STATIC ANALYSIS FOR CLASSIFICATION OF THE CANDIDATE PROCEDURES TO THE TASKIFICATION

Supervised by :Therese Lepage
Realized by :
Karim SMAIL
Sofiane BOUZAHER
Asma KREDDIA
Atef DORAI

April 27, 2020

SPECIAL THANKS

We wanted to write a sincere "Thank You" for your support Professor “Therese Lepage “, teaching and guidance throughout this past year. We want you to know that your students will remember this year of study with great joy for the rest of their lives.

VOCABULARY SPECIFIC TO OUR PROJECT

- **TASKIFICATION** : Taskification can be defined as a kind of automated parallelisation, and more precisely it is the transformation of an already existing code into a code based on well-identified tasks.
- **LLVM (Low Level Virtual Machine)** :is a compiler infrastructure designed for code optimization at compile time (it is an infrastructure that does not contain the necessary tools to compile C or C++ source code but only tools for optimizations and generation of machine code from an intermediate format).
- **CLANG -LLVM**: the overall structure of this infrastructure at the microscopic scale is built up in a way similar to any modern compiler.
- **COMPILATION**: defined as the set of steps that transform a code (. C) into an object code (. O). This is done using a program called 'Compiler'.
- **COMPILATOR** : a compiler is a program that transforms source code (written in a high-level abstraction programming language) into object code (written in a low-level programming language) in order to create a program executable by a machine.
- **STATIC ANALYSIS**: defined as the set of processes used to obtain information on the behaviour of a program during its execution without actually executing it.
- **DYNAMIC ANALYSIS**: unlike static analysis, is an analysis that requires the execution of the program and studies its behavior + the effects of its execution on its environment.
- **SHARED MEMORY(INTER-PROCESS COMMUNICATION)**: can be explained as being the memory segment allowing the access of several processes (the access of the different processes to the shared memory is ensured by synchronization).
- **DISTRIBUTED MEMORY**:a distributed system is defined as a set of physical and logical resources that are geographically dispersed and connected by a communication network in order to perform a common task. This set gives users a single view of the data from a logical perspective.
- **PURE FUNCTION** : is a function that returns always the same result with parameters are identical and its execution does not depend on the state of the system.
- **IMPURE FUNCTION**: is a function that has edge effects (as it may accidentally not have any).
- **PLUGIN**: a plugin or plug-in is a package that complements a software in order to give it and bring it new functionalities.
- **FRONTEND**: this is the first block of any compiler, its objective is to validate that the program is syntactically and semantically correct and then to translate it to an intermediate representation (IR for Intermediate Representation) one of the goals of this intermediate representation being to simplify the work of other blocks that cannot work with the complexity of C or even worse C++ source code.

- **MIDDLE END** : are in charge of analyzing and/or transforming the IR by optimizing certain things while preserving the semantics of the code. Its objective was very often the maximization of the code performance (for example by playing on the code size).
- **BACK END**: is in charge of transforming IR to machine code for a given architecture. In order to intervene at the different levels of the compilation chain, several tools will be used.
- **AST**: is a transformation of the source code into a hierarchical representation by the parser of our clang compiler.
- **MPI**: or Message Passing interface, is a standard designed for the passage of messages between remote computers or in a multiprocessor computer (so it is a means of message transfer). Created for better performance, it has become a communication standard for nodes running parallel programs on distributed memory systems. It defines a library of functions, usable with C, C++ and Fortran languages.
- **OPEN MPI (MPI+X) (an MPI library)** :is a project library that serves to combine the expertise, techniques and resources of the entire HPC community to create the best MPI library available.
- **RPC or REMONTE PROCEDURE CALLS** : is defined as a means of communication that allows procedures to be called on a remote computer using an application server.
- **AUTOMATIC PARALLELIZATION** : this is one of the stages of compilation of a program at the level of from which the source code is transformed into a parallelised executable for multiprocessor computers symmetrical, with the aim of simplifying and reducing the development time of parallel programs.
- **SMP or SYMMETRIC SHARED MEMORY MULTIPROCESSOR**: the main purpose of this parallel architecture is to multiply the number of identical processors on a computer, in order to improve computing power.
- **OPEN MP (OPEN MULTI-PROCESSING)**: is a library supported by several languages (C, C++ and Fortran) available on several platforms (Linux, Windows, OS X, ...). OpenMP groups together compilation directives and functions. The CLANG LLVM compiler supports OpenMP.

PROJECT SPECIFIC ABBREVIATIONS

- **MPI** : Message Passing interface.
- **AST** :Abstract syntax tree .
- **RPC**: Remonte Procedure Calls.
- **IR** : intermediate representation.
- **BE** :Back-End.
- **FE** :Front-End.
- **ME** :Middle-End.
- **LLVM** :Low Level Virtual Machine.
- **SMP**: symmetric shared memory multiprocessor.
- **OMP** :Open multi-processing.
- **SM** : shared memory.
- **DM** : distributed memory.