Approaches to and Tools for Project Performance Tracking and Analysis

SUPER
INSTITUTE FOR SUSTAINED PERFORMANCE, ENERGY, AND RESILIENCE

Patrick Worley, Philip Roth
Oak Ridge National Laboratory

Kevin Huck, Scott Biersdorff, Chee Wai Lee, Allen Malony, Suzanne Millstein, Sameer Shende, Wyatt Spear
University of Oregon

SUPER is providing performance infrastructure ranging from advice on best practices to sophisticated tools for auto-tuning, improving resilience, and minimizing energy consumption. One important best practice is end-to-end performance tracking and analysis of production work loads. This does not require sophisticated tools, just (project-wide) discipline. SUPER can advise on methodology and tools appropriate for each DOE computing center. SUPER is also developing low overhead instrumentation tools that provide additional insight even on heterogenous nodes, and data archive and analysis tools that can merge and correlate data from different sources.

Case Study

Code: Community Earth System Model (CESM)

Platform: Cray XK6 at the Oak Ridge Leadership Computing Facility

Issue: Performance variability leading to failed runs (time exceeded before target simulation time reached)

Response: Collect data from all production jobs, to determine nature of variability and quantify impact.

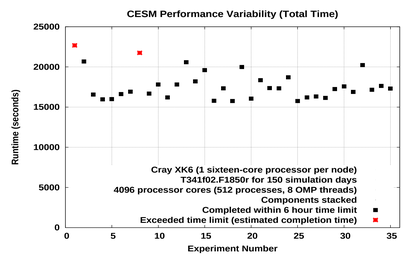
Approach: Modify production job scripts.

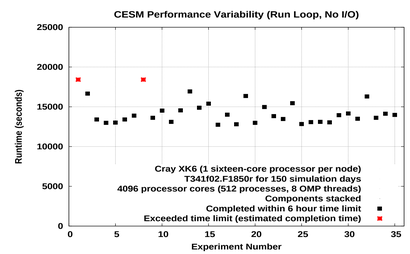
Data Collection

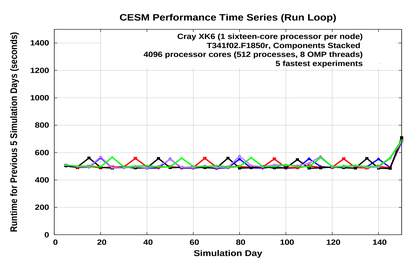
1. Data collected per user (by login name), per experiment case (user-defined), and per job (timestamp for job start), in a project directory (so not swept).
2. Before/as job starts, collect
   a) problem and code provenance
   b) current system topology ('xtdbproc')
   c) current system workload ('qstat', 'showq', 'xtnodestat')
   d) time spent in queue ('qstat') and in preprocessing steps
3. While job running, collect
   a) process to node/core mapping (set MPICH_CPUMASK_DISPLAY to 1)
   b) periodic updates to system workload (background job calling 'showq', 'xtnodestat')
   c) periodic measurements of job progress (via both background job monitoring of CESM output and native CESM instrumentation logic)
4. As/after job completes, collect
   a) time spent in postprocessing steps
   b) standard CESM-specific performance summary

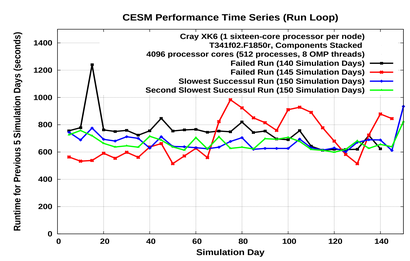
Example Experiment Case Analysis

- 4096 processor cores, 6 hour wall clock request
- Target of 150 simulation days (benchmarks indicated could compute 180)
- 35 jobs (between May 15 and Jun 29, 2012), two of which failed

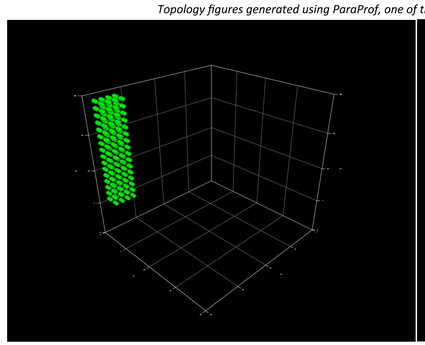




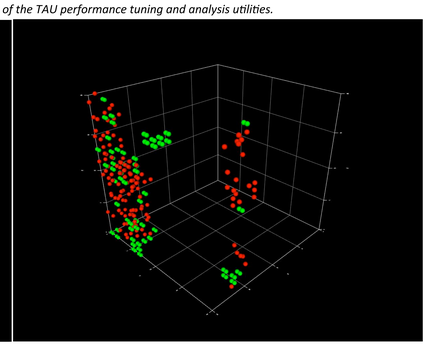




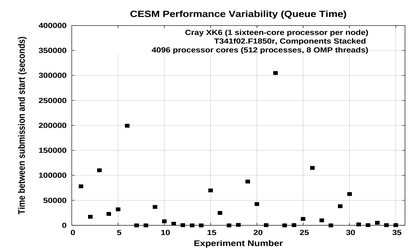
Topology figures generated using ParaProf, one of the TAU performance tuning and analysis utilities.



Compute node assignment for job with minimum execution time



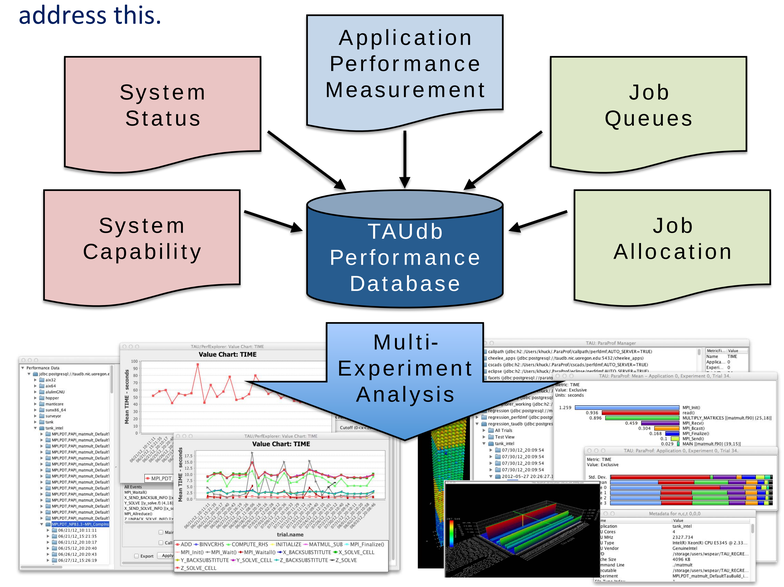
Compute node assignment for successful job with maximum execution time.



- Short queue time for many job submissions (25% under 1 minute), but over 40% were in the queue over 4 hours and 4 jobs were in the queue over a day (max. of 3 days).
- Decrease of target from 180 to 150 simulation days to decrease failure frequency, as well as rerunning part of failed jobs, increases total number of jobs. This results in more time spent in the queue, decreasing project throughput.

Conclusion

- Performance variability is real and has an impact on project performance.
- Variability is strongly correlated with node allocation, is not related only to I/O, and can affect even relatively small jobs.
- Data sufficient to identify problem and possible sources was easily collected. Such data provide a starting point in working with SUPER.
- Performance data management and analysis is time consuming. We are currently importing data into the TAUdb performance database to address this.



Support for this work was provided through the Scientific Discovery through Advanced Computing (SciDAC) program funded by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research. This work also used resources of the Oak Ridge Leadership Computing Facility located in the Oak Ridge National Laboratory, which is supported by the Office of Science of the Department of Energy under Contract DE-AC05-00OR22725.