11th. World Congress on Computational Mechanics (WCCM XI)
5th. European Conference on Computational Mechanics (ECCM V)
6th. European Conference on Computational Fluid Dynamics (ECFD VI)
July 20 - 25, 2014, Barcelona, Spain

USING HPC SOFTWARE FRAMEWORKS FOR DEVELOPING BSIT: A GEOPHYSICAL IMAGING TOOL

Mauricio Hanzich*, Juan Esteban Rodriguez*, Natalia Gutierrez*, Josep de la Puente* and José María Cela*

* Barcelona Supercomputing Center - Centro Nacional de Supercomputación
Gran Capità 2-4, Nexus I, 08034 Barcelona, Spain
e-mail: mauricio.hanzich@bsc.es, web page: http://www.bsc.es

Key words: *High Performance Computing, Software Frameworks, Efficiency, Reliability, Multi-platform, Multi-rheology*

BSIT (Barcelona Seismic Imaging Tools)[1] is an in-house multi-purpose set of frameworks focused on solving explicit and implicit scientific simulations using Finite Difference Methods as numerical scheme. Its framework was designed from scratch and was originally applied to solve geophysical problems such as Full Waveform Inversion [1] in an efficient and parallel manner. Moreover, BSIT platform is felixble enough to accomodate other kind of simulations, whenever based on a PDE.

Our initial aim while designing BSIT was twofold: to cope with the oil&gas goals in seismic software while maximizing the parallel efficiency of the produced systems. For those reasons, BSIT was designed for using structured meshes to discretize the problem domains, as these are better suited for optimization in modern accelerator-based architectures, that in turn provides the better hardware efficiency today.

To succeed in such challenge, BSIT frameworks needed also to have a modular design in order to ease development cycles, portability, reusability and future extensions of the frameworks. Figure 1 shows the main components of BSIT platform.

The *Kernel Framework* provides the means for developing a simulation solution for a single case, such as the migration of a single seismic shot [2]. The developer only needs to define its problem (e.g. code the equations) and the framework will tackle the interaction between computation, communication and input/output in order to overlap these stages as much as possible. The main aim of the Kernel Framework is to produce the most efficient code for solving a specific problem using a defined hardware architecture. Notice, that as only the equations must be coded the cost of porting any solution to a new architecture is reduced considerably.

[1] www.bsc.es/bsit

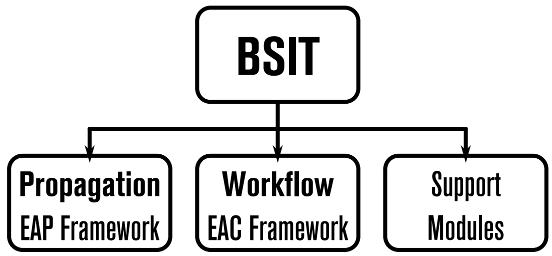

Figure 1: BSIT main structure

The *Workflow Framework* provides the means to define a workflow among different *kernels* to build some complex flow of tasks conforming the simulation process (e.g. Montecarlo simulations or optimization problems). For example, the parallel processing of thousands of seismic shots modelling and data gathering [3]. One of the main aims of the Workflow Framework is to provide the kernels with a robust environment for producing their results. For this the framework includes many resiliency characteristics such as: fault detection and recovery, re-launch failed kernels, status checkpointing, whole simulation restart and data corruption checking.

Finally, the *Support Modules* provides a set of backing modules to the previous frameworks. It contains modules for: data format management, data transformation (e.g. interpolation, cropping, filtering ,etc.) and common elements for message logging, parameters management or error handling. One important set of modules into the support is devoted to optimization. Such modules provides the capacity for run-time definition of efficiency-related parameters, just-in-time compilation of the most time consuming functions and some monitoring tools for tracing and timing the kernels and workflows.

Regarding results, using BSIT our group increase the kernel development loop from 1 kernel per year to more than 20, while increasing the overall kernel efficiency. Also, the workflows build in the same period went from 1 to 5. Hence, the usage of a software platform designed from scratch to provide flexibility, code reusage, portability and parallel performance greatly impact the development process of highly efficient distributed code.

REFERENCES

[1] J. Virieux and S. Operto. "An overview of full-waveform inversion in exploration geophysics."Geophysics74.6 (2009): WCC1-WCC26.

[2] M. Araya-Polo, J. Cabezas, M. Hanzich, M. Pericas, E. Morancho, I. Gelado, M. Shafiq, F. Rubio, J.M. Cela, E. Ayguad, N. Navarro and M. Valero. "Assessing Accelerator-based HPC Reverse Time Migration". Transactions on Parallel and Distributed Systems, Special Issue on Accelerators 22(1), 147-162 (2011).

[3] J.E. Rodriguez and M. Hanzich. "Supporting Massive Parallelism in Seismic Processing". 75th EAGE Conference & Exhibition. June 2013.