Parallel file compressor

Francesco Corti - 606922

May 2020

Abstract

The purpose of the assignment was to implement a parallel file compressor using building blocks provided by Fastflow library and Miniz open source compressor library.

1 Code structure

The program has a single file *ffc_farm.cpp* that take: *nworkers threshold file-or-directory* as input arguments. This can be compiled using the Makefile provided. The program expose three main component that are join together using the *ff_pipeline* object available inside the *FastFlow* library. These three components are the *Reader*, the *Compressor* and the *Write* struct. These three structs implement the pure virtual method *svc()* from the upper class *ff_node<T>* that is the standard *Fastflow* node.

2 Problems faced

The first problem that I faced was to deal with the split of the file in multiple chunks. To deal with the split of the big file in chunks I had to modify the *Task* object that was already given in the file. I added two new parameters: *chunkPointer* and *chunkName*. This was used to compress the correct chunk part of the file and to recognize the part of the file that was compressed in the chunk.

The second problem that I faced was to store in the correct position all the compressed chunks produced by the *Compressor* struct and received in *svc()* method of the *Writer* struct. I used the filesystem library available from C++17. This library was recently integrated into the standard library to replace the *Boost::filesytem* and let the programmer manage in a very simple way the filesystem. The program creates a *Files* folder and when a new chunk *arrive* in *Writer*, it checks if exists already a directory inside *Files* to store all the compressed chunks of a specific file. If there isn't it creates the directory and then stores the compressed file inside it. Otherwise store directly the chunk inside the correct directory. To store all the compressed file in a single file I exploit the definition of the *svc_end()* method present in *Fastflow*. Indeed this method is called after the *svc()* method is ended, so in our case all the compressed chunks have already been stored inside their folder, inside the *Files* directory. So at the end the program moves itself inside the folder *Files* using the *chdir* method and then iterate inside the directory calling the command *tar -zcf* on the directory containing all the compressed chunk of a specific file. This zipped folder is then stored inside *mergeZippedFiles* folder and after all the folders containing the compressed chunks are stored the *Files* folder is deleted.