Planning for case- and performance study

In the existing Stride software, we have added new features. By using the available data about population in different cities, the commuting behaviour of the population, about the households’ composition, etc we can generate different types of communities like schools, colleges, workplaces and we can assign population to those communities. We can visualise those information in a map and run simulation by using GUI.

For the case- and performance study we will be using ***GNU gprof*** and ***Performance Application Programming Interface (PAPI).*** GNU grof is a performance analysis tool which is used for profiling. By using this tool we will be able to find out the execution time required by different pieces of the program. We will also be able to visualize which functions called which other functions during the execution. In this way we will be able to see which part of the program take the most execution time and we will get the information about the functions which will be critical for the better performance. It provides different form of output which will be useful for our study.

The different types of output which can be useful for our case study are:

* **Flat profile:** for the information about how much time functions spend and how many times the function is called.
* **Call graph:** we will see which function calls which functions and how much time is spent on subroutines
* **Annotated source:** it will give the copy of the source code by showing how many times each line of program is executed.

***Performance API (PAPI)*** provides interface and methodology for use of the performance counter hardware. We will be able to find the relation between our software performance and performance event. We will be able to find the correlation between the structure of source code and the efficiency of the mapping of that code to that underlying architecture. The low level PAPI interface deals with hardware events in groups called EventSets. We will be able to see how the counters are most frequently used. This will relate different hardwares with one another for example relating cycles to memory references.

So for all the hardware related performance factors we will be using PAPI and related with our software will be determined by using GNU gprof. In this way we will find out the key performance factors for the epidemiological model.