

System-Directed Resilience for Exascale Platforms

Proposal No. FY-09-0016

1 Overview of the Problem and Idea

Resilience on MPP systems has traditionally been the responsibility of the application, with the primary tool being application-directed checkpoints. However, as systems continue to increase in size and complexity, the viability of application-directed checkpoint as a solution decreases. Recent studies performed at SNL projected that as systems grow beyond 100,000 components, a combination of factors lead to checkpoint overheads in excess of 50%. In this project, we will investigate critical changes required in MPP systems software to support system-directed resilience. The goal is to provide efficient, application-transparent resilience through coordinated use of system resources. The primary research topics focus around the problem of continuous computing in the event of a component failure. A preliminary list of required new capabilities include:

- **Application Quiescence:** the ability to suspend CPU, network, and storage services used by an individual application without interfering with the progress of other applications;
- **State Management:** the ability to identify, extract, and manage application state in a transparent, efficient, and non-intrusive way; and
- **Fault Recovery:** the ability to transparently replace a failed component without restarting the entire application.

2 Proposed R&D

2.1 Technical Approach and Leading Edge Nature of Work:

A viable solution to resilience on exascale systems requires extensive research in each area identified in the Section 1. Here, we describe our technical approach for each of the three areas and briefly discuss some of the challenges unique to the exascale domain.

To efficiently quiesce a large-scale application not only requires cooperation among the application processes, it also requires integration and cooperation with shared services like the network, scheduler, and storage system. This work first requires an extensive investigation of existing approaches with particular attention paid to the resource requirements and overhead costs of each approach. Next, we will design and validate, through performance modeling and/or simulation, an approach that is both resource efficient and has minimal impact on external applications. In particular, our approach will have to deal with messages in transit, in-progress file system operations, and interactions with various other shared services. Finally, we will implement a prototype of the design, integrate it with the other system components, and design and perform experiments to evaluate the approach.

Efficient state management is perhaps the largest performance challenge for exascale resilience. On today's systems, the I/O associated with checkpoint data (for application-directed checkpoints) accounts for nearly 80% of the total I/O of the system [8]. System-directed approaches are not viable even for small parallel applications because current approaches have tremendous resource requirements. The typical system-directed scheme extracts the entire memory footprint of an application for a checkpoint, while application-directed approaches only checkpoint between 20-50% of their processor memory¹. To address this issue, we identify the critical application memory required to restart a failed process by first characterizing the memory usage of existing MPP applications. In particular, we want

¹Based on an informal survey of Sandia application developers, applications write between 20-50% of the total memory used by the application to restart files.

to know what proportion of an application’s memory changes between checkpoints and what portion is absolutely critical to the application. For example, in a finite-difference code, memory for ghost cells can safely be excluded from a checkpoint. In addition to identifying critical memory, we will explore ways to further reduce the amount of state that needs to be managed (e.g., data compression); we will leverage Reliability, Availability, and Serviceability (RAS) systems to help guide how and when to extract state; and we will investigate diskless approaches so as to reduce latency of extraction and recovery. While each of these issues are important, due to funding constraints we will focus primarily on identifying critical state, and integration with RAS systems. However, as we discuss in Section 3.2, some of the work on diskless approaches and data-reduction is actively being researched as part of related project.

The third research area is system support for recovery. Applications currently use what is effectively a “roll-back” approach that requires the user to kill the application after a single node failure, re-allocate all the resources for the application, load the data from the last checkpoint, and finally continue computing. This approach is time-consuming, wastes valuable resources, and is unfair to applications that have to re-submit their failed job through a batch queue system. To address this issue, we will investigate modifications to current system software to enable dynamic resource scheduling, we will investigate virtualization (of the network and the operating system), and we will evaluate different algorithms for roll-back and roll-forward recovery techniques for large-scale applications.

The challenge (and inherent risk) in a systems-driven approach to resilience for MPP systems is (and always has been) performance. A viable solution needs to be sensitive to resource requirements, scale, and performance issues that are particularly demanding in exascale systems. Although there is risk that a system-directed approach may still require significant overhead on the application, we are confident that our combined expertise on MPP operating systems, networking, and storage systems will lead to a solution that is efficient and reduces the burden of resilience on the application developer.

2.2 Key R&D Goals and Project Milestones:

Goal	Milestone	Completion Date
App Quiescence	Investigate options for application quiescence.	02/01/2009
	Design systems software to support quiescence.	06/01/2009
	Complete prototypes that demonstrate quiescence.	02/01/2010
	Evaluate overheads and impact on external apps.	04/01/2010
State Management	Characterize application behavior.	03/01/2009
	Design of algorithm to identify critical state	09/01/2009
	Complete interface with RAS system to guide how/when to extract state	04/01/2010
Fault Recovery	Investigate system software to support dynaming node allocation, network/os virtualization, and MPI node recovery.	02/01/2009
	Design prototypes for independent node recovery.	09/01/2009
	Complete prototype system software for node recovery	02/01/2010
	Evaluate performance of prototype. Compare with traditional approach.	04/01/2010
System Integration	Develop prototype that integrates quiescence, state management, and fault-recovery components	12/20/2010
	Evaluate performance of complete prototype on real applications.	04/01/2011
Communicate Results	SAND Reports (Quiescence investigation, App characterization, Recovery)	04/01/2009
	SAND reports for prototype designs	12/20/2009
	Publish performance results of prototypes	04/01/2010
	Publish complete system design/performance (Journal)	06/01/2011

2.3 Technical Risk and Likelihood of Success

- Unable to identify critical state without application involvement.
 - Develop hybrid approach that allows application to identify critical state (e.g., special malloc).
- Unable to access testbed suitable for scalability testing.
 - Develop software on small-scale development systems. Work with vendors and Sandia production team to get dedicated time on large systems.
 - Develop detailed simulation systems (building on Seshat and SST) to evaluate scalability of designs.
- Underestimate development effort.
 - The goals and milestones are intentionally incremental. If the requested funding and staff is not sufficient to complete development, there will still be substantial progress made through research and design to advance future resilience efforts.

3 Relationship to Other Work:

3.1 Previous and Other Ongoing Work:

Fault tolerance has long been of interest in the parallel computing and cluster community. Elnozahy et al. present a nice survey of approaches for parallel computing [3]. Some of the more interesting approaches particularly relevant to this project include diskless checkpointing [9], fault-tolerant MPI [4], and virtualization techniques for fault tolerance [5]. Our project will leverage some of the techniques developed in this previous work and adapt them to more appropriately match an exascale computing environment.

We will also leverage work from an ongoing LDRD to investigate lightweight storage and overlay networks for fault tolerance. In its final year (FY09), that LDRD proposes to investigate novel ways to manage application state in the memory of overlay nodes as a way to avoid the I/O overhead costs of checkpoints. Since the goals of the ongoing LDRD overlap with some of our deliverables from Section 2.1, we will work closely with that team with the goal of incorporating their results into this project.

3.2 Relationship to Other Work:

Despite all of the previous work on system-directed approaches, the only approach widely used in practice on MPP systems is application-directed checkpointing. The primary reason is scalability. Using previous approaches, system supported resilience creates overheads well beyond the overheads of an application-directed approach. In small clusters, these overheads are relatively small, but when the application scales to tens of thousands of nodes, the overheads quickly become unmanageable. The next-generation of applications for exascale systems will likely use millions of cores. On systems of this size, even application-directed checkpoints are unreasonable. We intend to use our broad experience in MPP system software research in operating systems [1], networking [2], storage systems [7], and fault tolerance [6] to design and develop a systems approach to resilience that identifies critical application state needed for a restart, minimizes the I/O overhead of state management, and uses advanced techniques such as virtualization and dynamic process allocation for fast independent node recovery. With these ideas and the experience of this team, we are confident we can make substantial impact on the approach used for resilience on next-generation systems.

4 Resources

4.1 Key Research Team Members

The key team members for this project consist of a diverse group of experts with established records developing systems software for HPC systems. This group has particular expertise in operating systems design, system virtualization, networking, storage, architecture, RAS, performance modeling, and fault-tolerant algorithms.

Name	Org	FTE	Role
Oldfield, Ron (PI)	1423	0.5	Project Management, R&D
Pedretti, Kevin	1423	0.35	R&D support
Brightwell, Ronald	1423	0.35	R&D support
Riesen, Rolf	1423	0.35	R&D support
Laros, James	1422	0.35	R&D support
Stearley, Jon	1422	0	Consultant
Ulmer, Craig	8963	0	Consultant
Minnich, Ronald	8961	0	Consultant

4.2 Qualifications of the Team to Perform This Work:

Ron Oldfield has led a number of research projects in systems software, scalable I/O, and resilience. He was technical lead in the SciDAC-sponsored scalable systems software project (completed in 2005), and is currently PI of the CSRF-funded Lightweight File Systems Project, and an LDRD titled, “Lightweight Storage and Overlay Networks for Fault Tolerance”. He also leads a number of collaborations with students and faculty at the University of New Mexico, Georgia Institute of Technology, and the University of Texas at El Paso.

Kevin Pedretti, Ron Brightwell, Rolf Riesen, and Jim Laros are all key designers and developers of Red Storm system software. Pedretti’s expertise in Lightweight Kernel design and multi-core issues is particularly relevant to resilience issues for future architectures; Brightwell’s experience with operating systems, virtualization, Portals, and MPI will contribute to the quiescence and recovery pieces of the project; Riesen’s experience with Portals design and his recent work on simulation and performance modeling will aid the design and validation portion of the project; and Laros’ expertise in RAS systems, catamount development, and detailed knowledge of the low-level I/O systems will aid in all aspects of this project. Due to funding constraints, Stearley, Ulmer, and Minnich will not be consistently charging this project; however, we will occasionally leverage their expertise on RAS, system software, storage, and architecture and therefore have listed them as consultants.

5 Strategic Alignment and Potential Benefit

5.1 Relevance to Missions:

This project has direct relevance to the Science, Technology, and Engineering mission of Sandia National Laboratories, if successful, it will have a direct impact on applications in virtually all areas of advanced computing.

5.2 Programmatic Benefit to Investment Area, if Successful:

Current challenges in DOE ASC are: confidence levels derived through statistical verification and validation, and uncertainty quantification. These require capability runs at the tri-Labs. If successful, this project will significantly benefit these efforts, as well as other efforts in the *Enabling Predictive Simulation* investment area.

5.3 Communication of Results:

We will publish the initial studies and design documents for all aspects of this project in SAND reports. As appropriate, we will also present some of the studies at ACM/IEEE conferences as position papers or at user-group meetings (e.g., CUG) as works in progress. We will target well-known ACM/IEEE supercomputing conferences (e.g., IPDPS, SC, Cluster) for publication of performance analysis and prototype results. As we approach completion of the project, we will publish detailed results, design, and lessons learned in well known Journals such as the Journal of Parallel and Distributed Computing (JPDC) and IEEE Transactions on Parallel and Distributed Systems (TPDS).

In addition to the standard publications such as Conferences, Journals, and user-group meetings; we will actively pursue vendor participation to incorporate our research into a commercialized product. The ultimate goal is to see the results of our work continue beyond the life of the project. Our best chance at that is through vendor participation and adoption.

References

- [1] Ron Brightwell, William Camp, Benjamin Cole, Erik DeBenedictis, Robert Leland, James Tomkins, and Arthur B. Maccabe. Architectural specification for massively parallel computers: an experience and measurement-based approach. *Concurrency and Computation: Practice and Experience*, 17(10):1271–1316, March 2005.
- [2] Ron Brightwell, Tramm Hudson, Arthur B. Maccabe, and Rolf Riesen. The Portals 3.0 message passing interface. Technical Report SAND99-2959, Sandia National Laboratories, November 1999.
- [3] E. N. Elnozahy, L. Alvisi, Y. M. Wang, and D. B. Johnson. A survey of rollback-recovery protocols in message-passing systems. *ACM Computing Surveys*, 34(3):375 – 408, SEP 2002.
- [4] Joshua Hursey, Jeffrey M. Squyres, Timothy I. Mattox, and Andrew Lumsdaine. The design and implementation of checkpoint/restart process fault tolerance for Open MPI. In *Proceedings of the 21st IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE Computer Society, March 2007.
- [5] Arun Babu Nagarajan, Frank Mueller, Christian Engelmann, and Stephen L. Scott. Proactive fault tolerance for hpc with xen virtualization. In *ICS '07: Proceedings of the 21st annual international conference on Supercomputing*, pages 23–32, New York, NY, USA, 2007. ACM.
- [6] Ron A. Oldfield, Sarala Arunagiri, Patricia J. Teller, Seetharami Seelam, Rolf Riesen, Maria Ruiz Varela, and Philip C. Roth. Modeling the impact of checkpoints on next-generation systems. In *Proceedings of the 24th IEEE Conference on Mass Storage Systems and Technologies*, San Diego, CA, September 2007.
- [7] Ron A. Oldfield, Arthur B. Maccabe, Sarala Arunagiri, Todd Kordenbrock, Rolf Riesen, Lee Ward, and Patrick Widener. Lightweight I/O for scientific applications. In *Proceedings of the IEEE International Conference on Cluster Computing*, Barcelona, Spain, September 2006.
- [8] Fabrizio Petrini and Kei Davis. Tutorial: Achieving Usability and Efficiency in Large-Scale Parallel Computing Systems, August 31, 2004. Euro-Par 2004, Pisa, Italy.
- [9] L. M. Silva and J. G. Silva. An experimental study about diskless checkpointing. In *24th EUROMICRO Conference*, pages 395 – 402, Vasteras, Sweden, August 1998. IEEE Computer Society Press.