HPC16

Encapsulating HPC System Level Complexities via Software Framework

S. Das* (ExxonMobil Technical Computing Comapny)

SUMMARY

Leveraging petascale systems effectively and efficiently is not a trivial challenge. A good harmony between system architecture, algorithms, and application architecture is the key to success. In addition, when complex algorithms build up a workflow it becomes ever more challenging to get optimal execution. Successful business application requires not only the innovation in algorithm space but also efficient execution leveraging HPC environment. Positioning these complex technologies in a robust, scalable commercial usable form makes the business successful.

Here is a case study where different technologies are integrated in a software framework leveraging HPC environment. Also, leveraging a HPC system introduces additional complexities of achieving high level of parallelism. In the HPC group, as we started commercializing these technologies, we added them to a software framework to reduce end users complexities, improve technology uptake and effective HPC resource utilization.

Leveraging petascale systems effectively and efficiently is not a trivial challenge. A good harmony between system architecture, algorithms, and application architecture is the key to success. In addition, when complex algorithms build up a workflow it becomes ever more challenging to get optimal execution. Successful business application requires not only the innovation in algorithm space but also efficient execution leveraging HPC environment. Positioning these complex technologies in a robust, scalable commercial usable form makes the business successful.

Here is a case study where different technologies are integrated in a software framework leveraging HPC environment. Also, leveraging a HPC system introduces additional complexities of achieving high level of parallelism. In the HPC group, as we started commercializing these technologies, we added them to a software framework to reduce end users complexities, improve technology uptake and effective HPC resource utilization.

Hence, a framework has evolved to address five major common components:

- a) User facing canvas that addresses workflows, parameter validation
- b) Job submission, monitoring, project management
- c) Distribution and collection of data
- d) Support for different data formats and data ordering
- e) Parallelization strategy and job decomposition

a) Keeping the focus on simplicity, this framework enables a workflow canvas that presents complex pre-determined business workflows, which hides optional parameters and pre-populates default parameters as much as feasible. However, it also enables flexibility for the advanced users to delve deeper and customize the jobs to fit the business needs. This system provides capability to manage many permutations of an execution workflow enabling scenario testing. This approach has improved technology uptake, new technology rollout and made training much easier.

b) In a large scale production system, underlying infrastructure (queuing) is complex and ever changing to meet business needs. This component of the framework hides all aspects of job submission complexities from the users. In addition, users have the flexibility to add or remove jobs from the system while an execution of a workflow is in progress. Not only that, this structured approach enables effective data management and it provides real time job monitoring capability that users use to track project progress and predict project execution timeline.

c) Due to the distributed nature of the system, many of these workflows require collective I/O operations on large data volume (TBs). The ability to provide these types of operations at the framework level improves system scalability and operation. The reusability aspect of these components enables future workflow development faster. Effective strategies for data layouts, alignment, indexing makes the application leverage the high performance computing resource more effectively. Consideration for staging input/output data on node to reduce file system traffic to improve performance of latency proved to be beneficial.

d) Users are continuously faced with the challenge of going back and forth between multiple systems where the data format and assumptions are different. This poses a huge challenge for the users and the system. Multiple copies of the dataset (in different format/order) pose a challenge to disk space utilization. On-going work is in progress to resolve data exchange challenge.

e) Another major aspect is that some of these technologies were originally implemented as a single processor implementation. This framework allows the invocation of the technologies in parallel mode via a parallel launcher. Complexities of data decomposition and distribution are handled by the framework with guidance from the individual tools. Poorly decomposed and unbalanced problems waste compute and I/O resources and most importantly users' time. Optimal decomposition of input is done automatically at runtime with minimum decisions required from the users. Enabling dynamic

load balancing improves scalability and improves hardware resource utilization. In summary, this has a huge positive impact on performance gain and cycle time reduction.

Now, we can consider more complex algorithms which are parallel in nature and implemented in parallel. Depending on the complexity and the footprint of the application, application architecture can become quite complex. In a collaborative research environment, having a robust application architecture that hides the complexity of underlying data structure and MPI directives will be very beneficial for the researchers. If the underlying architecture can be abstracted, research can progress while migration to evolving HPC technologies can progress simultaneously, an idealistic scenario.

Overall, this approach of a software framework improves supportability and future expansion. A reusable codebase simplifies the development process by maintaining focus on exposing new geophysics capabilities. Framework enhancement benefits all technologies and broad pool of developers can support a larger set of technologies. Also, the researchers developing new ideas do not need to be aware of all the nuances of the complex infrastructure. This method accelerates research technology delivery and user uptake. With the unexpected success of this software framework in the business setting, users are driving growth – a perfect success story for a software framework!