FASTA/Q data compressors for MapReduce Hadoop genomics.

Abstract:

Storage of genomic data is a major cost for the Life Sciences, efectively addressed via specialized data compression methods. For the same reasons of abun-dance in data production, the use of Big Data technologies is seen as the future for genomic data storage and processing, with MapReduce-Hadoop as leaders. Somewhat surprisingly, none of the specialized FASTA/Q compressors is available within Hadoop. Indeed, their deployment there is not exactly immediate. Such a State of the Art is problematic.

Introduction:

Tis circumstance is managed by HDFS through the introduction of the input split concept or split, for short. It can be used, at the application level, to logically redefne the range of data to be processed by each map task, thus allowing a map task to process data found on HDFS data blocks diferent than the one it is processing.

Related works:

Due to the same reasons of massive data production, the development and use of Big Data Technologies for Genomics and the Life Sciences, have been indicated as directions to be actively pursued with MapReduce Hadoop and Spark being the preferred ones . Tis is not just a following of a “Big Data trend” that has proved successful in other felds of Science, since Bioinformatics solutions based on those techniques can be more efective than classic HPC ones, thanks to their scalability with available hardware and to their easiness of use. A map task can extract and decompress the compressed data blocks existing in the HDFS data block it is analyzing only if it knows their size and relative positions.

. Ten, each computing node makes available this information to the map tasks that it runs, thus allowing them to determine the list and the relative position of the compressed data blocks in their HDFS data blocks.

: methodology

An assessment of reading times savings The aim here is to deter mine if there is a positive trade-of between the time saved thanks to the smaller amount of data to read from HDFS and the cost to be paid for reading and unpacking compressed FASTA/Q files, once compressed with an HS or an HU Codec. Following, the methodology used in [22], this experiment is implemented by benchmarking a very simple Hadoop application. It runs only map tasks whose goal is to count the number of occurrences of the letters {A,C, G, T, N} in the input sequences, without producing any output. That is, the application spends most of its time reading data from HDFS. In what follows, we refer to such a task as benchmarking. Speedup has been evaluated by dividing the overall execution time of experiments each uncompressed file with respect to the overall execution time of the same experiment, but run on the same compressed file. The results are reported in Figs.



NAME/ Anas Samir ahmed