Utilizing High Performance Computing Techniques for efficiently anonymizing sensitive patient data

D. Ntalaperas, A. Bouras
UBITECH Ltd
Di Fuccio Raffaele
University of Naples
e-mail: dntalaperas@ubitech.eu

Keywords: Data Anonymization, High Performance Computing

1. Introduction

Using Data Cubes for anonymizing large sets of patient data is an approach that has been well researched and has been demonstrated to provide a faithful representation of the original content, while also ensuring that the transformed data do not contain any information that can be reversed engineered to identify patients (Antoniades et al. 2012). The Data Cubes approach has been successfully used previously in the context of the Linked2Safety[1] project (Perakis et al. 2013) and is now being used in the context of the SAGE-CARE project[2] to transform clinical and genetic data of patients being treated for melanoma, in a representation that is safe to distribute among institutions.

The methodology that achieves this result has already been implemented and documented (Ntalaperas et al. 2016); the main purpose of the present work is to describe how the methodology is enhanced by using High Performance Computing (HPC) in order to be able to cope with very large data sets. Indeed, in the case of genetic data it may be possible that the size of original data may become very big, since the gene expression information or Single Nucleotide Polymorphisms (SNPs) that need to be recorded per patient may be in the order of hundreds of thousands.

2. Theory

Figure 1 depicts the methodology pipeline for creating a Data Cube from source clinical and genetic data. Data come from two files, one for clinical and one for genetic data. A module aligns the data so that rows and columns of the source data are correctly aligned based on patient and, if a mapping file is given by the user, a categorization schema is applied to variables selected. The categorization schema defines the ranges that are going to be used for aggregation; a category for variable BMI of 0-25 for example, denotes that all patients with a BMI lower than 25 will be treated the same and will be aggregated together. After categorization, the values for each category are aggregated and a Data Cube is created. Data are then perturbed by adding a small, statistically insignificant, random noise and cells with small values are removed. These last two actions, ensure that the resulting data set is fully anonymized (Antoniades et al. 2012), (Forgó et al. 2012). A complete description of the methodology can be found in (Ntalaperas et al. 2016).

[1] http://www.linked2safety-project.eu/
[2] http://www.sage-care.eu/

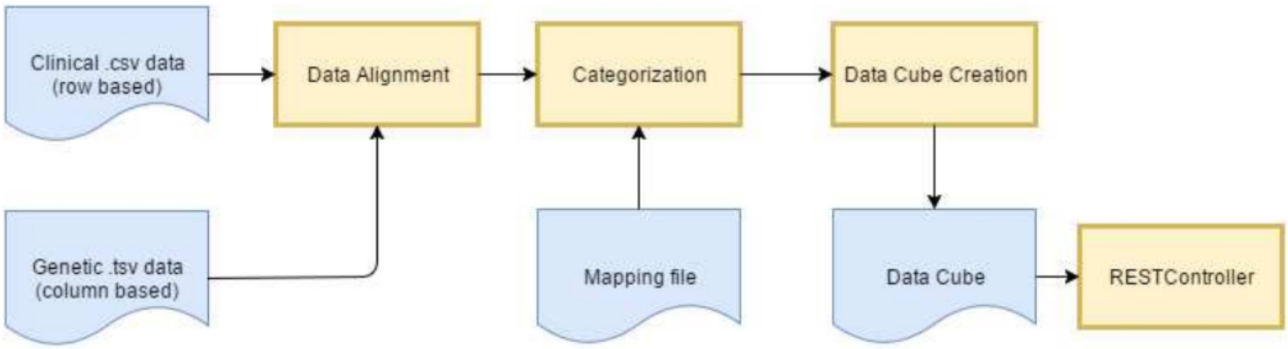

Figure 1: Data Cube Creation Methodology Pipeline

The most time consuming operations in the above pipeline are: a) the Data Alignment where each entry of the original two files need to be matched, b) the categorization where all data need to be traversed and be transformed to the categorical values defined in the mapping file and c) the aggregation of the data. Indeed, it is these three actions that involve operations that need to be performed on every entry of the source data.

3. Implementation and Results

The Data Cube creation algorithm was enhanced by parallelizing the most intensive, in terms of computation, steps. For the Data Alignment step, since each patient entry is independent from the other, a high level of parallelization was achieved. In case of N processors, each processor can be used to combine the two entries of the source files and construct the internal data representation that is to be used by the next steps. As is the case with the non-parallelized version, the main cost of this operation is the time required for I/O access.

In the case of categorization, there are n patient records and m variables to be categorized with each categorical variable having a list of m_i possible values. Parallelization can again be achieved to a high degree due to the fact that the categorization schema is applied to each one of the patient records independently. The only shared data is the mapping file and this is a read only, usually very small file, that can be loaded once in the memory. Data Cube creation finally, is a simple aggregation process where the values of each one of the cells of a multidimensional array is computed by aggregating the number of patient records that share a combination of categorically equal values that is represented by each cell.

Figure 2 depicts the typical work that can be performed by each processor during one iteration of the pipeline in the ideal case where the number of processors equals the number of the patients. Firstly, the processor will align the clinical and genetic data into a single entry. In the second step, it will iterate through each entry of the combined data and will perform a categorization operation according to the schematics defined in the Mapping file. Then, for the combined categorized entry, the corresponding cell in the Data Cube will be incremented. Consider for example that there three variables being monitored; namely BMI, smoking and diabetes. Suppose further that the categorized values of these variables for a specific patient are BMI=1 (corresponding to a normal BMI), smoking=1 (corresponding to a casual smoker) and diabetes=0 (corresponding to the patient not having diabetes). In the above described case, cell *D[1][1][0]* will be incremented by the processor, so that when all processors end, cell *D[1][1][0]* will have a value equal to the total number of patients with normal BMI who are casual smokers and do not have diabetes.

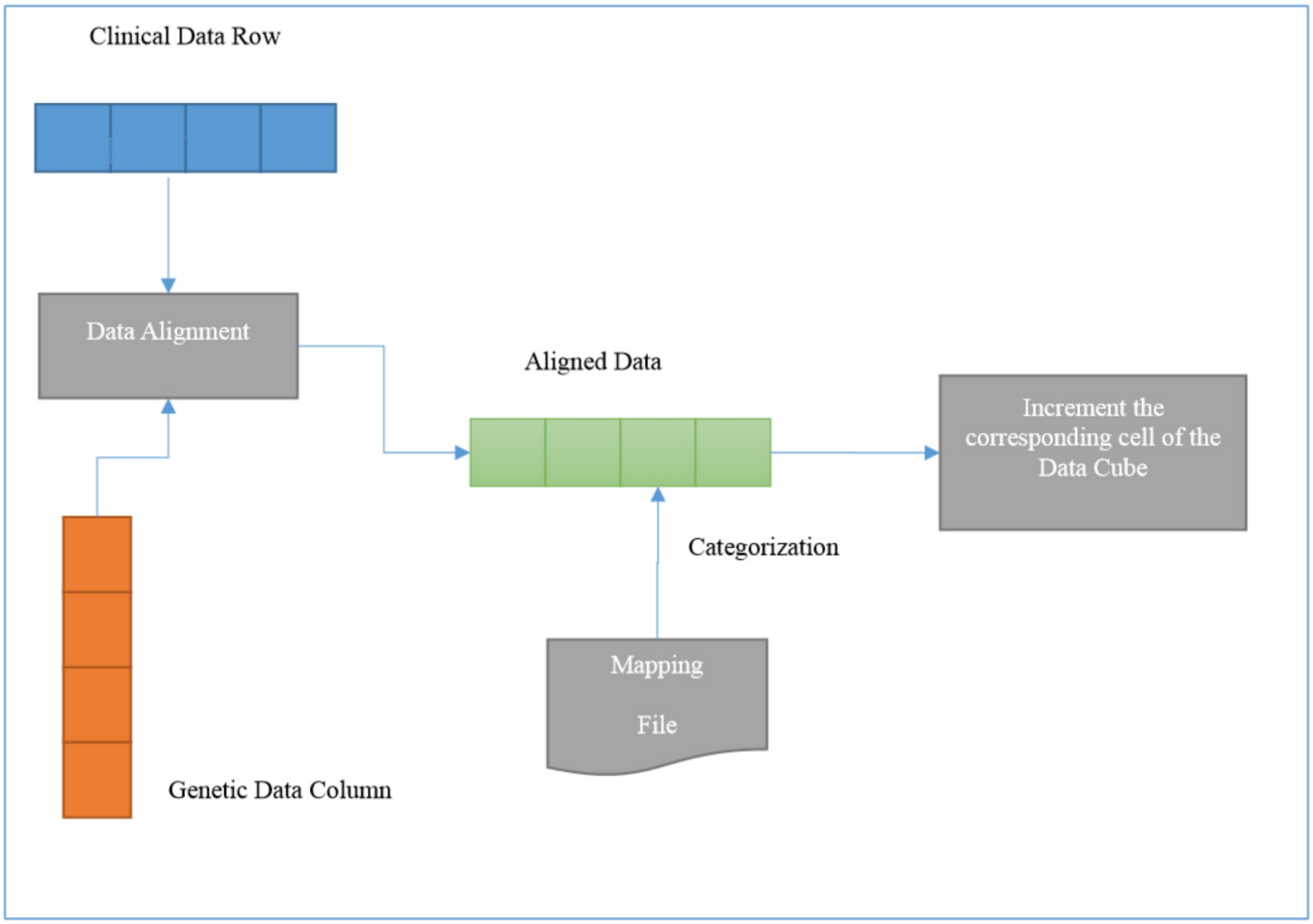

Figure 2: Workload for each processor

The above parallelization methodology has been implemented in the context of SAGE-CARE using the CUDA framework and was tested in a GeForce GTX 960M card. The control machine was the same as the test, the difference being that in the control the non parallelized version of the algorithm was run. First results in the case of a sample of 100.000 mock patients being monitored for three clinical and one genetic variable are depicted in Table 1.

Table 1: Measured response times for the parallelized and non parallelized algorithm. Times are averages of twenty runs

Non parallelized algorithm (ms)	**Parallelized algorithm (ms)**
4.817	12.8

It can be seen that the speedup achieved by the implementation is of the order of 10^2. This can somewhat be understood empirically, since the number of cores present in the GTX 960M card are equal to 1024. So, in theory, the parallelized algorithm should run 1000 times faster; however idle

core times introduced during the transfer of data between the RAM and the GPU memory seem to have a limiting effect.

4. Conclusions

Using HPC has been demonstrated to offer a significant speedup during the generation of anonymized patient data. This facilitates the real time request and acquisition of data instead of having to wait or having to pre-generating data that could become outdated.

Future work consists of evaluating the response times of the algorithm to various combinations of sample test data as well as monitoring and measuring processor idle times. The latter will help to determine more accurately what are the bottlenecks that introduce the biggest deviations from the theoretical minimal times and it will provide hints to further enhance the efficiency of the parallelization schema.

References

Antoniades, A. et. al. (2012). "The effects of applying cell-suppression and perturbation to aggregated genetic data", in: *12th Int. IEEE Conf. on Bioinformatics & Bioengineering (BIBE)*, IEEE.

Forgó, N., Góralczyk, M. and Graf von Rex, C. (2012) "Security issues in research projects with patient's medical data", in: *12th Int. IEEE Conf. on Bioinformatics & Bioengineering (BIBE)*, IEEE.

Ntalaperas, D., Mpouras, A. (2016). "An approach for anonymization of sensitive clinical and genetic data based on Data Cube Structures", in: *Collaborative European Research Conference, 2016*.

Perakis, K. et. al. (2013). "Advancing Patient Record Safety and EHR Semantic Interoperability", in: *IEEE International Conference on Systems, Man, and Cybernetics,* IEE.