Dependability library for iterative applications applied to 4D FWI

Carla Santana*[1], Marcos Bandeira[1], Danyelle Da Silva[1], Edwin Duarte[1], Idalmis Milian Sardina[1], Ramon C. F. Araújo[1], Hervé Chauris[2], Claude Tadonki[2], Samuel Xavier-de-Souza[1] ([1]UFRN-Brazil, [2]Mines Paris-PSL- France)

Abstract

Geophysical applications, such as full-waveform inversion (FWI), are usually tackled with high-performance computing due to the high demand for computational resources. An interruption during the application's execution may significantly impact it, as it will take several days or weeks to recover the lost computation. Using suitable fault tolerance techniques can mitigate these consequences. In the 4D seismic case, these fault tolerance techniques are even more significant since performing FWI on two or more datasets, i.e., the baseline and all the monitor surveys, is necessary.

DeLIA (Dependability Library for Iterative Applications) is a fault tolerance library developed for iterative applications with data synchronization between all processes, obtaining a globally consistent state after each iteration. FWI has this behavior and is used to validate the library. The library offers API functions at their abstraction level to C, C++, and Julia applications. DeLIA provides checkpointing and rollback, replication, and interruption detection.

DeLIA has implemented the detection of possible interruption with the identification of termination signals and the method of heartbeat monitoring (HM). The termination signals notify a process to terminate at some moment. Some supercomputers and cloud systems use them to notify a job that it will be interrupted for some reason, for example, when the time limit is achieved or the resources from one preemptive instance cloud will be reclaimed. Moreover, in HM, nodes send a message periodically to a leader node. If the leader node does not receive the heartbeat messages from a given node in a specific interval, it assumes that such node failed and notifies the other nodes.

The checkpointing and rollback allow saving the state of the application or the process, and if the execution stops for some reason, it can recover the last state saved. The data replication provides reliability, and in DeLIA, it happens between the process; each one sends its state data to the neighbor process; when the process makes the checkpointing, it will save its data and the neighbor's data.

A primary challenge is applying fault tolerance techniques that do not add significant overhead and can be flexible to the applications' specificities. We developed DeLIA with its configuration file and the possibility for programmers to choose how to save their data and which data to save. The input file is in JSON format, and the user can enter information such as checkpointing intervals.

DeLIA was already validated in 3D FWI implemented in C++, and the library presented low overhead and adequate data recovery. In the current state of the evolution of the library, we are applying DeLIA in 4D FWI implemented in Julia to analyze the adaptability of the library in this new configuration.