A Performance/Resilience/Power Co-design Tool for Extreme-scale High-Performance Computing

Christian Engelmann[*] and Thomas Naughton
Oak Ridge National Laboratory
[*] *(865) 574-3132 / engelmannc@computer.org*

Background and Motivation

With the deployment of 10-20 PFlop/s supercomputers and the exascale roadmap targeting 100, 300, and eventually 1,000 PFlop/s over the next decade, the trend in supercomputer architecture goes clearly in only one direction. Systems will dramatically scale up in size, i.e., in node and core counts [Kog08]. By 2020, an exascale system may have 100,000-1,000,000 nodes with 1,000-10,000 cores per node. This poses several challenges related to power consumption, performance, resilience, productivity, programmability, data movement, and data management.

The expected substantial growth in concurrency causes application scalability issues due to sequential parts, synchronizing communication, and other bottlenecks. High-performance computing (HPC) hardware/software co-design is crucial to enable extreme-scale computing by closing the gap between the peak capabilities of the hardware and the performance realized by applications (application-architecture performance gap). Investigating the performance of applications at scale on future architectures and the performance impact of different architecture choices is an important component of HPC hardware/software co-design. Without having access to future architectures, especially at scale, simulation approaches provide an alternative for estimating application performance on potential architecture choices. As highly accurate simulations are extremely slow and less scalable, different solution paths exist to trade off simulation accuracy to gain performance and scalability [Böh11, Gir00, Per10, Rod11, Zhe04].

Resilience, i.e., providing efficiency and correctness in the presence of faults, is one of the most important exascale computer science challenges as systems scale up in component count and component reliability decreases (7nm technology with near-threshold voltage operation by 2020) [Dal12, Kau12]. A number of advanced resilience technologies exist, including: checkpoint/restart-specific file/storage systems [Li10], incremental/differential checkpointing [Wan10], message logging for uncoordinated checkpointing [Lem04], fault tolerant message passing interface (FT-MPI) [Fag05], containment domains [Sul11], algorithm-based fault tolerance (ABFT) [Dav11, Du12], rejuvenation [Nak10], proactive migration [Wan12], and redundancy [Eng11]. However, there are currently no tools, methods, and metrics to compare them fairly.

Performance, resilience and power consumption are key HPC system design factors that are highly interdependent. To enable extreme-scale computing it is essential to perform HPC hardware/software co-design that identifies the cost/benefit trade-off between these design factors for potential future architecture choices.

Approach

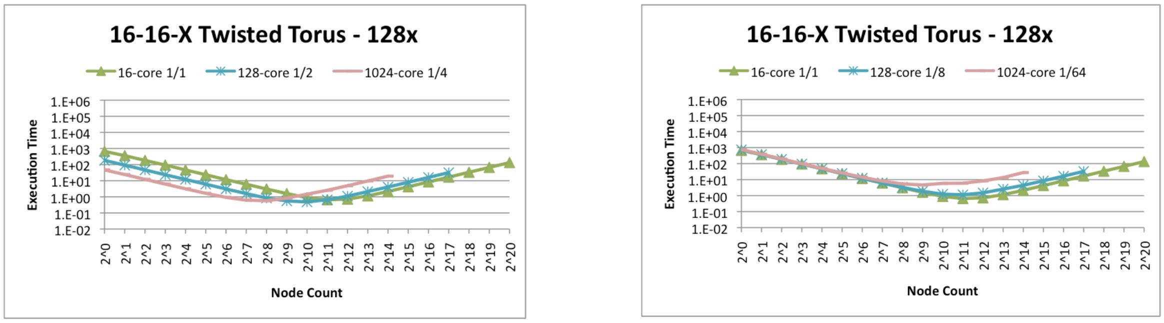
The proposed research and development aims at developing an HPC hardware/software co-design toolkit for evaluating the resilience/power/performance cost/benefit trade-off of future architecture choices. The approach focuses on extending a simulation-based performance investigation toolkit with advanced resilience and power modeling and simulation features, such as (i) fault injection mechanisms, (ii) fault propagation, isolation, and detection models, (i) fault avoidance, masking, and recovery simulation, and (iv) power consumption models.

Specifically, the planned effort targets features to: (1) permit the injection of different faults, errors, and failures into the simulation, (2) model various propagation, isolation, and detection properties of the simulated system, (3) support a variety of avoidance, masking, and recovery strategies, (4) model the power consumption of the entire simulated system, and (5) study the performance, resilience, and power consumption impact with different parameter sets for (1), (2), (3), and (4) using standardized methods and metrics.

The proposed work is novel in several ways. The proposed HPC hardware/software co-design toolkit would be (a) the first fault injection toolkit for extreme-scale systems, (b) the only fault injection toolkit with parameterized resilience properties and strategies, and (c) the first holistic HPC co-design toolkit that considers architectural performance and resilience parameters to optimize application performance within a given power consumption budget.

Preliminary Accomplishments[1]

The Extreme-scale Simulator (xSim) [Jon11, Böh11, Eng10] is a recently developed performance investigation toolkit that permits running HPC applications in a controlled environment with millions of concurrent execution threads. It allows observing application performance in a simulated extreme-scale system for hardware/software co-design. Using a lightweight parallel discrete event simulation (PDES), xSim executes an application on a much smaller HPC system in an oversubscribed fashion with a virtual wall clock time, such that performance data can be extracted based on a processor and a network model with an appropriate simulation scalability/accuracy trade-off (Figure 1). xSim is designed like a traditional performance tool, as an interposition library that sits between the MPI application and the MPI layer, using the MPI performance tool interface. It currently holds the world record in extreme-scale simulation, running up to 134,217,728 (2^{27}) communicating MPI tasks, each with its own process context, using just a 960-core Linux-based cluster. Its ability to simulate the architectural properties of extreme-scale HPC systems at reasonable accuracy and to run real applications or corresponding application proxies or models makes it the prime candidate for a HPC hardware/software co-design toolkit to investigate the performance/resilience/power trade-off of future HPC architecture choices with different performance characteristics, resilience properties (vulnerabilities), resilience strategies (mitigation techniques), and power consumption envelopes.



(a) 16, 128, 1024 cores/node with 1, ½, ¼ core performance

(b) 16, 128, 1024 cores/node with 1, ⅛, $^{1}/_{64}$ core performance

Figure 1: Scaling a Monte Carlo Application on Different Multi-Core Architectures (at 128x Problem Size)

Research Plan[2]

This effort targets a simulation-based HPC hardware/software performance, resilience, and power consumption co-design toolkit enabling reproducible experiments to estimate the cost/benefit trade-off given certain system and application properties. The planned work encompasses computer science research and development applied to DOE's needs on the path to exascale computing.

The targeted ***fault injection mechanisms*** focus on job abort, process fault, detected data corruption, and silent data corruption (SDC). The proposed ***fault propagation, isolation, and detection models*** are part of the fault injection mechanism and provide the capability to investigate fault detection latency issues, domino effects, and simultaneous faults and recoveries. The planned ***fault avoidance, masking, and recovery simulation*** enables the evaluation of resilience mechanism efficiency and aims at the most likely candidates for future-generation systems: (a) coordinated checkpoint/restart using different checkpoint storage systems, (b) uncoordinated checkpoint/restart using different message logging algorithms and checkpoint storage systems, (c) MPI process fault tolerance combined with ABFT, and (d) ABFT detecting and correcting SDC. The processor, network, and storage system performance models will be extended with corresponding ***power consumption models***, by extrapolating power consumption from component utilization and additionally supporting the simulation of energy-saving mechanisms, such as voltage/frequency scaling and powering off subcomponents, as well as, resources with different energy efficiency, such CPUs vs. GPUs. The planned ***investigation of the performance, resilience and power consumption cost/benefit trade-off*** will mostly focus on application proxies, standard benchmarks, and fault tolerant applications.

Related Work

The planned work relies on our previous efforts in (1) software fault injection at the operating system level [Nau09], (2) soft error injection at the MPI layer [Fia12], (3) modeling the efficiency of (a) checkpoint/restart to/from node-local memory or solid state disk (SSD) storage [Li10], (b) full and incremental checkpoint/restart to shared parallel file systems [Wan10], (c) proactive fault tolerance utilizing process migration [Wan12], and (d) process-level redundancy at the MPI layer [Eng11]. The proposed effort further utilizes recent accomplishments by others in the area of resilience solutions, modeling, and simulation, as well as, in performance/power modeling and simulation. Of particular interest to this project is work in reliability modeling of extreme-scale systems [Dal06, Yan12], soft error quantification of hardware components [Hwa12], modeling the efficiency of ABFT [Dav11, Du12], CPU-GPU performance modeling [Ker10], and CPU-GPU power modeling [Hon10, Riv08].

[1] Research sponsored by the Office of Advanced Scientific Computing Research, U.S. Department of Energy.
[2] Research sponsored in part by the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory.
This manuscript has been authored by UT-Battelle, LLC, under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy.

References

[Böh11] S. Böhm and C. Engelmann. xSim: The Extreme-Scale Simulator. Intl. Conf. on High Performance Computing and Simulation (HPCS), pp. 280-286, 2011.

[Dal06] J. Daly. A Higher Order Estimate of the Optimum Checkpoint Interval for Restart Dumps. Future Generation Computer Systems (FGCS) 22:303-312, 2006.

[Dal12] J. Daly et al.. Inter-Agency Workshop on HPC Resilience at Extreme Scale. Workshop Report, 2012.

[Dav11] T. Davies, C. Karlsson, H. Liu, C. Ding, and Z. Chen. High Performance Linpack Benchmark: A Fault Tolerant Implementation without Checkpointing. 25th ACM Intl. Conf. on Supercomputing (ICS), 2011.

[Du12] P. Du, A. Bouteiller, G. Bosilca, T. Herault, and J. Dongarra. Algorithm-Based Fault Tolerance for Dense Matrix Factorization. 17th ACM SIGPLAN Symp. on Principles and Practice of Parallel Programming (PPOPP), pp. 225-234, 2012.

[Eng10] C. Engelmann and F. Lauer. Facilitating Co-Design for Extreme-Scale Systems Through Lightweight Simulation. 12th IEEE Intl. Conf. on Cluster Computing (Cluster): 1st Workshop on Application/Architecture Co-design for Extreme-scale Computing (AACEC), 2010.

[Eng11] C. Engelmann and S. Böhm. Redundant Execution of HPC Applications with MR-MPI. 10th IASTED Intl. Conf. on Parallel and Distributed Computing and Networks (PDCN), pp. 31-38, 2011.

[Fag05] G. Fagg, T. Angskun, G. Bosilca, J. Pjesivac-Grbovic, and J. Dongarra. Scalable Fault Tolerant MPI: Extending the Recovery Algorithm. 12th European Parallel Virtual Machine and Message Passing Interface Conf. (Euro PVM/MPI), pp. 67, 2005.

[Fia12] D. Fiala, F. Mueller, C. Engelmann, K. Ferreira, R. Brightwell, and R. Riesen. Detection and Correction of Silent Data Corruption for Large-Scale High-Performance Computing. 25th IEEE/ACM Intl. Conf. on High Performance Computing, Networking, Storage and Analysis (SC), 2012.

[Gir00] S. Girona, J. Labarta, R. Badia. Validation of DIMEMAS Communication Model for MPI Collective Operations. 7th European PVM/MPI Users‘ Group Meeting (EuroPVM/MPI), pp. 39-46, 2000.

[Hon10] S. Hong and H. Kim. An Integrated GPU Power and Performance Model. SIGARCH Comput. Archit. News, 38(3):280–289, 2010.

[Hwa12] A. Hwang, I. Stefanovici, and B. Schroeder. Cosmic Rays Don’t Strike Twice: Understanding the Nature of DRAM Errors and the Implications for System Design. SIGARCH Comput. Archit. News, 40(1):111-122, 2012.

[Jon11] I. Jones and C. Engelmann. Simulation of Large-Scale HPC Architectures. 40th Intl. Conf. on Parallel Processing (ICPP): 2nd International Workshop on Parallel Software Tools and Tool Infrastructures (PSTI), pp. 447-456, 2011.

[Kau12] H. Kaul, M. Anders, S. Hsu, A. Agarwal, R. Krishnamurthy, and S. Borkar: Near-threshold voltage (NTV) design: Opportunities and challenges. 49th Annual Design Automation Conf. (DAC), pp. 1153-1158, 2012.

[Ker10] A. Kerr, G. Diamos, and S. Yalamanchili. Modeling GPU-CPU Workloads and Systems. 3rd Workshop on General-Purpose Computation on Graphics Processing Units (GPGPU), pp. 31–42, 2010.

[Kog08] P. Kogge et al.. ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems, 2008.

[Lem04] P. Lemarinier, A. Bouteiller, T. Herault, G. Krawezik, and F. Cappello. Improved Message Logging Versus Improved Coordinated Checkpointing for Fault Tolerant MPI. 6th IEEE Intl. Conf. on Cluster Computing, pp. 115-124, 2004.

[Li10] M. Li, S. Vazhkudai, A. Butt, F. Meng, X. Ma, Y. Kim, C. Engelmann, and G. Shipman. Functional Partitioning to Optimize End-to-End Performance on Many-Core Architectures. 23rd IEEE/ACM Intl. Conf. on High Performance Computing, Networking, Storage and Analysis (SC), 2010.

[Nak10] N. Naksinehaboon, N. Taerat, C. Leangsuksun, C. Chandler, and S. Scott. Benefits of Software Rejuvenation on HPC Systems. 8th IEEE Intl. Symp. on Parallel and Distributed Processing with Applications (ISPA), pp. 499-506, 2010.

[Nau09] T. Naughton, W. Bland, G. Vallée, C. Engelmann, and S. Scott. Fault Injection Framework for System Resilience Evaluation - Fake Faults for Finding Future Failures. 18th Intl. Symp. on High Performance Distributed Computing (HPDC): 2nd Workshop on Resiliency in High Performance Computing (Resilience), pp. 23-28, 2009.

[Per10] K. Perumalla. μπ: A Highly Scalable and Transparent System for Simulating MPI Programs. 3rd Intl. ICST Conf. on Simulation Tools and Techniques (SIMUTools), 2010.

[Riv08] S. Rivoire, P. Ranganathan, and C. Kozyrakis. A Comparison of High-level Full-system Power Models. Conf. on Power-aware Computing and Systems (HotPower), 2008.

[Rod11] A. Rodrigues, K. Hemmert, B. Barrett, C. Kersey, R. Oldfield, M. Weston, R. Riesen, J. Cook, P. Rosenfeld, E. Coper-Balis, B. Jacob, The Structural Simulation Toolkit. SIGMETRICS Perform. Eval. Rev., 38:37–42, 2011.

[Sul11] M. Sullivan, D.H. Yoon, and M. Erez. Containment Domains: A Full-System Approach to Computational Resiliency. Technical report TR-LPH-2011-001, The University of Texas at Austin, 2011.

[Wan10] C. Wang, F. Mueller, C. Engelmann, and S. Scott. Hybrid Checkpointing for MPI Jobs in HPC Environments. 16th IEEE Intl. Conf. on Parallel and Distributed Systems (IPDPS), pp. 524-533, 2010.

[Wan12] C. Wang, F. Mueller, C. Engelmann, and S. Scott. Proactive Process-Level Live Migration and Back Migration in HPC Environments. Journal of Parallel and Distributed Computing (JPDC), 72(2):254-267 2012.

[Yan12] X. Yang, Z. Wang, J. Xue, and Y. Zhou. The Reliability Wall for Exascale Supercomputing. IEEE Transactions on Computers, 61:767-779, 2012.

[Zhe04] G. Zheng, G. Kakulapati, and L. Kale. BigSim: A Parallel Simulator for Performance Prediction of Extremely Large Parallel Machines. 18th IEEE Intl. Parallel and Distributed Processing Symp. (IPDPS), 2004.