In-situ Big Data Analysis at Scale for Extreme Scale Systems

Alok Choudhary
Henry and Isabel Dever Professor
Electrical Engineering and Computer Science
Northwestern University.
http://www.eecs.northwestern.edu/~choudhar

The challenges of extreme scale computing systems exist across multiple dimensions including architecture, energy constraints, memory scaling, limited I/O, scalability of software and applications. These constraints, and the need for faster scientific discovery make it is clear that larger the simulations using extreme-scale systems, (1) the greater the need for effective data analysis and derivation of insights, (2) at a faster pace, and (3) within the constrains of limited (relative) storage space, deeper and complex memory hierarchies, minimization of data movements (particularly external to the systems) due to energy, I/O and other constraints. The traditional model of *store raw and/or derived data and analyze later* is so-cost prohibitive that it is infeasible for most cases. Furthermore, the "top-down" approaches, that continuously involve human in the loop for analyzing data, become less effective at this scale due to the size and complexity of data, and the underlying design assumptions of data being available based on the current and previous modes of analysis. For in-situ analysis, the designs of analysis algorithms and software if by simply extending the assumptions made based on the off-line model may not work, and therefore, rethinking and redesign of analysis algorithms, runtime and software is needed. For example, an assumption typically made in offline data analysis is that an algorithm can go back and forth in time to look at or process data. Such an assumption will no longer be valid for in-situ analysis, or at least will be constrained to within a much smaller temporal window. Thus a design and development rethinking (or "reformulation") is necessary for analysis algorithms and software.

In order to keep pace with the ever increasing computational parallelism demands by large-scale simulations, the development of analysis algorithms must address the following design criteria and dimensions among others: (1) Can the algorithms and software design be hierarchical and component-based, where lot of computations is done within the nodes (closest to data, minimizing data-movement, based on local information, *sharing processor/memory*), some more in intermediate nodes (e.g., staging, with additional global information) and the rest on analysis nodes and systems; (2) Can general components be both function and service oriented so that the runtime system and applications can use them flexibly and optimally; (3) Can they be designed with the ability to store intermediate state and partial computations with portable and consistent formats using SSD/NVRAM; (4) Can they be guided by basic learning algorithms to understand local distributions, constrain computations (e.g., approximations); and (5) Can they be adaptable and parameterized so that they fulfill the need for memory-size tradeoffs, computation power tradeoffs and performance (time-to-result) tradeoffs (compute-heavy/memory-heavy tradeoffs). Furthermore, they are customizable to the needs of simulation similarities (and data types). A thin service and portable runtime layer that understands basic kernels of analysis, statistics and learning algorithms (to reduce, transform data and results and providing feedback) would be important.

Design approaches must be driven by rethinking and reformulation within the constrains of in-situ analysis requirements at multiple levels. This is particularly important for big data analytics because the most existing techniques developed, validated and optimized on small data sets may not be scalable nor may they be suitable for in-situ analytics on the emerging extreme scale computing systems. Traditionally these algorithms have been developed as monolithic components, and mostly without considering the requirements and constraints described above. Another key component will be to incorporate a co-design approach to development of scalable algorithms and software by taking the full advantage of new architecture rather than simply considering and scaling existing (sequential) techniques.

The following issues must be considered in the design of algorithms and software: (1) what parts of the computation can be done close to the data within the nodes, while it is still in memory; (2) what analysis components can (and should) be performed in staging and analysis nodes within a system; (3) is approximation possible and to what extent. This would depend on the type of algorithms and needs for certain applications. Thus multiple options, that can be dynamically used, or provided as service would need to be considered and developed; (4) Given the typical spatio-temporal nature of the data, what type of derived distributions and statistics can be kept locally in order to both accelerate computations and meet energy constraints in subsequent iterations and phases; (5) What type of self-describing formats can be embedded within the data for local computations so that they can be kept persistent within SSDs if needed and understood at staging and analysis nodes, thereby providing portability and flexibility; (6) What type of service oriented functions can be developed that can be scheduled locally by the underlying runtime system based on the dynamic requirements of applications; This approach has the advantage of being flexible, where components of analytics algorithms can be easily changed without the application having to know about it.