## Life cycle management of big data for extreme-scale simulation

Kenji Ono, Advanced Institute for Computational Science, RIKEN

Toward next generation supercomputing, it is obviously essential to develop a computer system to realize an extreme-scale computing environment. Further, it is of great importance to develop high performance applications and its utilization, which contribute to society through scientific discovery.

This position paper describes that peripheral technologies and data management of simulation life cycle, which are essential for effective utilization of large-scale parallel simulations.

In order to contribute to society using high performance computing, the development of a runtime and support environment for applications is required in addition to enhancing a code performance. For instance, a large-scale parallel fluid simulation, which is essential technology for the design of airplane or automobile, has peripheral processes such as generation of large-scale analysis model, project management, parameter generation for grid search, management and execution of job task, visualization, cooperation with a data base, and so on. These peripheral processes are expected to maximize the effect of simulations.

A typical sequence of physical simulations is composed from model generation, computation, post processing (data processing and visualization), and archive for re-use. These processes form a pipeline of computation through the simulation, then the data flows the processes, eventually, gives information that helps understanding phenomena and making decision. Many files are generated and in use in the execution of a large-scale parallel simulation. In extreme-scale simulation, the increase of the number of trial and its scale brings the difficulty. In terms of the efficiency, it is demanded to manage the resource required to the simulation and to control the workflow to automate process through the entire process of the simulation. The issues are how to design an execution environment of simulations and a way of the data management, and how to provide an efficient simulation environment to the end-users.

To help the efficient and functional execution of simulations, a unified simulation runtime environment was proposed. This execution environment combines sub-systems that have different functions, and each sub-system interchanges data between them via API. This implementation makes sub-systems independent so that we can use the existed software or can replace the sub-system by other products.

The unified runtime environment will provide a data management function for the entire process of simulations. The policy of the management and its design become a key for the efficiency. We introduced two categories to manage the resource used in simulations. One is 'case' as a unit of execution of a simulation, and the other is 'project' as a set of cases, and they are defined by a case information file (CIF) and a project information file (PIF), respectively. PIF describes all information including directory path and all files related the project. CIF saves parameters and file names related to a specific execution of the simulation. These files are written by the XML format. In addition to the resource management, PIF and CIF enable us to provide provenance function by adding annotations such as a time stamp, a specification of used machine, and relationship between files.

In the proposed environment, data flows pipelined processes. A workflow sub-system controls the data flow and is described by the conventional shell and perl script to be operated on various target machines. To realize the automation in the runtime environment, each sub-system must provide API to interchange information. This workflow also provides functions to submit a batch job script and to retrieve the computed data from remote supercomputers after the job ends. In particular, this mechanism is quite useful in a grid search that demands many jobs for different parameters.

A distributed file management including I/O performance should be much more paid attention for large-scale parallel computation. Because data transfer between the simulation and the post processing is time consuming part to read and write files in the whole simulation process. A distributed file management can be described by the viewpoint of the data structure and decomposition method, which covers some application region.

We designed and constructed the distributed file management libraries for the Cartesian and a hierarchical Cartesian data structure. These libraries provide functions of file management and the restart. The restart function includes restarting with refinement, with staging, and restarting from different number of processes besides normal restarting. These restart scenarios greatly help large-scale simulation. Data compression and streaming technique will be also introduced in near future.

Post processing stage yields derived data, meta data like graphs, images and movies. The project information file manages these abstracted data.

All files used in the simulation become basic data of verification and validation. Therefore, accumulating examples of simulation, the access of a public database will lead enhancing the utilization of simulators. To do so, the mechanism of automatic data registration plays an important role. In the future, the accumulated data will be effectively exhibited by introducing a curation service.