Grant-in-Aid for Scientific Research S (Kiban S)

「Fault Tolerant Infrastructure

Toward Billion of Parallelization and Exa-scale Supercomputer」

(Adopted FY2011)

1) Motivations

The growing computational power of high performance computing (HPC) systems enables increasingly larger simulations. However, as the number of system components increase, the overall failure rate of systems increases. Further, the mean time between failures (MTBF) of future systems is projected to be on the order of tens of minutes or hours [1, 2, 3] at exascale. The main objectives are to identify bottlenecks in the current design approach from the highest level, algorithms, to the lowest level, system design, which prevents applications to run at exascale. Our works focus on the indirect performance/scalability improvements achievable with fault tolerant techniques and an enhanced resilient network layer, rather than the direct improvements achievable via source code redesign.

Checkpointing is one of indispensable fault tolerance techniques, commonly used by HPC applications that run continuously for hours or days at a time. A checkpoint is a snapshot of application state that can be used to restart execution if a failure occurs. However, when checkpointing large-scale systems, tens of thousands of compute nodes write checkpoints to a parallel file system (PFS) concurrently, and the low I/O throughput becomes a bottleneck. Although simple, this straightforward checkpointing scheme can impose huge overheads on application run times.

2) Researches (Our way of addressing the problem)

Asynchronous Checkpointing System: We developed an asynchronous checkpointing system to minimize checkpointing overhead/workload to PFS (Figure 1). The asynchronous checkpointing system solves the problem through agents running on additional nodes that asynchronously transfer checkpoints from the compute nodes to the PFS. Our asynchronous checkpointing model optimize the checkpointing frequency (Figure 2). Our approach has two key advantages. It lowers application checkpoint overhead by overlapping computation and writing checkpoints to the PFS. Also, it reduces PFS load by using fewer concurrent writers and moderating the rate of PFS I/O operations.

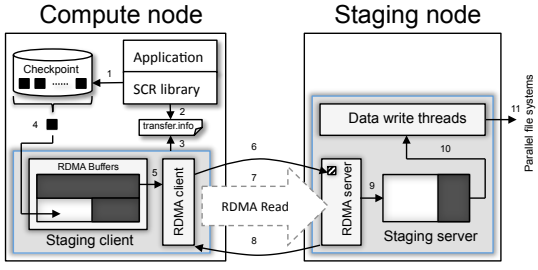

Figure 1 : Asynchronous Checkpointing client/server using RDMA

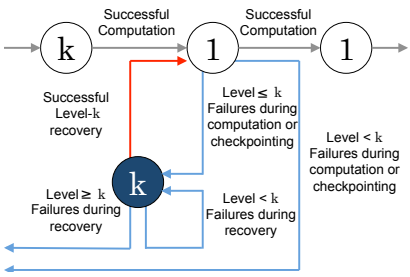

Figure 2 : The basic structure of the asynchronous checkpointing model

Exploration of Multi-tier Storage Design with Burst Buffer: We explored multi-tier storage design for resilient architecture using burst buffers. Burst buffers have been proposed to alleviate the problems of I/O operations to a shared PFS [4, 5]. A Burst buffer is a new tier in the storage hierarchy to fill the performance gap between node-local storage and the PFS, and is shared by a subset of compute nodes. The new tire can absorb the bursty I/O requests from applications and thus can reduce the effective load on the PFS. We considered using burst buffers from different viewpoint, and tried to improve system resiliency with burst buffer storage design. With burst buffers, an application can store checkpoints on a smaller number of dedicated burst buffer nodes, so the probability of lost checkpoints is decreased. We explored how burst buffers can improve efficiency and resiliency compared to using node-local storage instead of burst buffers based on a performance model combined with our multi-level asynchronous checkpoint/restart model.

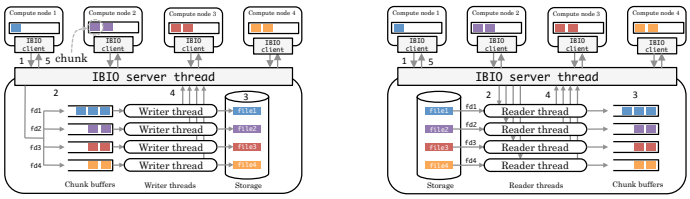

Figure 3 : IBIO Write/Read : four IBIO clients and one IBIO server

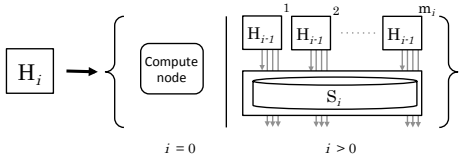

Figure 4 : Recursive structured storage model

FMI (Fault Tolerant Messaging Interface): We proposed FMI for fast and transparent recovery. FMI is a survivable messaging interface that uses fast, transparent in-memory checkpoint/restart and dynamic node allocation. With FMI, a developer writes an application using semantics similar to Message Passing Interface (MPI). The FMI runtime ensures that the application runs through failures by handling the activities needed for fault tolerance, such as checkpointing, failure detections, and recoveries. All of this motivates the need for a survivable messaging runtime system. Such a system should be able to maintain processes and connections that are unaffected by the failure while starting and integrating replacement processes as needed.

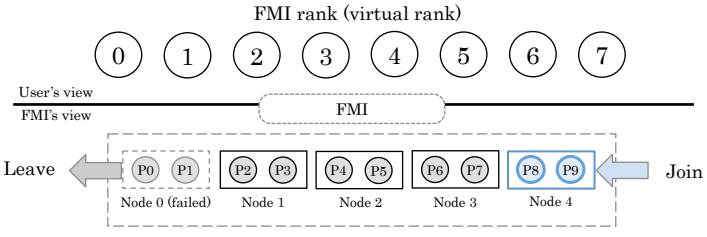

Figure 5 : Overview of FMI

Fail-in-Place Large scale Network Design: As a first stage, we built a tool chain (Figure 6), which allows us to simulate the performance of different routing algorithms on state-of-the-art network topologies. This was combined with an injection of network failures to make design decisions for a future fail-in-place network. Fail-in-place networks will enable high communication performance, a crucial component for application scalability, while being extremely resilient for non-critical network failures, which increases the system availability for scientific simulations and decreases the need for application level resiliency.

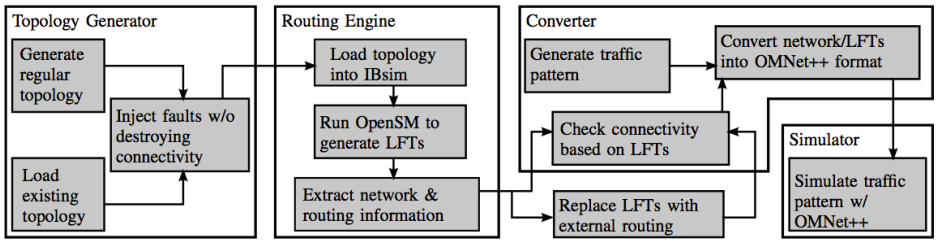

Figure 6 : Toolchain to evaluate the network throughput of a fail-in-place network

Lossy compression for fast checkpint/restart : To reduce checkpoint and restart time, we explored application-level lossy compression approach based a wavelat transformation [6] . Although lossy compression can introduce errors to the simulation data after a restart, applications can proceed and produce approximate results even with high failure rate (Figure 7).

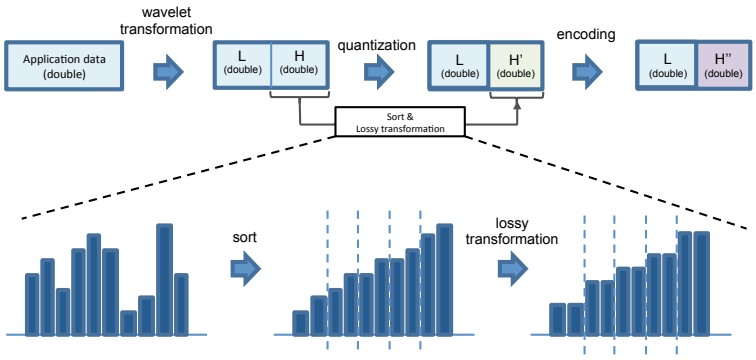

Figure 7 : Application-level lossy compression using a wavelet transformation

3) Results (what did we learn, what did you produce?)

Asynchronous Checkpointing System: Our asynchronous (or Non-blocking) checkpointing system coupled with a multi-level checkpoint/restart technique maintains a given application efficiency with significantly lower PFS requirements than simple synchronous (or Blocking) checkpointing, which write checkpoints such that all processes write their own checkpoints concurrently, and are blocked until the checkpoint operation completes. Especially, our results show that combining asynchronous and multi-level checkpointing results in highly efficient application runs with low PFS bandwidth requirements.

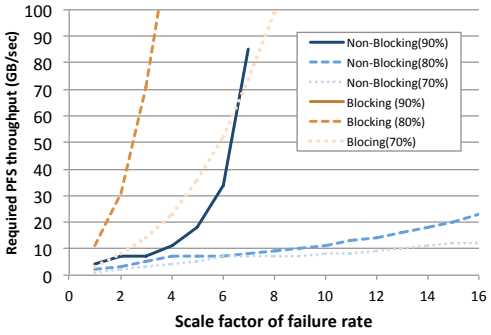

Figure 8 : Required PFS throughput at different failure rates

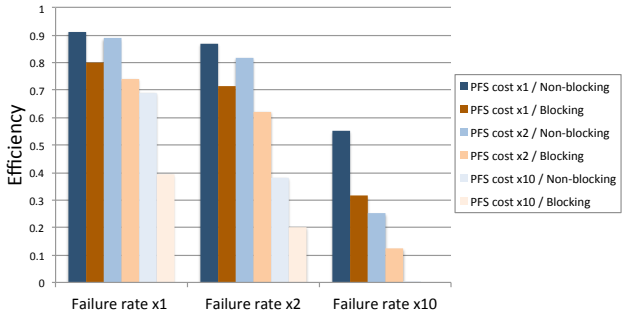

Figure 9 : Efficiency of synchronous and asynchronous checkpointing

Exploration of Multi-tier Storage Design with Burst Buffer: The key contributions of this work include an InfiniBand-based file system (IBIO) that exploits the bandwidth of burst buffers, and exploration showing how system resiliency improves from the use of burst buffers, and uncoordinated checkpinting. Especially, these contributions can benefit system designers in making the trade-offs in performance of components so that they can create efficient and cost-effective machines for future extreme scale systems.

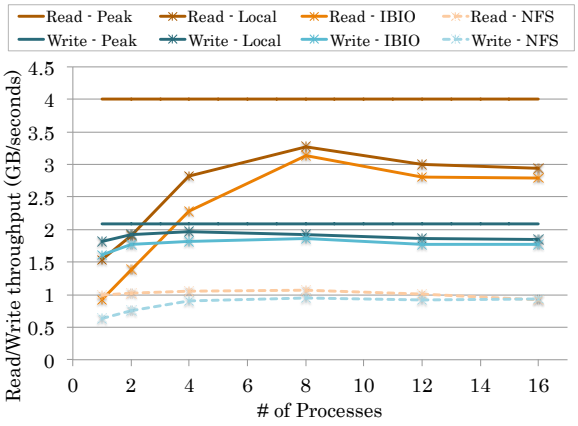

Figure 9 : Sequencial read and write throughput of local I/O, and I/O with IBIO and NFS via FDR InfiniBand networks

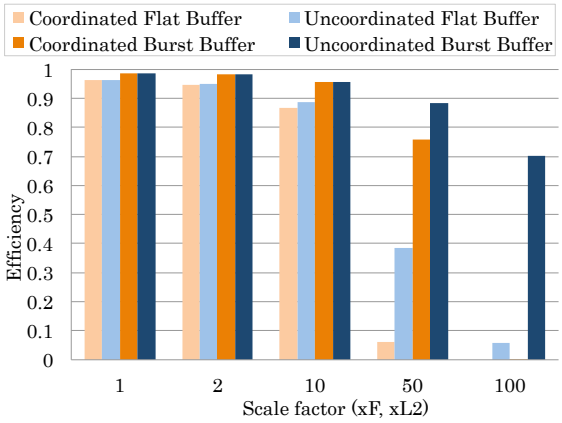

Figure 10 : Efficiency of multilevel coordinated and uncoordinated checkpoint/restart ona flat buffer system and a burst buffer system

FMI (Fault Tolerant Messaging Interface) : The key contributions include a simplified programming model to enable fast, transparent fault tolerance based on checkpoint/restart; implementation of a runtime that withstands process failures and allocates spare resources; a new overlay network structure called log-ring for scalable failure detection and notification; and demonstration of the fault tolerance and scalability of FMI even with a MTBF of 1 minute. Especially, our implementation of FMI has failure-free performance that is comparable with MPI, and our experiments with a Poisson equation solver show that running with FMI incurs only a 28% overhead with a very high mean time to failure of 1 minute.

	1-byte Latency	Bandwidth (8MB)
MPI	3.555 usec	3.227 GB/s
FMI	3.573 usec	3.211 GB/s

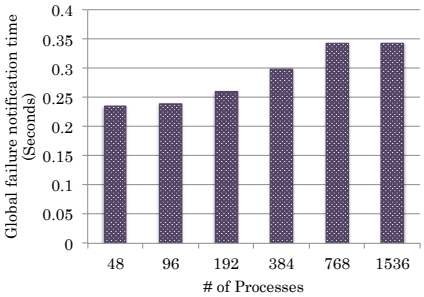

Figure 11 :Failure notification time with log-ring overlay network

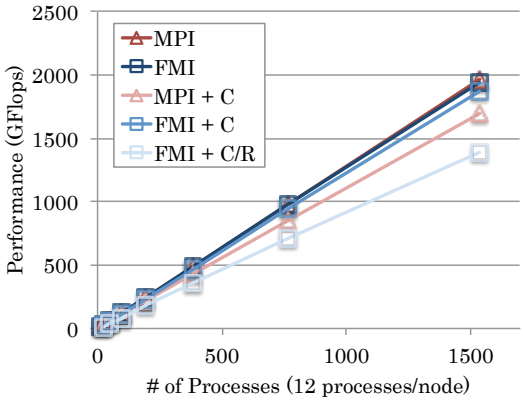

Figure 12 : Himeno benchmark (Checkpoint size : 821MB/node, MTBF : 1 minute)

Fail-in-Place Large scale Network Design: In conclusion of our simulations, the change of the routing algorithm from the currently used Up*/Down* routing to DFSSSP routing on TSUBAME2.5 would not only lead to a higher performance of the MPI_Alltoall on the fault-free network, as shown in Fig. 2, but also will increase the fail-in-place characteristic of the network. Both will support the efforts to achieve exascale scientific simulations.

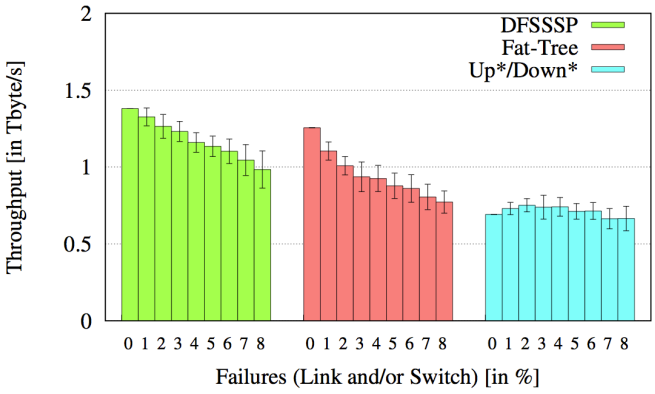

Figure 13 : MPI_Alltoall runtime simulation for TSUABME2.5 using different routing algorithms while network failures have been injected

Lossy compression for fast checkpint/restart : Our preliminary studies show that our lossy compression approach can reduce size of simulation data of a real climate application, NICAM, to 12-13% with 0.2 to 0.8 of errors on average error.

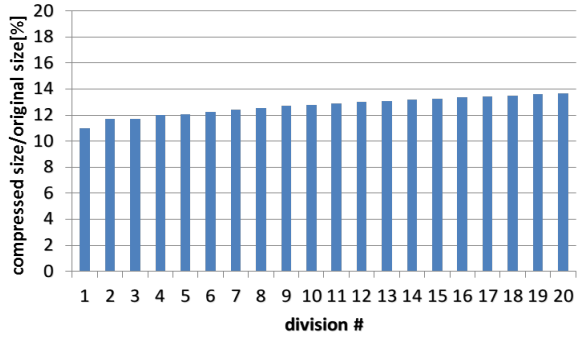

Figure 14 :Ratio of compressed checkpoint size to original checkpoint size under increasing division #

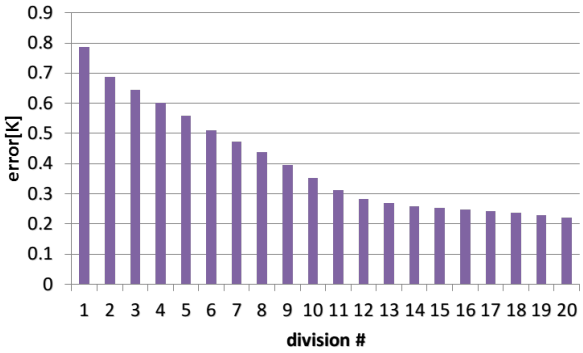

Figure 15 : Average errors of uncompressed checkpoint to original value under increasing division #

4) Meeting (other than the monthly conf calls and the G8 ECS workshops)

- Meeting with Franck Cappello at Titech, ANL-UIUC and INRIA, 2th-4th June 2012
- Meeting with John Dennis at Titech, 31th March 2014
- Meeting with Leonard Bautista Gomez at Titech, 31th March 2014

5) Visits (visiting partners or hosting partners other than G8 ECS workshops)

- Ana Gainaru (Ph.D. at University of Illinois), June 2012 to August 2013

6) Impact

Publications:

[SMK12] Kento Sato, Adam Moody, Kathryn Mohror, Todd Gamblin, Bronis R. de Supinski, Naoya Maruyama and Satoshi Matsuoka, "Design and Modeling of a Non-blocking Checkpointing System", In Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis 2012 (SC12), Salt Lake, USA, Nov, 2012.

[SSS13] Takafumi Saito, Kento Sato, Hitoshi Sato and Satoshi Matsuoka, "Energy-aware I/O Optimization for Checkpoint and Restart on a NAND Flash Memory System", In the Workshop on Fault-Tolerance for HPC at Extreme Scale 2013 (FTXS2013) in conjunction with the International Symposium on High Performance Parallel and Distributed Computing (HPDC13), New York, USA, June, 2013.

[SMM13] Kento Sato, Satoshi Matsuoka, Adam Moody, Kathryn Mohror, Todd Gamblin, Bronis R. de Supinski and Naoya Maruyama, "Burst SSD Buffer: Checkpoint Strategy at Extreme Scale", IPSJ SIG Technical Reports 2013-HPC-141, Okinawa, Sep, 2013.

[SMM14] Kento Sato, Adam Moody, Kathryn Mohror, Todd Gamblin, Bronis R. de Supinski, Naoya Maruyama and Satoshi Matsuoka, "FMI: Fault Tolerant Messaging Interface for Fast and Transparent Recovery", In Proceedings of the International

Conference on Parallel and Distributed Processing Symposium 2014 (IPDPS2014), Phoenix, USA, May, 2014.

[SMM14-2] Kento Sato, Kathryn Mohror, Adam Moody, Todd Gamblin, Bronis R. de Supinski, Naoya Maruyama and Satoshi Matsuoka, "A User-level InfiniBand-based File System and Checkpoint Strategy for Burst Buffers", In Proceedings of the 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid2014), Chicago, USA, May, 2014. (Best Paper Finalist)

[SSE14] Naoto Sasaki, Kento Sato, Toshio Endo and Satoshi Matsuoka, "Exploration of Application-level Lossy Compression for Fast Checkpoint/Restart", In HPC in Asia Workshop in conjunction with the International Supercomputing Conference (ISC'14), Leipzig, Germany, June, 2014.

7) Future plans for the next 18 months (please look carefully at the tasks and deliverable)

Our objectives in the next 18 months are related to part of task 3.3 and D3.3:

Asynchronous Checkpointing System: We will integrate the asynchronous checkpointing system with SCR (Scalable Checkpoint/Restart), which was developed by LLNL.

Exploration of Multi-tier Storage Design with Burst Buffer: We will integrate the asynchronous checkpointing system with SCR (Scalable Checkpoint

FMI (Fault Tolerant Messaging Interface): Although the current FMI prototype has demonstrated promising results, it not yet complete enough to support a broad range of applications. Future FMI will support I/O interfaces like MPI I/O, restoring communicators along with checkpoints, exploit multi-level checkpointing.

Fail-in-Place Large scale Network Design: A next step will be a detailed analysis of the communication layer of the applications and a co-design phase to match the needs of the applications with the underlying network infrastructure, i.e., topology and routing algorithm.

Lossy compression for fast checkpint/restart : We will reduce the errors, and compression time.

8) References

[1] Bianca Schroeder and Garth A. Gibson. Understanding Failures in Petascale Computers. Journal of Physics: Conference Series, 78(1):012022+, July 2007

[2] Al Geist and Christian Engelmann. Development of Naturally Fault Tolerant Algorithms for Computing on 100,000 Processors, 2002.

[3] John Daly et al. Inter-Agency Workshop on HPC Resilience at Extreme Scale, February 2012.

[4] Ning Liu, Jason Cope, Philip H. Carns, Christopher D. Carothers, Robert B. Ross, Gary Grider, Adam Crume, and Carlos Maltzahn. On the Role of Burst Buffers in Leadership-Class Storage Systems. In Symposium on Mass Storage Systems and Technologies, MSST 2012, April 2012.

[5] Dries Kimpe, Kathryn Mohror, Adam Moody, Brian Van Essen, Maya Gokhale, Rob Ross, and Bronis R. de Supinski. Integrated In-System Storage Architecture for High Performance Computing. In Proceedings of the 2nd International Workshop on Runtime and Operating Systems for Supercomputers, ROSS '12, 2012.

[6] A. Graps, "An introduction to wavelets," Computational Sci- ence Engineering, IEEE, vol. 2, no. 2, pp. 50–61, Summer 1995.