# Disk-Stream Spatial-Aware Join Processing in Real-Time Data Warehouses

isam.aljawarneh3@unibo.it

## *ABSTRACT*

The advancement of mobile technologies has enabled the accumulation of huge amount of spatial data that is mostly georeferenced (tagged with multidimensional GPS coordinates for example). This data is normally utilized for unleashing the power of spatial-oriented advanced analytics. Smart city spatial data are used for healthcare, pollution and crowd management, to mention just a few. As a matter of fact, this data arrives in *streams*, and mostly are joined with *disk-resident* data aiming at answering an interesting query. Consider the following example; a patient (with a chronic disease; for example, heart failure) wearing a sensor-enabled device (communicating her real-time location and health status) is commuting in a city, where we seek to find the most reliable person to provide a first-aid in case of emergency (heart attach for example). Finding such person requires searching for all nearby passing-by volunteers (based on some distance proximity constraints), which is in this sense a *stream* of data. Thereafter, selecting the profile of each of those (*disk-resident* data, data already stored in disk), thus joining the stream data with *disk-resident* data. Thereafter, electing the most suitable passing-by volunteer who satisfy spatial-oriented conditions (for example, has been there before and was able to provide first-aid for similar emergencies). Obviously, this is compute-intensive, and scanning all disk-based elements is infeasible. Perhaps most significantly, data need to be joined based on a spatial-key (for example, longitude and latitude), which exaggerates the complexity of the problem. State-of-the-art solutions are relevant for generic workloads (where the join key is not spatial), thus are regarded irrelevant in the context of spatial-referenced workloads. Therefore, novel feasible solutions are required to accomplish a performant *disk-stream* join processing for spatially-tagged big datasets. The implementation is recommended to be based on Apache Spark Streaming (for data streams) and MongoDB (for *disk-resident* datasets). Figure 1 tells the story.

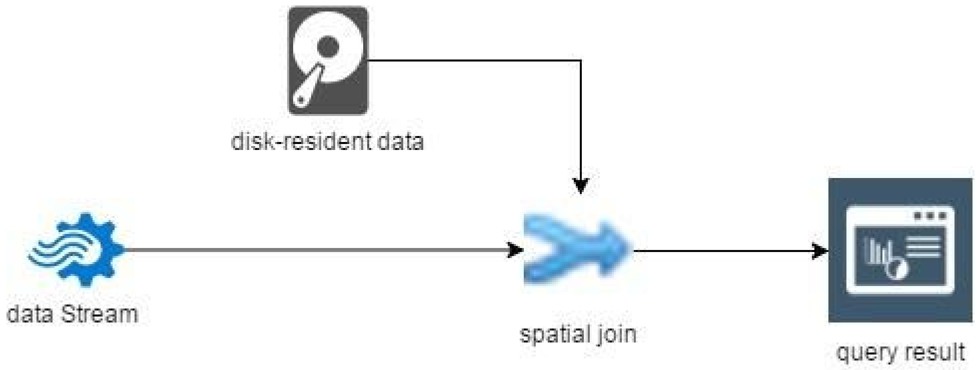

*Figure 1*: Disk-stream spatial join.