Keywords: Scientific Data Management, Spatial Indexes, Neuroscience

Context: Scientists in all kinds of disciplines like biology, chemistry, physics etc. produce vast amounts of data through experimentation and simulation. The amounts of data produced are already so big that they can barely be managed. And the problem is certain to get worse as the volume of scientific data doubles every year. In the DIAS laboratory we are working on next generation data management tools and techniques able to manage tomorrow’s scientific data.

We work with neuroscientists in the Blue Brain Project (http://bluebrain.epfl.ch) to manage the vast amounts of data they produce. Their research, modeling and simulating a fraction of the rat brain, already produces gigabytes of data. With the recent upgrade of their computing infrastructure (IBM Blue Gene/P), the volume of data will soon be in the order of terabytes.

Current solutions are inadequate to manage this data volume and we are thus investigating new methods to index and store it in order to provide efficient access. A particular problem we are currently addressing is the retrieval of objects in space, i.e., accessing neurons based on their position (“What elements are in a given region of the brain?”). While it is simple to index several thousand neurons, we will have to do it for several millions or even billions of neurons.

Project: To analyze the spatial models efficiently, the neuroscientists load them in partially or entirely into main memory (depending on available main memory) of their desktop machines. Most spatial indexes are however not efficient when used in memory, because they are optimized for disk.

In this project the student will implement a spatial index optimized for use in memory. The implementation will be loosely based on the CR-Tree [1] and will be done in a readily available spatial indexing library [2].

The development of the new index will be based on two key insights. First, compressing the index structure in memory has the biggest potential for improving performance [1]. Second, as opposed to indexes on disk, the size of nodes of in-memory indexes can be chosen more freely and more fine granular (one is not restricted to chose the disk page size): the performance is equally well for a range of different node sizes [3].

As opposed to produce nodes/partitions of equal size, this project thus investigates if the STR bulkloader can be adapted to produce variable size partitions, which can be
compressed at a higher ratio. The partitions are chosen so that the values in each dimension are in a range that can be compressed at a higher ratio.

**Milestones:**
1. Get acquainted with STR & the spatial indexing framework
2. Get acquainted with the CR-Tree implementation
3. Study the IEEE 754 standard about floats and doubles
4. Develop analytical model for the compression ratio
5. Implement the compression optimized STR bulkloading approach
6. Thoroughly measure performance of the R-Tree bulkloaded with the new STR variant

**Knowledge:** C/C++

**Supervisor:** Prof. Anastasia Ailamaki (anastasia.ailamaki@epfl.ch)

**Responsible** Thomas Heinis (thomas.heinis@epfl.ch), Farhan Tauheed

**Collaborator(s):** (farhan.tauheed@epfl.ch)

**Duration:** 2 months

**References**