Final Project Report
DE-FG02-08ER25850

Scalable fault tolerance runtime technology for petascale computers

Ohio State University PI: P. Sadayappan (Project PI: Sriram Krishnamoorthy, PNNL)

Statement of Work/Abstract

With the massive number of components comprising the forthcoming petascale computer systems, hardware failures will be routinely encountered during execution of large-scale applications. Due to the multidisciplinary, multiresolution, and multiscale nature of scientific problems that drive the demand for high end systems, applications place increasingly differing demands on the system resources: disk, network, memory, and CPU. In addition to MPI, future applications are expected to use advanced programming models such as those developed under the DARPA HPCS program as well as existing global address space programming models such as Global Arrays, UPC, and Co-Array Fortran. While there has been a considerable amount of work in fault tolerant MPI with a number of strategies and extensions for fault tolerance proposed, virtually none of advanced models proposed for emerging petascale systems is currently fault aware. To achieve fault tolerance, development of underlying runtime and OS technologies able to scale to petascale level is needed. Our team, involving two DoE laboratories, one university and two industrial partners, is proposing to investigate, develop, and evaluate a comprehensive range of runtime techniques for fault tolerance for advanced programming models. Our approach is based on virtualization, checkpoint-restart, and dynamic adaptation. In particular this will include an OS level virtualization based on Xen, global recovery line for the communication runtime layer, fault reconfiguration, transparent system initiated checkpoint-migration-restart, as well as a fault tolerant framework for malleable applications of the global address space programming model. We will evaluate and validate these technologies in context of DOE applications in the computational chemistry and bioinformatics areas.

Note: This was a sub-award to Ohio State University as part of a project led by Sriram Krishnamoorthy at Pacific Northwest National Laboratory. This final project report summarizes work on the entire project.

Significant Studies and Results

Checkpoint-restart is one of the most used software approaches to achieve fault tolerance in high-end clusters. While standard techniques typically focus on user-level solutions, the advent of virtualization software has enabled efficient and transparent system-level approaches. We presented an innovative approach to cluster fault tolerance by integrating Xen virtualization with the latest generation of the InfiniBand network. A major contribution of this approach was the automatic identification of global recovery lines to freeze the status of the machine. We developed a global coordination mechanism and deployed it in the Aggregate Remote Memory Copy Interface (ARMCI) one-sided communication library that has been used as a runtime system for several PGAS languages and libraries. The experimental results showed that it is possible to virtualize communication and computation with minimal overhead and to provide seamless migration capabilities. We developed solutions to transparently distribute checkpoint requests and to backup virtual machines memory images and file systems. We demonstrated that NWChem, a large computational chemistry application, can be executed, without any modification to the source code, on a virtualized 8-node cluster with very little overhead (below 3%). We measured system-size dependent components of the checkpoint time on up to 1024

cores (128 nodes), demonstrating the scalability of our approach in medium/large-scale systems. These results were published in Cluster'07 [8] and Computing Frontiers'09 [6].

We developed extensions to the Global Arrays (GA) toolkit to support user-coordinated fault tolerance through checkpoint/restart operations. GA implements a global address space programming model, is compatible with MPI, and offers bindings to multiple popular serial languages. Our approach used a spare pool of processors to perform reconfiguration after the fault, process virtualization, incremental or full checkpoint scheme and restart capabilities. Experimental evaluation in an application context showed that the overhead introduced by checkpointing is less than 1% of the total execution time. A recovery from a single fault increased the execution time by 8%. These results were published in Parallel Computing [9].

We developed a fault-tolerance mechanism that reduced the cost of failure recovery by maintaining shadow data structures and performing redundant remote memory accesses. Results from NWChem running at scale showed that our techniques provide applications with a high degree of fault tolerance and low overhead. These results were published in PDP'11 [4]. Algorithm-based fault tolerance (ABFT) is a promising approach that involves modifications to the algorithm to recover from faults with lower overheads than replicated storage and a significant reduction in lost work compared to checkpoint-restart techniques. Fault-tolerant linear algebra (FTLA) algorithms employ additional processors that store parities along the dimensions of a matrix to tolerate multiple, simultaneous faults. Existing approaches assume regular data distributions (blocked or block-cyclic) with the failures of each data block being independent. To match the characteristics of failures on parallel computers, we extended these approaches to mapping parity blocks in several important ways. First, we handled parity computation for generalized Cartesian data distributions with each processor holding arbitrary subsets of blocks in a Cartesian-distributed array. Second, techniques to handle correlated failures, i.e., multiple processors that can be expected to fail together, were developed. Third, we handled the colocation of parity blocks with the data blocks and do not require them to be on additional processors. Several alternative approaches, based on graph matching, were designed that attempt to balance the memory overhead on processors while guaranteeing the same fault tolerance properties as existing approaches that assume independent failures on regular blocked data distributions. Evaluation of these algorithms demonstrated that the additional desirable properties are provided by the proposed approach with minimal overhead. These results were published in Computing Frontiers'11 [3] and a special issue of IJPP [1].

Designing an application that is resilient to system failures requires careful evaluation of the impact of various approaches on preserving key application state. We studied the effort required to make NWChem fault tolerant. We constructed the data access signatures of key application modules to evaluate alternative fault tolerance approaches. We developed the instrumentation methodology, characterized various application modules, and evaluated fault tolerance techniques using the information collected. The application signatures developed captured application characteristics not traditionally revealed by performance tools. These results and experiences were published in Euro-Par'11 [2].

Publications, Presentations, IP Resulting from Project
(list all that were generated from the project – IP or ERICA numbers required)

[1] N. Ali, S. Krishnamoorthy, M. Halappanavar, and J. Daily. "Multi-fault tolerance for Cartesian data distributions". International Journal of Parallel Programming, Special

Issue: Computing Frontiers 2011 Best Papers vol:41(3), pp:469-493, 2013. ERICA: PNNL-SA-87233

[2] N. Ali, S. Krishnamoorthy, N. Govind, K. Kowalski, and P. Sadayappan. “Application-specific fault tolerance via data access characterization”. Euro-Par 2011, August 2011. ERICA: PNNL-SA-79368

[3] N. Ali, S. Krishnamoorthy, M. Halappanavar, and J. Daily. “Tolerating correlated failures for generalized Cartesian distributions via bipartite matching”. ACM International Conference on Computing Frontiers, May 2011. ERICA: PNNL-SA-76095

[4] N. Ali, S. Krishnamoorthy, N. Govind, and B. Palmer. “A redundant communication approach to scalable fault tolerance in PGAS programming models ”. 19th Euromicro International Conference on Parallel, Distributed and Network-Based Computing, February 2011. ERICA: PNNL-SA-75835

[5] J. Dinan, A. Singri, P. Sadayappan, and S. Krishnamoorthy. “Selective recovery from failures in a task parallel programming model ”. 10th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing – Resilience Workshop, May 2010. ERICA: PNNL-SA-70950

[6] O. Villa, S. Krishnamoorthy, J. Nieplocha, and D.M. Brown Jr. “Scalable transparent checkpoint-restart of global address space applications on virtual machines over Infiniband”. Conference on Computing Frontiers 2009, April 2009. ERICA: PNNL-SA-64617

[7] “Handling faults in a global address space programming model”. Sriram Krishnamoorthy. Invited talk at the NSF TeraGrid workshop on fault tolerance, Albuquerque, NM, March 2009. ERICA: PNNL-SA-65871

[8] D. P. Scarpazza, P. Mullaney, O. Villa, F. Petrini, V. Tipparaju, D.M.L. Brown, J. Nieplocha. “Transparent system-level migration of PGAS applications using Xen on InfiniBand”. IEEE International Conference on Cluster Computing, September 2007. ERICA: PNNL-SA-55723

[9] V. Tipparaju, M. Krishnan, B. Palmer, F. Petrini, J. Nieplocha. “Towards fault resilient global arrays”. Parallel computing: architectures, algorithms, and applications, 2007. ERICA: PNNL-SA-54426

[10] F. Petrini, J. Nieplocha, V. Tipparaju, “SFT: scalable fault tolerance”. Operating Systems Review volume 40(2), 2006. ERICA: PNNL-SA-52256