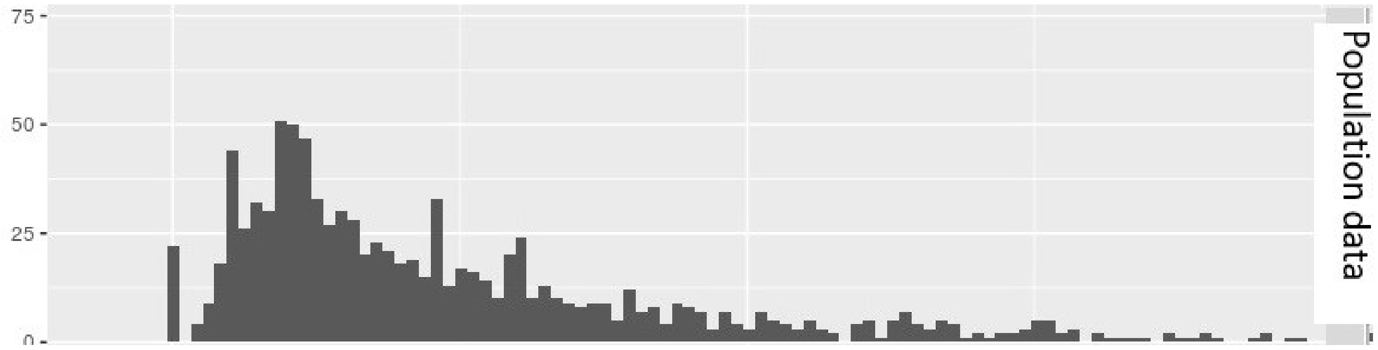
Introduction to Spatial sampling

Spatial Online Sampling

- formally expressed with a **ternary** $(\psi, \mathfrak{I}, \mathfrak{R})$,
  - $\mathfrak{R}$ is the **embedding space** (often two- or three-dimensional space) from which samples are drawn,
  - $\mathfrak{I}$ is the **sampling frame** (i.e., SRS, SSS) **overlaying** the **survey area** (i.e., **embedding** space),
  - ψ is the **statistic** for estimating a **variable of interest** (e.g., '***total***' and '***mean***' of a **parameter** in study area)
- The **choices** of $\mathfrak{I}$ and ψ heavily **affects** the **goodness** of the **spatial sampling design**
- Those configurations enforce an **uncertainty** on the **spatial sample estimation** and the common goal is to reach an **unbiased estimation** with the **lowest** possible **variance**,
  - in spatial distribution, is normally achieved by being **attuned to the characteristics of the spatial data**, where the sample is **spatially representative** and **well-spread out** over the **sampling space**

Spatial Online Sampling challenges

- **Deterministic** solutions for data analytics problems do not play well with **fast arriving huge data streams** that are mostly **geo-referenced** with complex **data structures** that show **oscillation** in **data arrival rates** and **skewness**
- in **geo-statistics**, **approximations** that yield plausible **error-bounded statistical** results are acceptable
  - **well-selected representative** sample can be safely exploited for **geostatistical** analytics such as the approximation of target study variables (e.g., '***average***', '***total***' and '***proportion***')
- **observing** all items of a **population** could be **intractable**, such as observing **migrating birds** in a huge location, which are **spatially unevenly distributed**



Mobility data. NYC taxicab dataset is highly skewed

Spatial Online Sampling challenges (cont.)

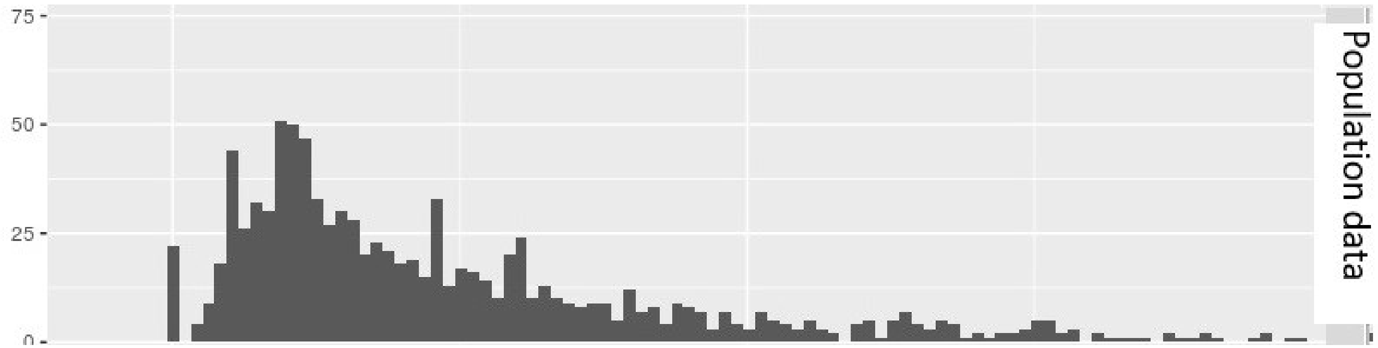
- **Preserving** spatial **co-locality** through a sampling design is known to yield better estimates
  - A principle that complies with **Tobler's first law of geography** → **nearby** spatial **objects** are **more related** than those **far apart**
  - imagine the **earth flattened out** (i.e., two-dimensional planar **irregular grid-like** representation) and **sample** proportional quantities from each **subregion** (i.e., **cell** or **polygon**),
    - known to yield **plausible statistical** results with **reduced estimation errors**
- Current Stream Processing Engines (SPEs) with their related **spatial-aware extensions** and plugins focus on striking a **weighted balance** between few **QoS** goals (e.g., **low-latency** and **high-accuracy**)
  - by either **overprovisioning** resources (i.e., **scaling in/out**) or
  - **dropping-off** (a.k.a. **sampling** or **shedding**) portions from the arriving data, thus loosing tiny **accuracy** for plausible **latency** gains.
  - **overprovisioning** resources, that are not normally released after a spike, conflicts with the target of **high resources utilization**
- state-of-art SPEs exploit sampling schemes that are basically embracing randomness, based mostly on SRS
  - rendering them **non-attuned** for **spatial characteristics** that surround objects in **proximate locations**

Spatial Online Sampling challenges (cont.)

- **SRS** does not serve the ***estimation quality*** QoS target in **spatial patchy environment**
  - spatial objects are normally **clumped** into **few patches (skewness)**
  - SRS normally **unduly** chooses **random** counts with **unfair fractions** from all **cells (stratum)** of the **survey area** (analogous to **strata** in stratified sampling)
  - **geo-near** spatial **objects** have strong **ties** with **contexts** of their surroundings (i.e., **ecological**, anthropogony, etc.,)
- selecting **geographically spread-out samples** is known to affect **estimations quality**
  - ***geospatially representative samples***
- works of the related art consider only **static finite** populations
  - as opposed to continuous **infinite** populations that always have superpopulations
- **GOAL**: designing **stratified-like** spatial **sampling** methods that select **well-spread** out proportional **spatial samples** from **irregular regions** in the sampling space (polygons)
  - requirements → constrained to selecting spatial samples in **non-stationary**, **anisotropy online** settings with temporal **fluctuations** in **arrival** **rates** and **skewness**, thus the term stream sampling (a.k.a. online sampling)

Data skewness & partitioning challenge

- Some data in specific domains is highly **skewed**
  - **Skewness** is the asymmetry of a distribution of a variable's value around its mean
- Some keys in the data may have more **frequency** than others
  - Hashing in this case does not help **load balancing** as few keys may dominate the distribution, and will be routed to same partitions, turning them into **hotspots**
  - As this is domain-specific problem
    - In most cases, it can not be automatically mitigated at the **system level**
    - It, otherwise, need to be managed at the **application level**
      - More **logistics** handling

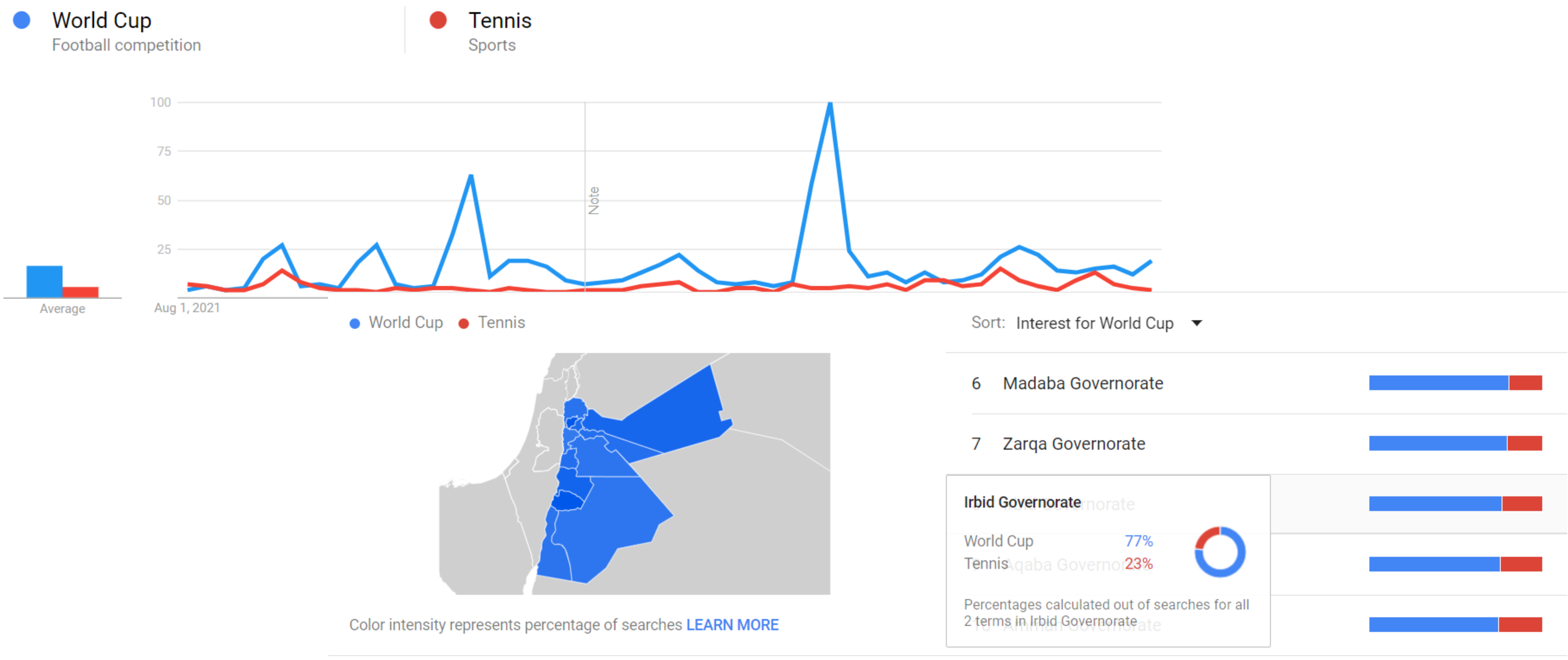


Mobility data. NYC taxicab dataset is highly skewed

Why approximate query processing suffices

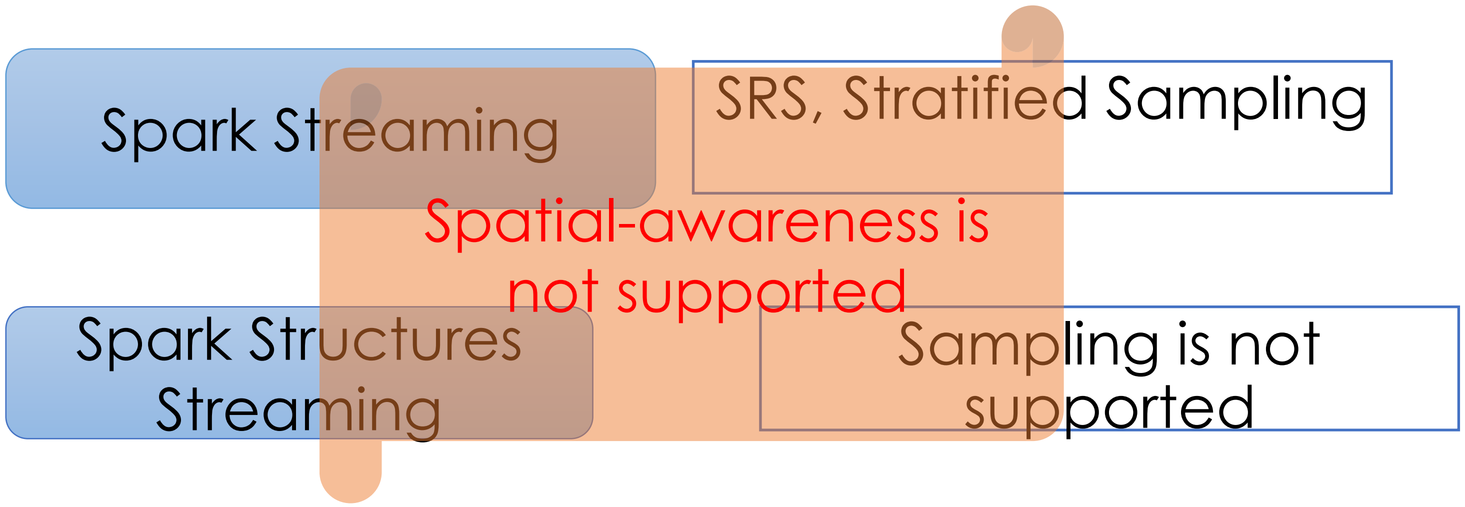
Queries search for trends rather than exact numbers

Example → Google **Trends** ,,,, "**World cup**" against "**Tennis**" per **region** in Jordan (2022)



Spatial approximate query processing in the Cloud

The problem

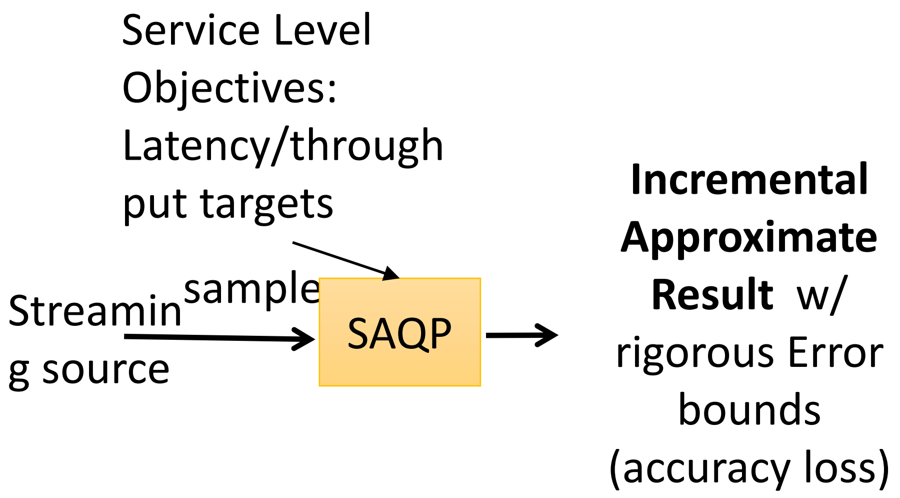


In spatial **patchy distributions**, where **spatial** points are **clumped** into few **patches**, selecting a sample depending on Simple Random Sampling (**SRS**) potentially results in **inaccurate results** is it may tend to select **disproportional** quantities from each **patch** (**area**).

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



Original work source

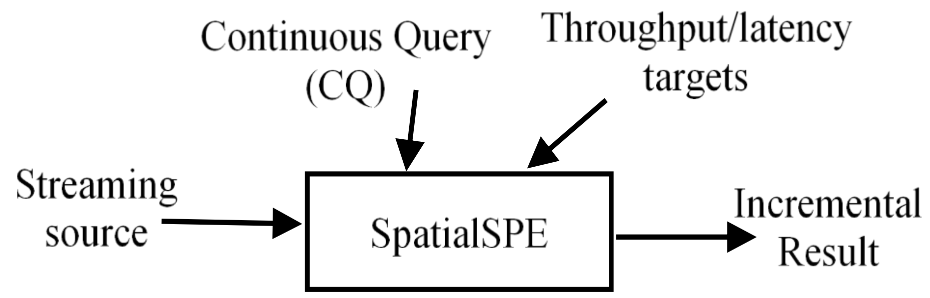
Efficient distributed SAQP system

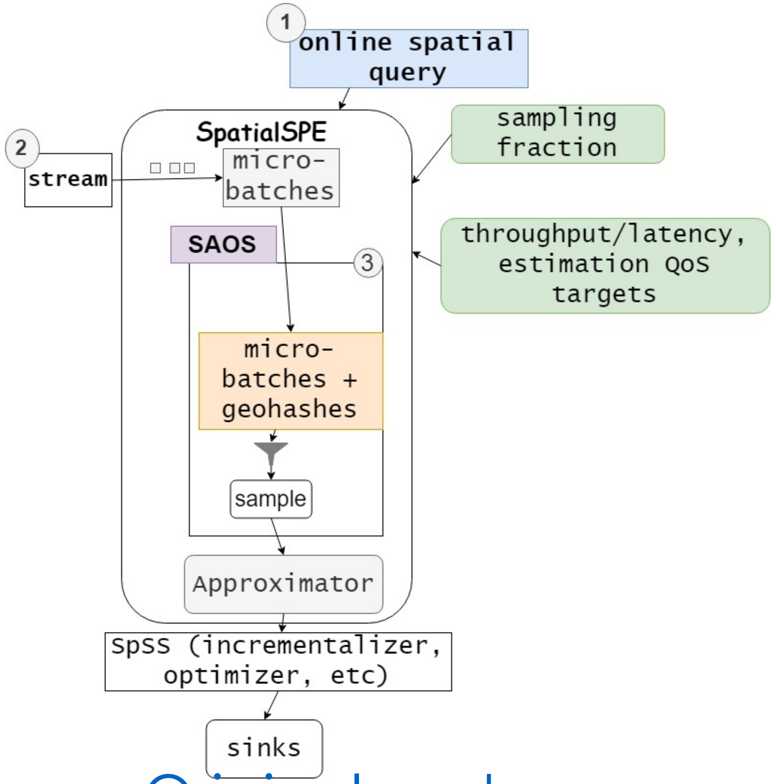
- Spatial data maintain spatial **trends** that affect the observed responses
  - *spatially representative samples* → selecting spatially **well-spread out** samples **positively affects** the **accuracy** of estimators (**average**, **median**, etc.).
- ***Example Continuous Query (CQ).*** "measuring 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.***

QoS requirements

- Balanced **Latency/throughput**
- High **computing resources utilization**
- Higher **accuracy**

```
df = samplepointDF_SSS.groupBy($"geohash").
count().orderBy($"count".desc)
```

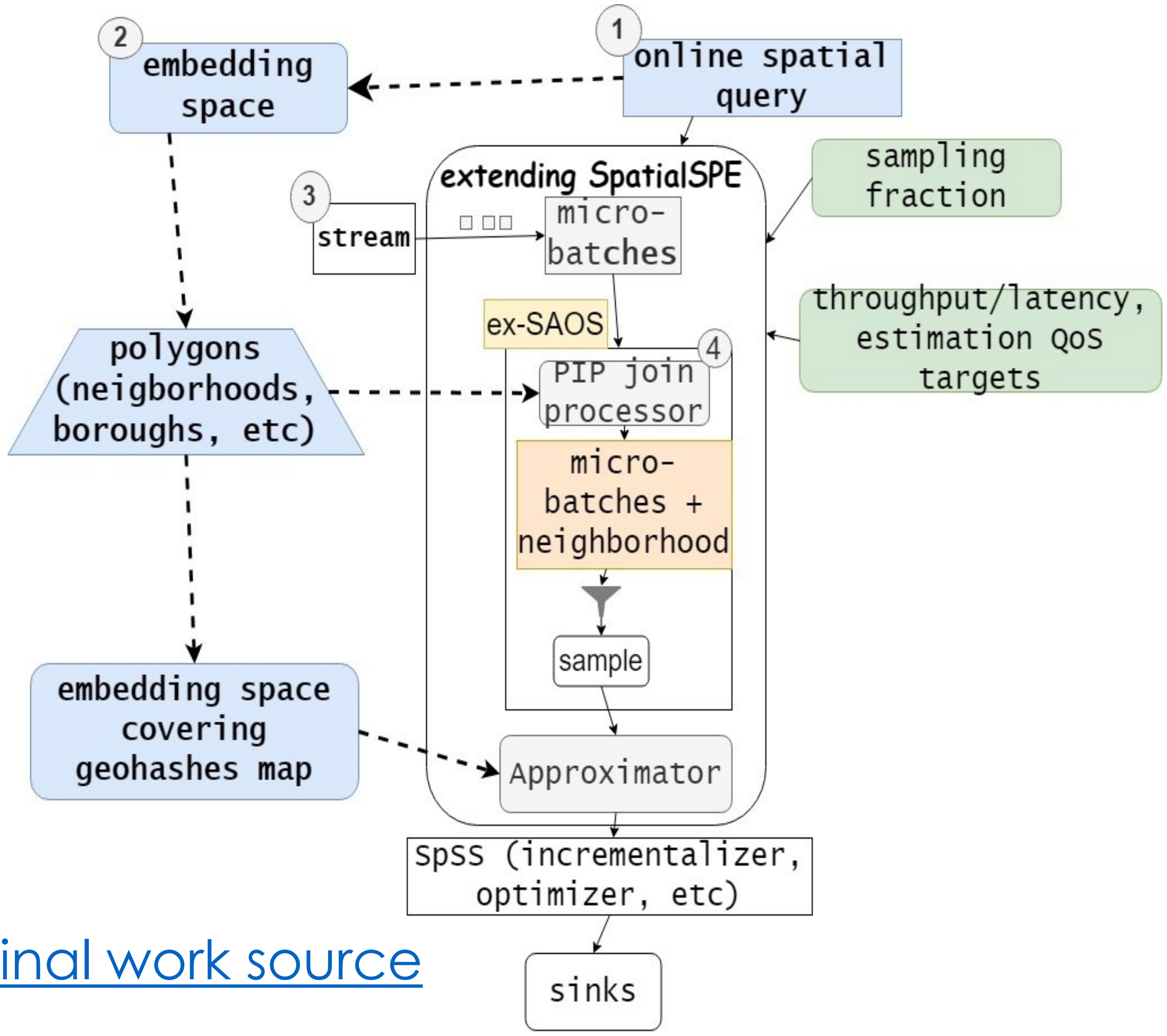




Original work source

Spatial online sampling on a coarser level

- Applying ‘**filter-and-refine**’ to solve the **PIP** test before sampling.
- Discarding ‘**false positives**’.
- We exactly sample **same fractions** from each **neighbourhood** (**borough**, **district**, etc.,)
- Yields more accurate results.

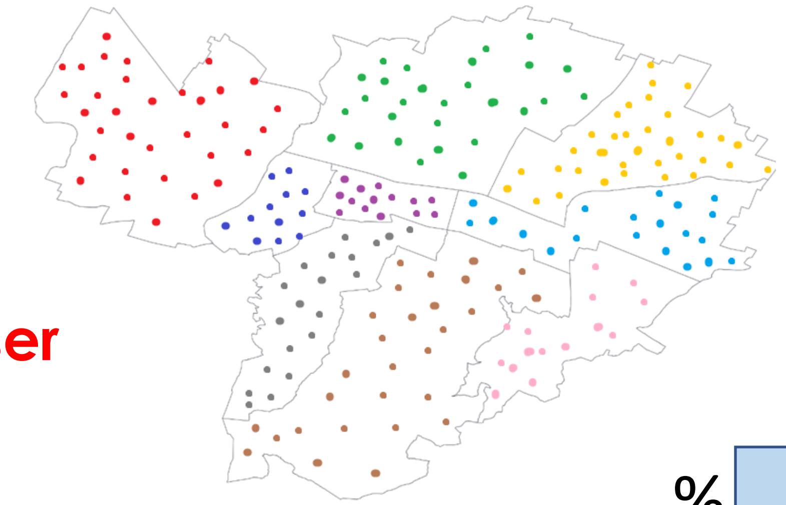


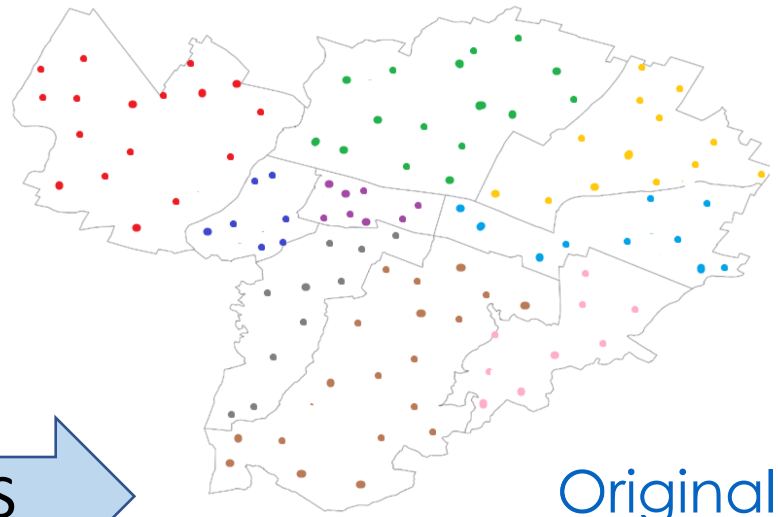
Original work source

Extended version

Spatial Aware Online Sampling (SAOS): overview

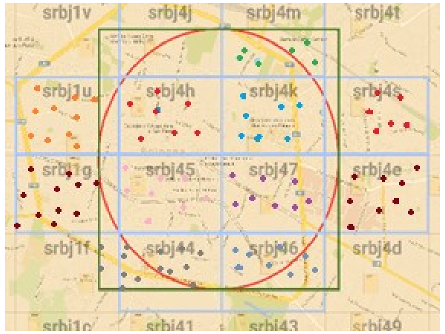
Coarser

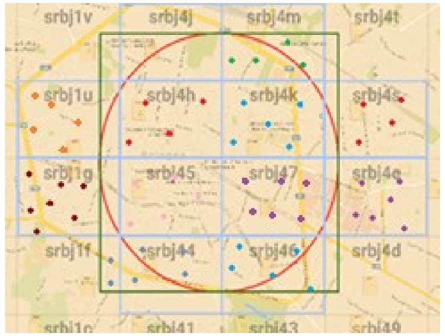




[Original work source](#)

Granular



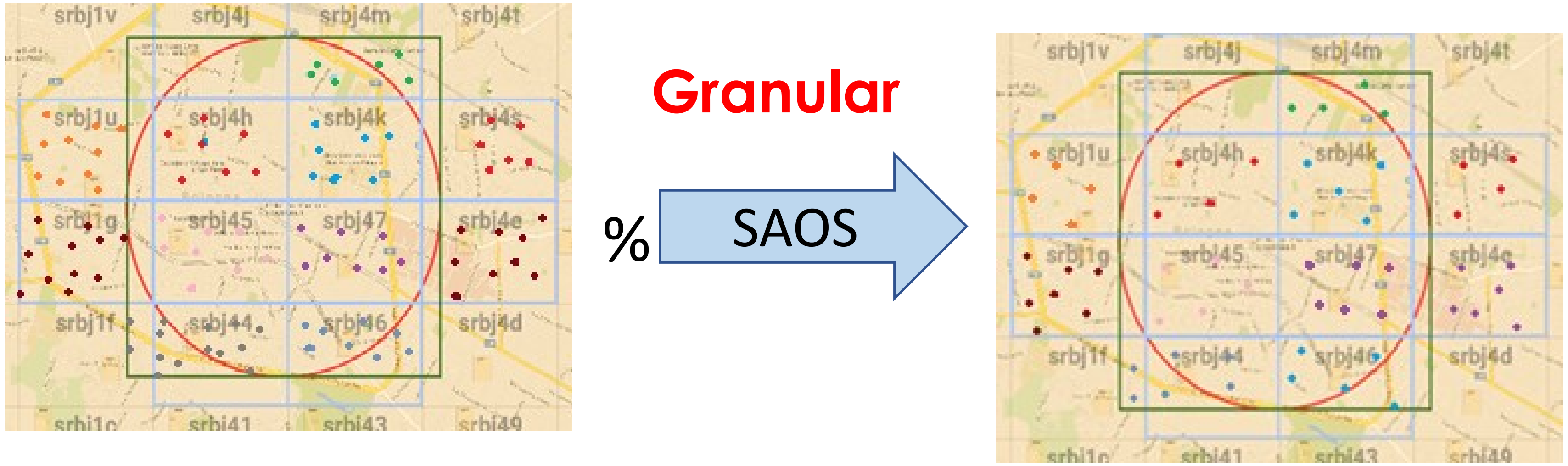


- Nearby points share the same geohash prefixes
- **SAOS** focuses on **SDL preservation**

Spatial Aware Online Sampling (SAOS): overview

- **Nearby** points share the same **geohash** prefixes, thus **reducing** the **two-dimensional** point representations to **one-dimensional** string **ordering**.

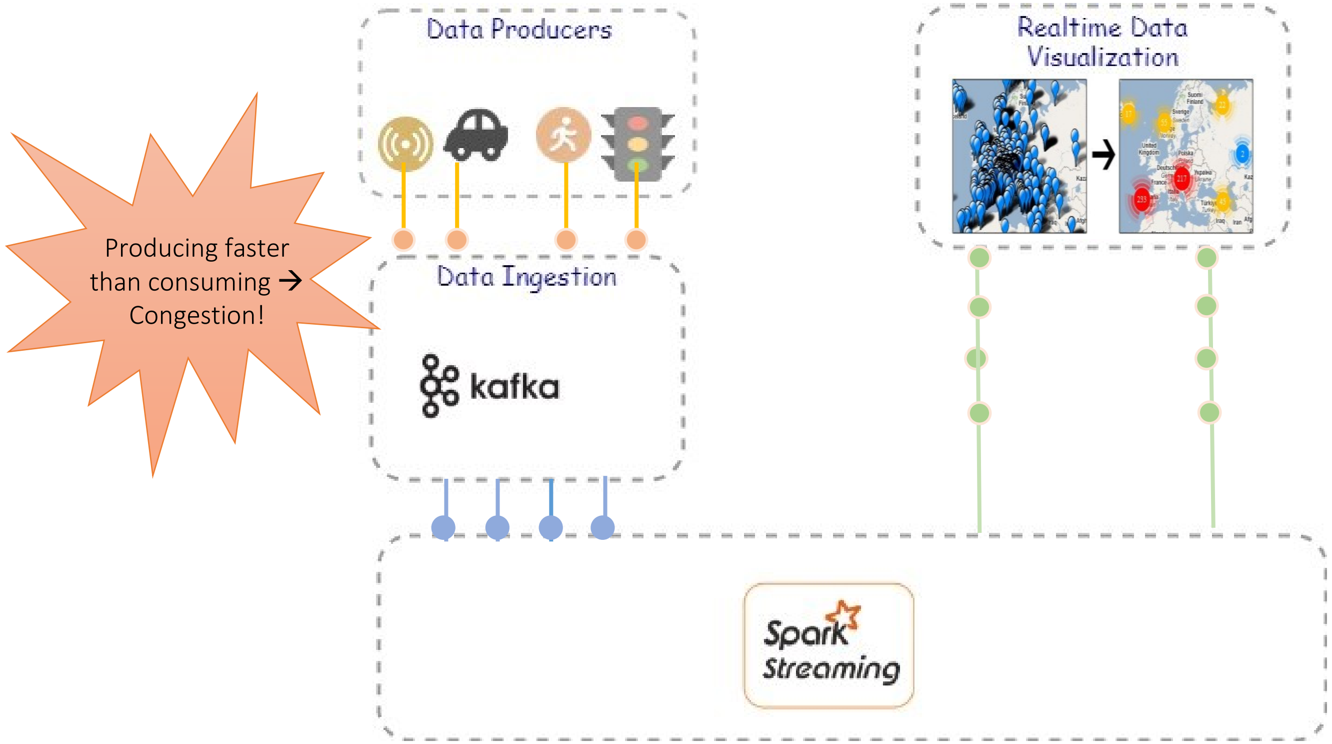
- **Geohash** indexing. An ordering (**string representation**) imposed on **grid** surface earth **planar** representation.



- Nearby points share the same geohash prefixes
- Only the ‘filter’ stage of the ‘filter-and-refine’!
- **SAOS** focuses on **SDL preservation,** but with ‘**false positives**’
- ‘**False positives**’ are those tuples that have the same geohash, but do not belong to the same neighborhood

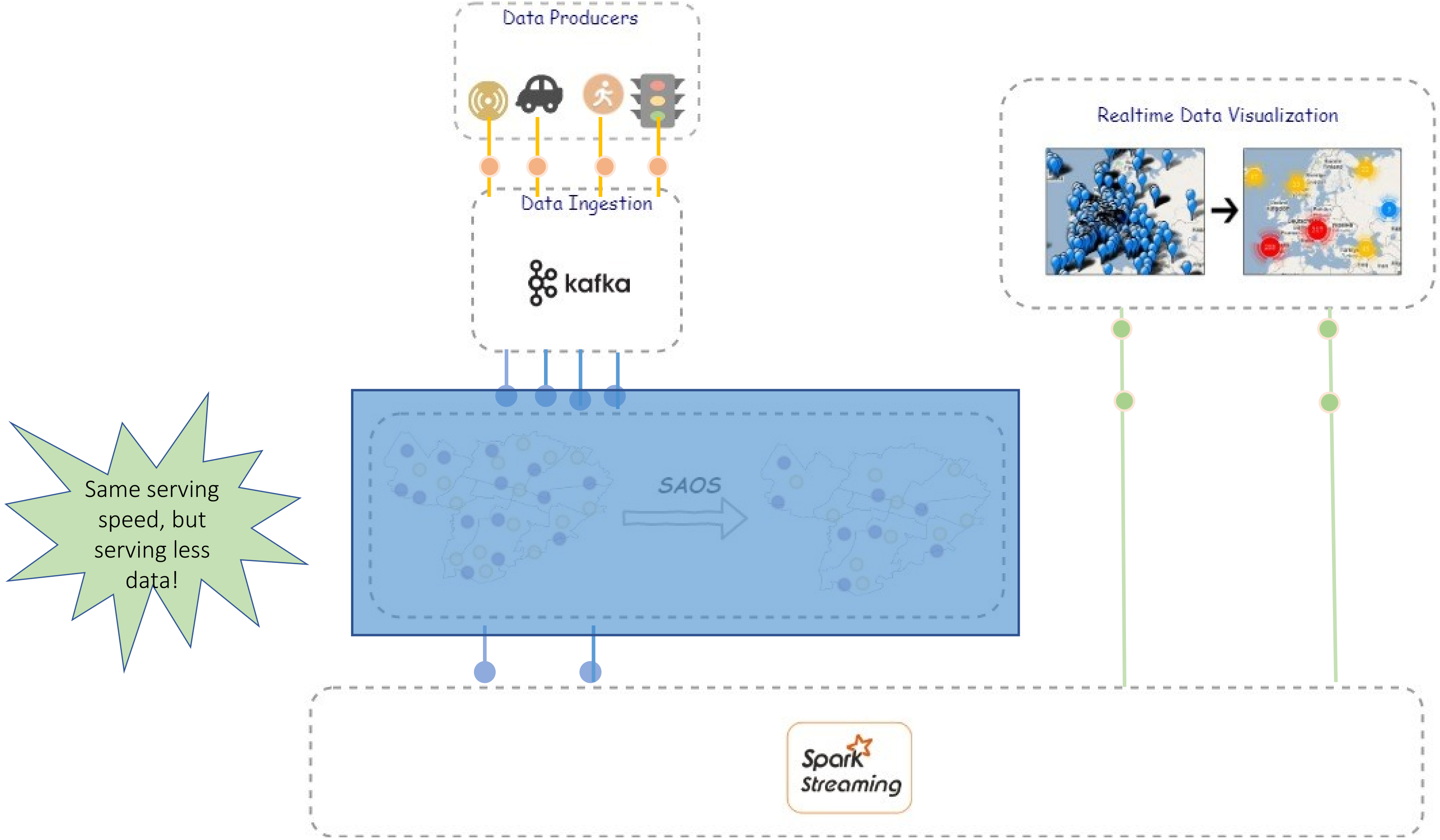
[Original work source]

Typical pipeline architecture w/o SAOS



Original work source

The improved architecture w/ SAOS



SQL-alike continuous geospatial query

[Original work source]

Spatial Queries Supported

- **Single spatial queries** (i.e., **linear**)
  - "find the average trip distance travelled by taxis originating from a specific district in a metropolitan city"
- SAOS resorts to a **stratified-like** sampling **design**, we depend on the **theory of stratified sampling** for estimations (e.g., '***means***', '***totals***', etc.,)
- estimating the '***average***' is formalized as follows.
  - Imagine that we have *K* **geohashes** in total (each **geohash** overlays a **stratum**, imagining both as **grid cells**),
  - y_{kj} is a value of a j_{th} tuple in **geohash** *k*, then t (pronounced ***tau***) is a **population** '*total*' for **stratum** *k*, which follows that a population '***total***' for the **target parameter** y is estimated by SAOS through applying the formula

$$\hat{t}_{SAOS} = \sum_{k=1}^{K} t_k = \sum_{k=1}^{K} N_k \bar{y}_k$$

Spatial Queries Supported

- using SAOS, the **average** is estimated by applying

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

- $\hat{t}_{SAOS}$ is the **estimated** '***total***' by applying SAOS,
- N is the **number of tuples** received thus far,
- N_i is the **number of tuples** received heretofore in **stratum** *i*,
- $\bar{y}_i$ is the **incremental '*average*'** in **stratum** *i* calculated up to now

Spatial Queries Supported

```
data.where("city = NY").groupBy(window("time","60
seconds").avg("trip_distance")
```

- "calculate the '***average***' trip **distance** travelled through all taxi trips in NY City, USA every minute"
- For SRS baseline, we first apply, to estimate the '***mean***'

$$\bar{Y}_{SRS} = \sum_{k \in SRS} y_k / n$$

- where y_i are the **values of target variables** in every **time window**, n is the **size** of the **sample** in every time window

stateful spatial online aggregation queries (i.e., ensembles)

"which are the top-10 boroughs in NYC where people tend to order green taxi pickups"

```
val sampleStatistics = sample .groupBy($"borough ", window($"time", "1 minute"))

  .count().orderBy($"count".desc)

val query = sampleStatistics.writeStream

 .queryName("statistics")…start()

statistics.select($"borough",$"count").limit(10)
```

- **Online aggregations** (as opposed to **static batch** counterpart)requires **managing state** between **batch intervals**
  - Top-N (a.k.a. top-K) online aggregations
- SAOS is applied to arriving spatial points ,
  - thereafter they are **grouped** by **geohash keys** (Also it is possible to group on a coarser level such as neighborhoods, boroughs, or districts),
  - and then a **count predicate** is applied **calculating tuples** number for **every geohash incrementally** and a **sorting** function is applied in a descending style.

Quantifying the Uncertainty Associated with Sampling

- **Estimating** target variables by **sampling** instead of the **population** is naturally **bounded** to an uncertainty
  - should be **quantified** to **measure** the ability of the sampling design in **achieving** the **QoS goals**
- Online spatial sampling that resorts to **stratified-like** sampling **design** → theory of stratification applies.
  - rely on the theory of stratified sampling and the theory of **random** sampling for quantifying the uncertainty of applying spatial queries in (linear) to estimate **target** variables

Quantifying the Uncertainty Associated with Sampling (cont.)

- estimations of the **accuracy** of **approximations** for ***single*** queries that are obtained by applying **stratified-like** online sampling instead of **SRS**

$$\hat{v}(\hat{t}_{SAOS}) = \sum_{k=1}^{K} (N_k - n_k/N_k)\ (N_k^2 s_k^2/n_k)$$

- Where n_K is the **number of tuples** thus far in **stratum** *k*,
- N_K is the **total number of items** up to now in all **strata**,
- s_K^2 is the **standard deviation** in **stratum** *k*.
  - All those magnitudes are calculated **incrementally**

- to compute an **estimated variance** for the **estimated total** → incorporate the result in an equation to estimate a **variance** for the estimated **average** of the **target variable**, by applying

$$\hat{v}(\overline{Y}_{SAOS}) = \hat{v}(\hat{t}_{SAOS})/N^2$$

Where $\hat{v}(\overline{Y}_{SAOS})$ is the **estimated variance** of the **estimated mean**, $\hat{v}(\hat{t}_{SAOS})$ is the **estimated variance** of the **estimated total**

Quantifying the Uncertainty Associated with Sampling (cont.)

Thereafter, we compute **standard error (SE)** depending on

$$SE(\bar{Y}_{SAOS}) = \sqrt{\hat{v}(\bar{Y}_{SAOS})}$$

we carry the value obtained of SE and apply it in

$$\bar{Y}_{SAOS} \mp z_{a/2} SE(\bar{Y}_{SAOS})$$

In order to approximate **100(1− a)% confidence interval** (CI) of the **population mean** $\bar{Y}_{pop}$, where $z_{a/2}$ is the **upper $\alpha/2$ point of normal distribution**

Thereafter we define **relative error**. SE **measures** sampling distribution **variability** (not to be confused with **standard deviation**, which measures the **variability** on points level)

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

The **intuition** behind this **adjusted error metric** is that **values** of SE metric are normally **small**, so we have used a **relative error** as a **representative** that **preserves** the same **SE trend** but being **more meaningful**

Quantifying the Uncertainty Associated with Sampling (cont.)

- We also define an **accuracy loss**
  accLoss = |estimatedMean – trueMean| / trueMean

- We also define the **gain** by applying **SAOS** **instead** of the **SRS-based** baseline

$$gain_{SAOS} = \hat{v}(\bar{Y}_{SAOS}) / \hat{v}(\bar{Y}_{SRS})$$

where $\hat{v}(\bar{Y}_{SAOS})$ is the **estimated** **variance** resulted by applying SAOS, whereas $\hat{v}(\bar{Y}_{SRS})$ is the **estimated** **variance** resulted by applying an **SRS** baseline

Quantifying the Uncertainty Associated with Sampling (cont.)

- apply the following equations from the **theory of SRS** to calculate the **estimated variance estimated average** and other **quantities**
- calculate the **estimated variance** of the **estimated mean**

$$\hat{V}(\bar{Y}_{SRS}) = (({}^{N-n}/_{N})({}^{s^2}/_{n})$$

N is the total **number** of **records** arrived at the system **at the time of computation**, s^2 is the **incrementalized variance** calculated from the **sample** drawn thus far

calculate the **standard error**

$$\text{SE}(\bar{Y}_{SRS}) = \sqrt{\hat{V}(\bar{Y}_{SRS})}$$

calculate a **relative error**

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

Quantifying the Uncertainty Associated with Sampling (ranking geo-statistics)

- **online spatial stateful aggregations** (specifically **Top-K**) queries
- measure every method ability in **preserving** an original **ranking** that would be obtained if we have access to a **population** or a **superpopulation**
    - **online stateful aggregations** → compute by **sampling** instead of **population**
- apply a **Spearman's** rank correlation coefficient (read **Spearman's *rho***)
    - A measure for **statistical dependency** between the **ranking** of **two variables** in a dataset

Quantifying the Uncertainty Associated with Sampling (ranking geo-statistics) --- cont.

- our application of ***rho***
  - **collect** the **ranks** (i.e., **orderings**),
    - and once the spatial **CQ** **stops** (i.e., **shutdown** by user, or **depending on a query window** semantics) we take the collected **orderings** of the original **aggregations** (i.e., those that would result from a population without sampling, we consider the **total number of tuples** **emitted** by the sources at that point as the **population**)
    - and the ranking that is calculated by applying the online sampler (same applies to SRS baseline)
    - Then we serve those figures to **Spearman's** ***rho*** and apply

$$\rho_{rg} = \text{covariance}(\text{rank}_{\text{nosampling}}, \text{rank}_{\text{sampling}}) / (\sigma_{\text{rank}_{\text{nosampling}}} \cdot \sigma_{\text{rank}_{\text{sampling}}})$$

where ρ_{rg} (i.e., *rho*) is spearman's **correlation coefficient** applied for **ranking** **statistics** ,

covariance$(\text{rank}_{\text{nosampling}}, \text{rank}_{\text{sampling}})$ is the **covariance** of the **rank** variables,

$\sigma_{rank_{\text{nosampling}}}$ and $\sigma_{rank_{\text{sampling}}}$ are the **standard deviations** of the rank variables, without and with sampling, respectively

Summary of geo-statistics

Single queries

"average trip distance travelled by taxis in Rome, Italy during the last three months"

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

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

Online continuous query

Incremental geo-stats

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

Stateful aggregation

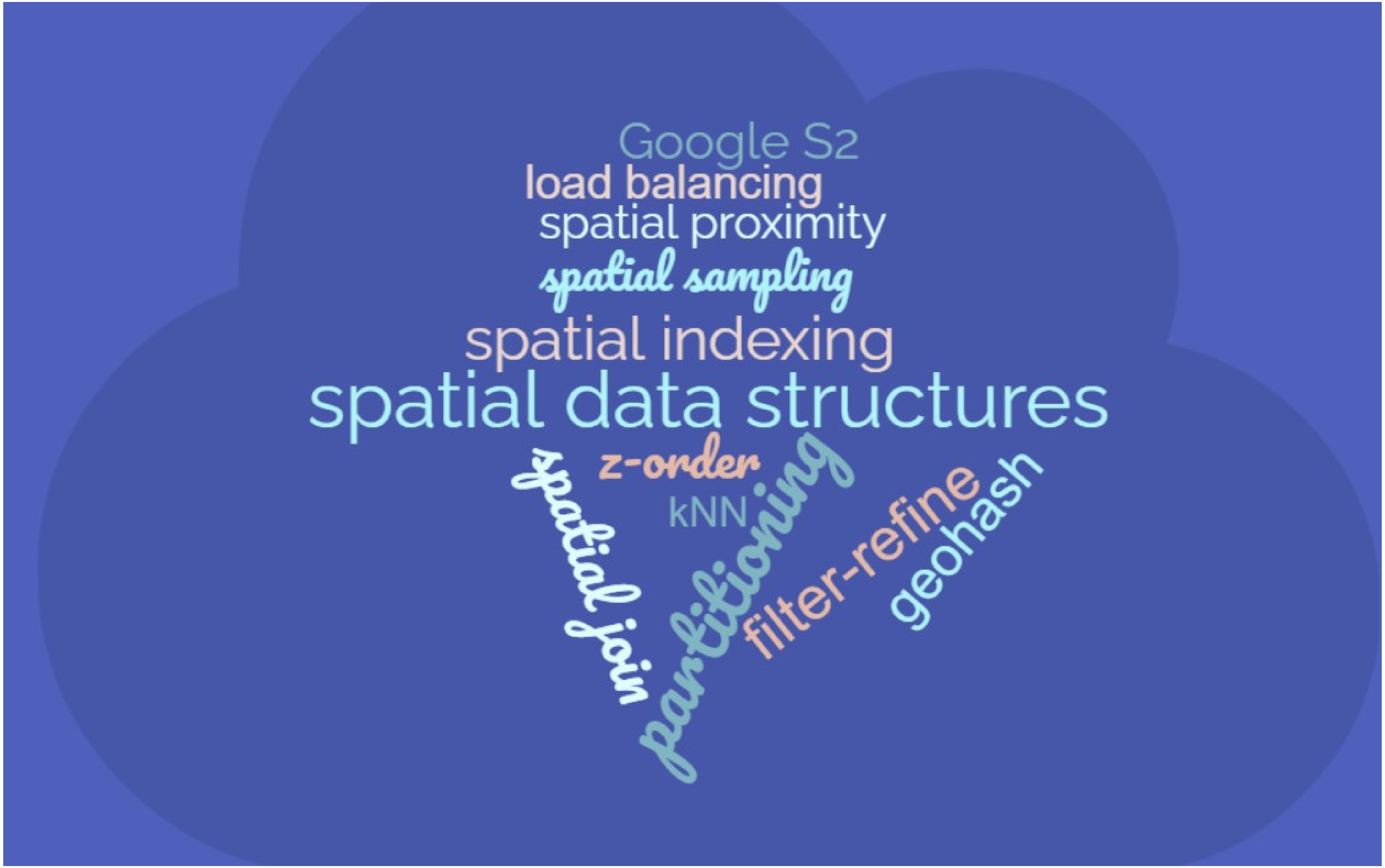
"top-3 boroughs in Bologna city in Italy where people tend to check out shared bikes".

spatial incremental Top-N online aggregation function

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

No pre-knowledge on the streaming geo-statistics is required, we depends on ***incrementalization***

Original work source


Google S2
load balancing
spatial proximity
spatial sampling
spatial indexing
spatial data structures
z-order
kNN
spatial join
partitioning
filter-refine
geohash