Deploying Checkpoint/Restart for Production Workloads at NERSC

Zhengji Zhao
NERSC
Lawrence Berkeley National laboratory
Berkeley, USA
zzhao@lbl.gov

Rebecca Hartman–Baker
NERSC
Lawrence Berkeley National laboratory
Berkeley, USA
rjhartmanbaker@lbl.gov

Gene Cooperman
Khoury College of Computer Sciences
Notheastern University
Boston, USA
gene@ccs.neu.edu

***Abstract*—Checkpoint/restart (C/R) is a critical component of fault-tolerant computing, and provides scheduling flexibility for computing centers to support diverse workloads with different priorities. Because existing C/R tools are often research-oriented, there is a gap to close before they can be used reliably with production workloads, especially on cutting-edge HPC systems. In this talk, we present our strategy to enable C/R capabilities on NERSC production workloads, which are dominated by MPI and hybrid MPI+OpenMP applications. We share our journey to prepare a production-ready MPI-Agnostic Network-Agnostic (MANA) Distributed Multi-Threaded CheckPointing (DMTCP) tool for NERSC. We also present variable-time job scripts to automate preempted job submissions, queue policies and configurations we have adopted to incentivize C/R usage, our user training effort to increase NERSC users' uptake of C/R, and our effort to build an active C/R community. Finally, we showcase some applications enabled by C/R.**

***Index Terms*—checkpoint/restart, MANA, DMTCP, production, HPC, preemption**

Checkpoint/restart (C/R) is critical to fault-tolerant computing, and provides scheduling flexibility for computing centers to support diverse workloads with different priorities. For example, the DOE SC's experimental and observational facilities often require real-time access to computing resources to generate immediate feedback for follow-up experiments. These kinds of real-time workloads will continue to increase in the near future at NERSC, meaning we may soon need the capability to preempt partial or full system running jobs when these demanding real-time jobs enter our system. It is therefore vital to get transparent C/R capability working at NERSC.

For many years, no viable C/R tools were available for production workloads on HPC systems. Even now, existing C/R tools are often research-oriented and meant to demonstrate promising C/R capabilities on specific platforms. There is thus a large gap to close before these tools can reliably handle a wide range of different applications and cutting-edge HPC systems. Unfortunately, maintaining C/R codes for production use has low priority among C/R researchers, as it does not produce novel research results that can secure funding from stakeholders. NERSC is therefore the perfect player to bring C/R research to production usage, because our funding is not dependent on producing novel research results. With this knowledge and our own need, we started our journey to enable C/R capability for NERSC production workloads in mid-2017.

We began by analyzing the many challenges in C/R code maintenance, as making transparent C/R tools available for production workloads is hard! It is labor intensive and highly complex because of ever-changing HPC systems and diverse production workloads. First, for HPC workloads which are dominated by MPI and hybrid MPI+OpenMP applications, C/R tools that work across many combinations of MPI implementations and networks are exceedingly difficult to maintain. This is the so called M (# of MPI implementations) $\times N$ (# of Networks) maintenance penalty [1] (see Figure 1). Additionally, since front-edge computing centers introduce new HPC systems every few years, the already numerous MPI implementation-network combinations also constantly change. This fast turnover is extremely difficult for code developers to keep pace with, meaning there are no production-ready transparent C/R tools on cutting-edge HPC systems. Second, C/R tools that require long-term coordination between MPI library, kernel, and resource manager/scheduler developers have proven unsustainable; this was the primary challenge in the once promising and influential Berkeley Lab Checkpoint/Restart (BLCR) [2]. Third, transparent C/R tools often interact directly with OS kernels, and frequent OS updates on HPC systems can easily break C/R tools. Finally, using C/R tools incurs runtime overheads and imposes extra work upon users, which hinders the uptake of the C/R approach.

To address these challenges, we selected Distributed Multi-Threaded CheckPointing (DMTCP) [3] as our C/R tool, which lives completely in user space and does not require coordination with MPI, kernel and batch system developers. With DMTCP, we can also bring C/R capabilities to NERSC in an incremental manner, which is critical to the project's long-term success.

Our collaboration with the DMTCP team began in March 2018, with the initial goal of enabling DMTCP to work with NERSC's serial/threaded applications reliably [4]. Meanwhile, we developed variable-time job scripts [5] [6], which auto-

This work was supported by the Office of Advanced Scientific Computing Research in the Department of Energy Office of Science under contract number DE-AC02-05CH11231. The work of the third author was supported by National Science Foundation Grant OAC-1740218 and by a grant by Intel Corporation.

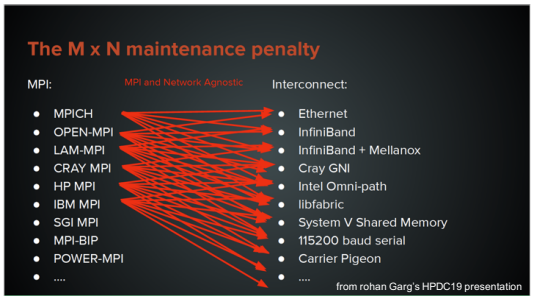

Fig. 1. The $M \times N$ maintenance penalty of C/R tools.

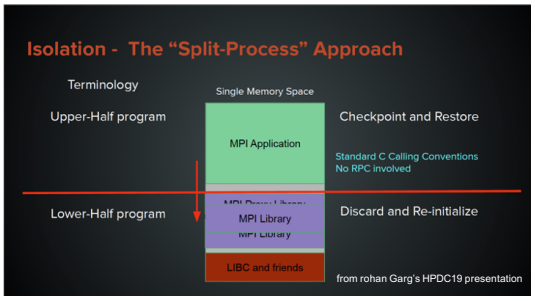

Fig. 2. MANA is based on the "split-process" approach, in which a system process contains two separate programs, the upper half, containing the original MPI application and data, and the lower half, containing MPI library, libc and other system libraries. The MPI-Agnosticism is achieved by checkpointing only the upper half memory and discarding the lower half, and then reinitializing MPI upon restart.

matically split a long-running job into multiple shorter ones that self-resubmit until the job completes. This was a key step towards promoting user uptake of the C/R approach, making it easier for users to work with preempted jobs with or without external C/R tools. We also rolled out queue policies that provide a large charging discount for variable-time jobs to incentivize C/R usage. We hosted multiple user training sessions [7]- [11] to help NERSC users to adopt variable-time job scripts and the C/R approach in their production workloads with or without DMTCP.

A breakthrough in the DMTCP implementation in May 2019, called the MPI-Agnostic Network-Agnostic transparent checkpointing (MANA) [1] [12], addressed the critical $M \times N$ maintenance issue. MANA is implemented as a plugin for DMTCP. Based on a novel "split-process" approach [1], MANA is fully transparent to the underlying MPI, network, libc library and underlying Linux kernel. In a split-process, a single system process contains two programs in its memory address space, the lower half, containing an MPI proxy application with MPI library, libc and other system libraries, and the upper half, containing the original MPI application and data. (see Figure 2). MANA tracks which memory regions belong to the upper and lower halves, and achieves MPI agnosticism by checkpointing only the upper-half memory and discarding the lower-half memory at checkpoint, and then reinitializing MPI library upon restart. MANA achieves network agnosticism by draining MPI messages before checkpointing. This was a huge step forward toward ready-to-use C/R tools on future HPC platforms!

In this talk, we present our long-term C/R strategy. First, we plan to use MANA-enabled DMTCP as our production C/R tool. As of this writing, we have collaborated with the DMTCP team and summer interns to make MANA reliably work with a few applications at NERSC. Chief among them is our top ranked application, VASP [13], a materials science code consuming more than 20% of computing cycles at NERSC [14]. We have also evaluated C/R overhead at scale with HPCG [15] on Cori's [16] Lustre file system and Burst Buffer to prepare for MANA deployment with large-scale applications.

Second, we actively promote the uptake of C/R for NERSC users. We developed variable-time job scripts to automate preempted job submissions, queue policies and configurations to incentivize C/R usage, and user training to increase the uptake of C/R. The "flex" queue we created offers a 75% charging discount in exchange for users' flexibility about job walltime. A flex job must request a minimum walltime of at most two hours, and can use up to 256 KNL nodes for up to 48 hours. This creates opportunities for the Slurm scheduler to perform backfill and increase machine utilization. In 2020, the "flex" queue was used by over 150 distinct users, who consumed nearly 3% of all KNL cycles, enabling NERSC to achieve more than 90% utilization on Cori. More than half of these "flex" users ran their jobs with variable-time job scripts.

Our third strategy is to build an active and strong C/R community to promote production-ready C/R tools development. An active community will ensure the long-term sustainability of the C/R approaches.

Finally, we showcase a few applications enabled by DMTCP and MANA. One such application is SPAdes [17], a genome assembler written in Python, C++, and parallelized with OpenMP. SPAdes is one of the most important production workloads of The Joint Genome Institute at Lawrence Berkeley National Laboratory. SPAdes jobs run much longer than the max walltime allowed on the system, but there is no built-in C/R support. DMTCP has enabled long running SPAdes production workloads on Cori, and the variable-time job scripts have made checkpointing and restarting SPAdes jobs more manageable. This was possible only after the DMTCP team fixed several bugs exposed by this special workflow, which uses a large number of threads (e.g., 72), saves a checkpoint image of about a terabyte to the Lustre file system, and requires temporary files that get deleted over the course of execution to properly restart the run, etc. The largest cases with about a terabyte memory footprint still suffer from high checkpoint overhead [18], and we are currently looking into various I/O options for them.

A second application we showcase is VASP [13]. VASP is a widely used materials science code, written in Fortran 90 and parallelized with MPI (version 5) or MPI+OpenMP (version 6). VASP represents more than 20% of computing cycles at NERSC [14], with 455 active users in 2020. VASP's atomic

relaxation jobs have been running with variable-time scripts in the flex queue with its internal C/R; however, some features implemented in VASP have no internal C/R support, e.g., Random Phase Approximation (RPA) jobs, which could run for much longer than 48 hours, the max walltime allowed on Cori. While we had to make special reservations for these jobs in the past, they can now run on Cori by checkpointing/restarting with MANA. MANA also makes running variable-time atomic relaxation jobs much easier by allowing more predictable checkpoint overhead, and by checkpointing/restarting at any point of execution. MANA also helps save machine time significantly for some long running VASP jobs, which spike in memory usage in their final computation stage. To accommodate this extra memory usage, these jobs currently have to be run on a larger number of nodes using a reduced number of cores per node to allow more memory per task. With MANA, these jobs can be run on a smaller number of nodes and checkpoint before the memory spike, then restart with a larger number of nodes to complete the computation. MANA can thus significantly reduce machine time usage for these jobs.

A third showcased application is Gromacs [19] - a widely used molecular dynamics simulation code. Gromacs is written in C++ and parallelized with MPI+OpenMP. While Gromacs has internal C/R, a benefit of MANA is that a Gromacs computation can be checkpointed at any point in its execution and resumed to generate exactly the same results as an uninterrupted run. MANA's C/R transparently saves all states, including random seeds. This makes it an ideal tool to restart chaotic MD simulations, for which trajectories diverge rapidly with even slight changes in restart data.

The talk concludes with future work. These include: community building, e.g., hosting a symposium on checkpointing for supercomputing [20]; further user training on MANA; deploying a preempt queue for real-time workloads; extending C/R with MANA to be robust at all scales (e.g., 1024 MPI ranks and more); and planning for the next NERSC supercomputer, Perlmutter, an NVIDIA GPU system. Concerning support for Perlmutter, we are planning to integrate some recent work on transparent checkpointing of CUDA, using the CRAC plugin for DMTCP [21], into NERSC's C/R environment.

Of special interest for future work is the goal of optimizing I/O for large checkpoint images at all scales. With the onset of checkpoint images of about one terabyte (e.g., SPAdes), we are exploring various I/O options to reduce the time required to write such large images to stable storage. We will also work on making the checkpoint overhead at large scales more manageable by exploring heterogeneous storage architectures and libraries, e.g., [22]. Currently, tests with HPCG using 512 MPI ranks and 8 threads per rank on Cori Haswell show that generating a checkpoint image of 5.8 TB takes 640 seconds on Cori's Lustre file system, but 30 seconds using the Burst Buffer. Restarting from this 5.8 TB checkpoint image file takes 110 seconds on the Lustre file system, but 39 seconds using the Burst Buffer [23].

ACKNOWLEDGMENTS

The authors would like to thank Rohan Garg at Nutanix, Twinkle Jain, and Prashant Chouhan at Northeastern University, Harsh Khetawat at North Carolina State University, and Tiffany Connors, Stephen Leak, and Christopher Samuel at NERSC. Without their work, it would not be possible to have made progress thus far toward enabling C/R in NERSC production workloads.

REFERENCES

[1] R. Garg, G. Price, and G. Cooperman, "MANA for MPI: MPI-Agnostic Network-Agnostic Transparent Checkpointing", Proc. of 28th Int. Symp. on High-Performance Parallel and Distributed Computing (HPDC'19), June 2019, Pages 49-60, https://doi.org/10.1145/3307681.3325962.

[2] BLCR. [Online]. Available: https://crd.lbl.gov/departments/computer-science/class/research/past-projects/BLCR/.

[3] DMTCP team, DMTCP code. [Online]. Available: http://dmtcp.sourceforge.net/.

[4] DMTCP website for NERSC users. [Online]. Available: https://docs.nersc.gov/development/checkpoint-restart/.

[5] Tiffany Connors, Rebecca Hartman-Baker, Zhengji Zhao, and Stephen Leak, Variable-time job scripts. [Online]. Available: https://github.com/NERSC/variable-time-job

[6] Variable-time job scripts. [Online]. Available: https://docs.nersc.gov/jobs/examples/#variable-time-jobs.

[7] Hands-on user training on variable-time job scripts for VASP users, June 2020. [Online]. Available: https://www.nersc.gov/users/training/events.

[8] Hands-on user training on variable-time job scripts, May 2020. [Online]. Available: https://www.nersc.gov/users/training/events.

[9] User training on DMTCP for users running serial and threaded applications, November 2019. [Online]. Available: https://www.nersc.gov/users/training/events.

[10] Hands-on user training on variable-time job scripts for VASP users, June 2019. [Online]. Available: https://www.nersc.gov/users/training/events.

[11] Zhengji Zhao, "Variable-time Jobs", a presentation in the Monthly NERSC User Group (NUG) Webinars, May 2018. [Online]. Available: https://www.nersc.gov/users/NUG/teleconferences/nug-webinar-may-17-2018/.

[12] Rohan Garg, Gregory Price, Prashant Chouhan, and Gene Cooperman, MANA code. [Online]. Available: https://github.com/mpickpt/mana.

[13] VASP. [Online]. Available: https://www.vasp.at/.

[14] Benjamin Driscoll, and Zhengji Zhao, "Automation of NERSC Application Usage Report", Proc. of Seventh Annual Workshop on HPC User Support Tools (HUST 2020), November 11, 2020.

[15] HPCG. [Online]. Available: https://www.hpcg-benchmark.org/.

[16] Cori, a Cray XC40 system at NERSC. [Online]. Available: https://docs.nersc.gov/systems/cori/.

[17] SPAdes. [Online]. Available: https://cab.spbu.ru/software/spades/.

[18] Jie Wang, Alicia Clum and Alex Copeland. Internal communications.

[19] Gromacs. [Online]. Available: http://www.gromacs.org/.

[20] First International Symposium on Checkpointing for Supercomputing (SuperCheck21). [Online]. Available: https://ckpt-symposium.lbl.gov

[21] Twinkle Jain and Gene Cooperman, "CRAC: Checkpoint-Restart Architecture for CUDA with Streams and UVM", Proc. of Int. Conf. for High Performance Computing, Networking, Storage and Analysis (SC'20), IEEE Press, to appear, (and earlier technical report at https://arxiv.org/abs/2008.10596)

[22] Nicolae, B., Moody, A., Gonsiorowski, E., Mohror, K. and Cappello, "VeloC: Towards High Performance Adaptive Asynchronous Checkpointing at Large Scale", The 2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS'19), pp. 911–920, Rio de Janeiro, Brazil, 2019. [Online] Available: https://doi.org/10.1109/IPDPS.2019.00099

[23] Harsh Khetawat, *et al.*, "Scale-up study with DMTCP/MANA on Lustre File System and Burst Buffer". Manuscript in preparation (Aug., 2020).