

FOX: A Fault-oblivious Extreme-scale Execution Environment (fox.xstack.org)



Exascale computing systems will provide a thousand-fold increase in parallelism and a proportional increase in failure rate relative to today's machines. Systems software for exascale machines must provide the infrastructure to support existing applications while simultaneously enabling efficient execution of new programming models that naturally express dynamic, adaptive, irregular computation; coupled simulations; and massive data analysis in a highly unreliable hardware environment with billions of threads of execution. Further, these systems must be designed with failure in mind. FOX is a new system for the exascale which will support distributed data objects as a first class object in the operating system itself. This memory-based data store will be named and accessed as part of the file system name space of the application. We can build many types of objects with this data store, including data-driven work queues, which will in turn support applications with inherent resilience.

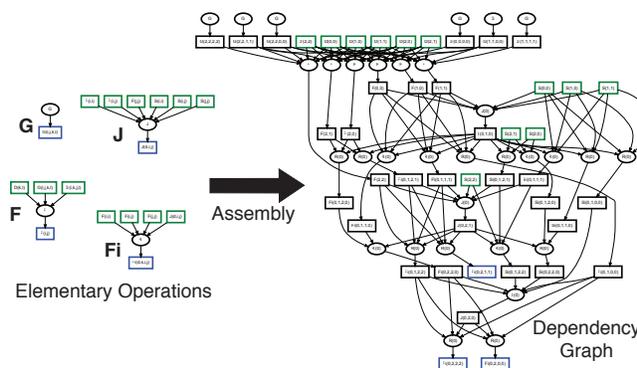


Figure 1. Dependency graph for a portion of the Hartree-Fock procedure.

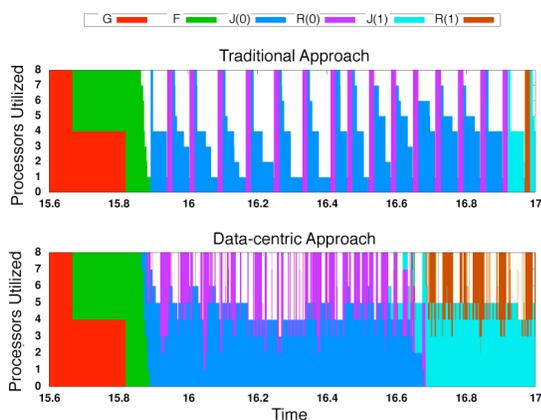


Figure 2. A comparison of processor utilization for traditional (top) and data-centric (bottom) approaches using simulated timings for a portion of the Hartree-Fock procedure.

Work queues are a familiar concept in many areas of computing; we will apply the work queue concept to applications. Figure 1 shows a graphical data flow description of a molecular modeling application, which can be executed using a work queue approach.

We have simulated an application modified to use this model and, in simulation, parallel performance improves. Figure 2 shows the comparative performance, with the traditional SPMD-style program on the left, and the data-driven program on the right. Utilization is higher, cycle times for an iteration are shorter, and work on the next cycle can begin well before the current cycle completes.

Progress to date

SST/Macro

The team recently met (physically and virtually) at Sandia National Labs for a “deep dive” into the SST/Macro simulator. We learned how to measure and characterize applications behavior as well as “skeletonize” an application so it can be moved to the simulator.

To make SST/Macro more widely available, we have packaged it as a virtual machine image for VMWare. This allows users who might not want to build the software to simply download and boot the VMWare image. Use of a virtual machine eliminates OS issues.

SST/Macro has been extended to support a mail box capability, as well as active messages. Fault injection support is in development.

Applications

LLNL is providing a graph processing application for study. This is a challenging application as it has only run well on shared-memory machines, and features very small messages. Getting it to run well on MPI requires modifications such as message aggregation that are not an optimal fit to the application. We will use SST/Macro to study what kind of message-passing interface might best suit the application, and the implications of such an interface for both hardware and software architecture.

PNNL has undertaken characterization of key modules of NWChem from a fault tolerance perspective. This process will be used to identify critical application phases, kernels from which will be studied using SST/Macro. Some early results have been submitted for publication in Euro-Par 2011.

PNNL has begun preliminary investigation into evaluating energy usage. The study of candidate applications will be used to develop kernels for SST/Macro.

Runtimes

Sandia has been experimenting with a specialization of Plan 9 to support active messages. The initial changes to Plan 9 now support one-sided active messages with sub-10-microsecond latency. The send overhead is just 1.165 microseconds. We believe we can improve on this time, such that we can have a kernel-based active message support rivaling the performance of MPI. Vita Nuova has focused on developing a new service to support messaging and sharing of large chunks of data, using new system interfaces on Plan 9, including experimenting with different representations of RDMA on BG/P. PNNL is developing fault tolerant data stores. The work on efficient fault tolerance data stores for read-only matrices under generalized cartesian distributions has been accepted for publication in Computing Frontiers 2011. PNNL and OSU are investigating fault tolerant dynamic load balancing schemes through additions to work stealing. This involves minimizing the space and communication overhead in identifying communication progress and tasks executed, and minimizing work lost due to failures.

Operating Systems

The Osprey O/S being developed by the Network Systems group at Bell Labs, which builds on the experience gained from applying the Plan 9 O/S on systems ranging from embedded network components to BG/P.

The second is a Library OS from IBM which is also coming up on test hardware. This new OS builds on work done for the DARPA HPCS program.