2SC7694 – Energy optimization and acceleration of a cloud financial calculation graph

Instructors: Stephane Vialle
Department: DOMINANTE - MATHÉMATIQUES, DATA SCIENCES, DOMINANTE - INFORMATIQUE ET NUMÉRIQUE
Language of instruction: ANGLAIS
Campus: CAMPUS DE PARIS - SACLAY
Workload (HEE): 80
On-site hours (HPE): 48,00

Description

Project topic in partnership with ANEO.

Application Context

Modern **insurers** have a highly regulated but at the same time relatively broad field of activity: different types of insurance, banking services, etc. One of the difficulties in assessing the accounts of an insurance company (or bank) lies in the valuation of financial assets (e.g. EDF shares, life or car insurance contracts, etc.) and the underlying risks. The approach generally used consists of evaluating for each asset the cost of a devaluation as well as the associated risk. For simple assets such as shares, the calculation is simple. For more complex assets such as insurance products or derivatives, the calculation is more complex since it is usually based on the consideration of many factors. Depending on the valuation of the risks taken, regulations resulting from various economic crises, such as Solvency II or IFRS17, require the insurance or bank to tie up a certain amount of equity capital. Certain risks may cancel each other out between different assets (e.g. the risk of a life insurance asset based on the euro/dollar rate may be covered by other assets based on the euro/yen and yen/dollar rates). In order to maximise this potential for offsetting through the assets owned, these bodies will consolidate the accounts on the widest possible scale, the group as a whole. That is, they will carry out risk analyses as if all assets belonged to a single entity. One of the difficulties of the exercise then consists in distributing the capital requirement among the different legal entities whose accounts have been consolidated, so this process is actually more complex than a simple pooling of assets followed by a global risk analysis.

The ***process of constructing consolidated accounts for an insurance company therefore generates numerous calculations***. These calculations concern, on the one hand, the modelling of the cost of repayment of contracts according to various factors and, on the other hand, the modelling of investments made with the money available. As an example for a life insurance contract, risk modelling is based on mortality tables provided by INSEE and taking into account different factors such as geography, socio-professional category, family situation, etc. To do this, the life of the contract is simulated year after year in order to take into account changes in these factors. Different scenarios are played out in order to reflect all possible changes in situations (moves, changes in family situation, etc.). These scenarios are then aggregated. This process is of course a simplified view and does not take into account various elements such as the aggregation of contracts in order to reduce the volume of calculations, which aggregation is in itself the subject of various optimization works. Other elements of the process include the consolidation of asset/liability risks by contract type, consolidation by legal entity and taking into account the specific regulatory requirements of each country, and the use of these simulations to optimise the risk of the contracts offered as well as their price.

Problem addressed in this project

The process of constructing the consolidated financial statements takes several weeks and includes calculation steps as well as manual steps; we will consider the latter here as instantaneous. The calculation steps correspond to the equivalent of 413177 hours of calculation time, i.e. just over 10 full days on an infrastructure of 1700 cores. However, in reality, the process cannot take place in 10 days on such an infrastructure because of the dependencies between the computing tasks: there are times when there are not enough tasks to occupy the grid. A fine analysis of the dependencies shows that the critical path duration is 11h30. This duration would be that of the whole computation if an infrastructure of infinite size was available.

Project subject: In order to optimize costs without investing in a very large computing grid that would ultimately be little used, we want to use on-demand resources available in the cloud. To make the most of this, **we want to optimize the execution of the task graph** by searching for :

- The best strategy for switching compute nodes on and off.
- The best scheduling of tasks on the available nodes.

The study will have to take into account the following elements:

- The dependencies between tasks
- The duration of the tasks, known in advance
- The duration of the transfer of results between tasks (only the relevant files will be listed).

It should be noted here that the cost mentioned can be energy as well as financial, and that the two are closely linked: in use, more than half of the cost of owning a computing infrastructure corresponds to the cost of electricity, even in France with nuclear energy. We will make the (very simplifying) assumption that the network has no cost.

Project objective: to provide an optimization application working on two files describing on one hand the task graph (durations and dependencies) and on the other hand the characteristics of the computing infrastructure, and which will provide in output a file describing the infrastructure scheduling (switching on and off nodes) as well as the task scheduling (placement of a task on a node at a given time).

Students will be provided with :

- Documentation describing input and output formats and performance evaluation criteria.
- Examples of graphs and infrastructures
- A REST API (calculation function that can be called directly from the Internet) to evaluate the quality of the proposed solution.
- Access to distributed computing resources

Quarter number
ST7

Prerequisites (in terms of CS courses)
First year courses:

- SG1 common course "Systèmes d'Information et Programmation" (1CC1000)
- ST2 common course "Algorithmique et complexité" (1CC2000)

Courses of the ST:

- ST7 common course "Optimisation" (2CC3000)
- ST7 specific course "Méthodes et algorithmes parallèles pour l’optimisation" (2SC7610)

Others prerequisites:

- Parts of common course "CIP - Convergence, Intégration et Probabilités" (1SL1000)
- Parts of common course "EDP - Equations aux dérivées partielles" (1SL1500)
- Knowledge of linear algebra will also be needed

Syllabus

Main steps of the study:

- Complement of courses in hardware architecture of computer systems, including energy aspects.
- Formalization of the problem and the cost function to be optimized.
- Choice of a meta-heuristic optimization method adapted to the problem, examples: genetic algorithms, ant colonies, variable neighborhood method...
- Handling of remote computing resources (in cloud or supercomputer on which ANEO has access).
- Design of an algorithm parallel to the chosen optimization method, capable of scaling in terms of size or complexity of the task graph processed.
- Implementation of a parallel Python code supporting the planned scalability.
- Execution of the parallel resolution code on real data sets provided by ANEO, and within the limits of the computational resources allocated to the study.
- Analysis of the quality of the results of the resolution code, the performance of the resolution calculations performed (calculation speed, scalability), and the associated cost from an industrial exploitation perspective.
- The study will end with a report and an oral presentation aimed at evaluating the overall relevance of the solution found and tested, and the management of the quota of calculation resources that will have taken place during the project.

Rmk: The different groups of students will be confronted with different hypotheses on the targeted computing platforms, leading to equally different choices and implementations of optimization methods.

Class components (lecture, labs, etc.)

Part 1 (40HEE) :

- Steps 1 to 4: course additions, formalization of the problem, choice of an optimization method, and handling of calculation resources.
- Steps 5 and 6: first functional parallel implementation of the solving algorithm, small-scale tests.

- Intermediate report and presentation of progress and the work planned in part 2

Part 2 - *final sprint* (40HEE):

- Steps 7 and 8: execution of the resolution algorithm on intensive computing resources, and evaluation of the results obtained and the performance measured.
- Winding up in part 5 for the improvement of the resolution algorithm and its parallel implementation.

- Final report and full oral presentation

Grading

This project will be evaluated by a midterm talk at the end of part 1 (40HEE), and by a final talk at the end of part 2 (*final sprint* 40HEE). Talks will be done by the entire team, but will lead to individual marks in case of strongly heterogeneous teams.Each talk evaluation will consider the overall quality of the talk, of the slides and of the progress summary. Each talk mark will be 50% of the total mark.

Resources

Teaching staff:

- **A. Rimmel** (CentraleSupélec & LISN)
- **W. Kirschenmann** (ANEO)

Workplace and computing resources:

- Students will work at CentraleSupelec, in a classroom with electrical outlets and reliable wifi Internet access.
- Students will use their laptops to access remote computing resources (cloud or supercomputer on which ANEO has access).
- Final oral exam will take place at CentraleSupelec the last afternoon of the project.

Learning outcomes covered on the course

At the end of this project, students will be able to:

- **Learning Outcome 0 (AA0):** to identify the CPU consuming parts in an optimization chain, and to model the energy cost of a distributed calculation,
- **Learning Outcome 1 (AA1):** to identify optimization methods adapted to the minimization of the execution time of a task graph, and adapted to a large scale parallelization,
- **Learning Outcome 2 (AA2):** to design a parallel algorithm supporting scaling, to implement it and to develop its code on a distributed architecture,

- **Learning Outcome 3 (AA3):** to deploy intensive simulations on remote computing resources
- **Learning Outcome 4 (AA4):** to identify the limits of the study according to the available computational resources
- **Learning Outcome 5 (AA5):** to manage a quota of calculation resources during an intensive calculation campaign

Description of the skills acquired at the end of the course

- **C4:** Have a sense of value creation for his company and his customers
- **C7:** Know how to convince
- C8: Lead a project, a team