Avida Checkpoint/Restart Implementation

Nilab Mohammad Mousa: McNair Scholar

Dirk Colbry, Ph.D.: Mentor

Computer Science

Abstract

As high performance computing centers (HPCC) continue to grow in popularity, issues of resource management and reliability are becoming limiting factors in scheduling long-running jobs. One of the factors is time. Efficacy of Avida, the platform of research for digital evolution, is limited by the allocated time available on HPCC. Saving the state of the software at a point and restarting the execution from the paused state will allow for better resource utilization, as well as help reduce restrictions that constrain the potential of the software. This research examines whether or not Berkeley Lab Checkpoint/Restart (BLCR) can be integrated with the run script that is used for Avida. Adding checkpoint/restart functionality to Avida will have multiple impacts on, effectiveness of the software and future research.

Introduction

Increasingly, scientists use computer experiments and simulations to study real-world models and systems. Computer simulation software has significantly altered the way research is being conducted. Often, in science, technology, engineering, and mathematics fields, scientists use computer modeling when the real-world system is not feasible to study. For example, when studying evolution, scientists need to overcome the obstacle that evolution happens extremely slow [3]. Considering this fact, related experiments are conducted only with organisms that can undergo many generations in a short amount of time [4]. At the Bio-Computational Evolution in Action Consortium (BEACON) Center, researchers utilize Avida software, an artificial life platform, to study evolutionary biology.

The digital Avida organisms, Avidians, are capable of evolving and replicating very quickly. In a computer, it is easy to measure any number of generations one might be interested in studying with arbitrary precision. However, the number of generations and the time it takes to propagate organisms for several thousand generations is limited by the processing power available [4]. At Michigan State University the High Performance Computer Center (HPCC) only provides, at most, one week of continuous computational power for a job running on the cluster. In other words, if an experiment requires a time frame of more than one week or 168 hours of wall-time, HPCC will not be able to provide the resources necessary.

When one week of computational time is not sufficient it, it seems obvious that researchers would want to save the state of the software and restart from the same point for another period of time, repeating the process as necessary. However, Avida software cannot save its state. In other words, the application does not have comprehensive checkpoint/restart feature implemented. The goal of this research project is to improve computational tools that scientists and researchers use to study evolutionary biology models by adding checkpoint/restart functionality to Avida software.

Background

In this research we examine whether or not Berkeley Lab Checkpoint/Restart (BLCR) can be integrated with the dist_run program that is used to schedule Avida jobs. This section will attempt to explain these modules.

Checkpoint/restart

Checkpoint/Restart mechanism saves the entire state of a program to a file and allows the program to be stopped and resumed from that file on the same system or similar system. For applications, productive checkpoints can reduce or even eliminate loss of work in case of hardware or software failure.

Berkeley Lab Checkpoint/Restart (BLCR)

Berkeley Lab Checkpoint/Restart (BLCR), developed by Lawrence Berkeley National Laboratory, is a loadable kernel module (LKM) that allows checkpoint and restart of Linux processes. LKM is an object file that extends the base kernel of the operating system to add support for new hardware, file systems, or for adding system calls. Checkpoint/Restart is vital for long-running jobs on high-performance clusters because it aids computations with fault tolerance and reliability. BCLR has grown in popularity for its versatility and ease of integration with other frameworks, since its first release in 2005. Since then twenty-five more versions were released with added support to checkpoint wide range of applications [2]. One of the early projects that utilized BLCR as a back-end support for checkpoint/restart systems was the Local Area Multicomputer (LAM)/MPI, project (LAM/MPI is a high-performance, open source implementation of the Message Passing Interface (MPI) standard that is researched, developed, and maintained at the Open Systems Lab at Indiana University [1]). BLCR kernel level process checkpoint system was integrated LAM/MPI to design a system for providing coordinated checkpointing and rollback recovery for MPI-based parallel applications, allowing the system to be used for cluster maintenance and scheduling reasons as well as for fault tolerance [6]. For applications that can be checkpointed, productive checkpoints can reduce or even eliminate loss of work in case of hardware or software failure.

dist_run

dist_run is a program written by David Bryson, a digital evolution research specialist at the BEACON Center at Michigan State University. Researchers use dist_run as a submission script to Avida software when it needs to be run on multiple computer systems. The dist_run tool is widely used by researchers at Michigan State University and is integrated deeply in the research groups' workflow. The disadvantages of using dist_run for implementation of checkpoint/restart is that the tool is not up-to-date with advanced job scheduler functionality, such as the those provided by Terascale Open-Source Resource and QUEue Manager (TORQUE), adopted by HPCC at Michigan State University. However, we will be working on modernizing the dist_run tool as we append checkpoint/restart functionality.

Methods and Design

The advantage of using Berkeley Lab Checkpoint/Restart for us is that it is has been successfully built, installed, and configured on HPCC and is capable of providing checkpoint/restart support to shell script running scientific workloads on wide variety of networks. The module easily fits production systems due to the fact that the source code of application and binaries does not require modification or special compile/link in most cases. Although statically linked executables need linking with BLCR's 'libcr_run' library to secure that the application contains some library code that BLCR provides, dynamically linked executables only require to be started with cr_run command to load the BLCR library into the application at startup time. Since Avida is dynamically linked cr_run can easily ensure that Avida will be checkpointed and restarted with minimum effort. After a careful evolution of dist_run program it was modified to include BLCR commands with the aim of adding checkpoint/restart functionality to Avida. Figure 1 demonstrates post modification flow of dist_run program execution.

Testing

Modifications of the program were tested considering the four test cases below.

1. Submitted job does not contain a checkpoint file and runs longer than wall-time. The program will need to checkpoint at the end of wall-time and resubmit to continue computation.

2. Submitted job does not contain a checkpoint file and runs less than wall-time. The program does not need to checkpoint at the end of wall-time and needs to exit.

3. Submitted job contains a checkpoint file and runs longer than wall-time. The program will need to checkpoint at the end of wall-time and resubmit to continue computation.

4. Submitted job contains a checkpoint file and runs less than wall-time. The program does not need to checkpoint at the end of wall-time and needs to exit.

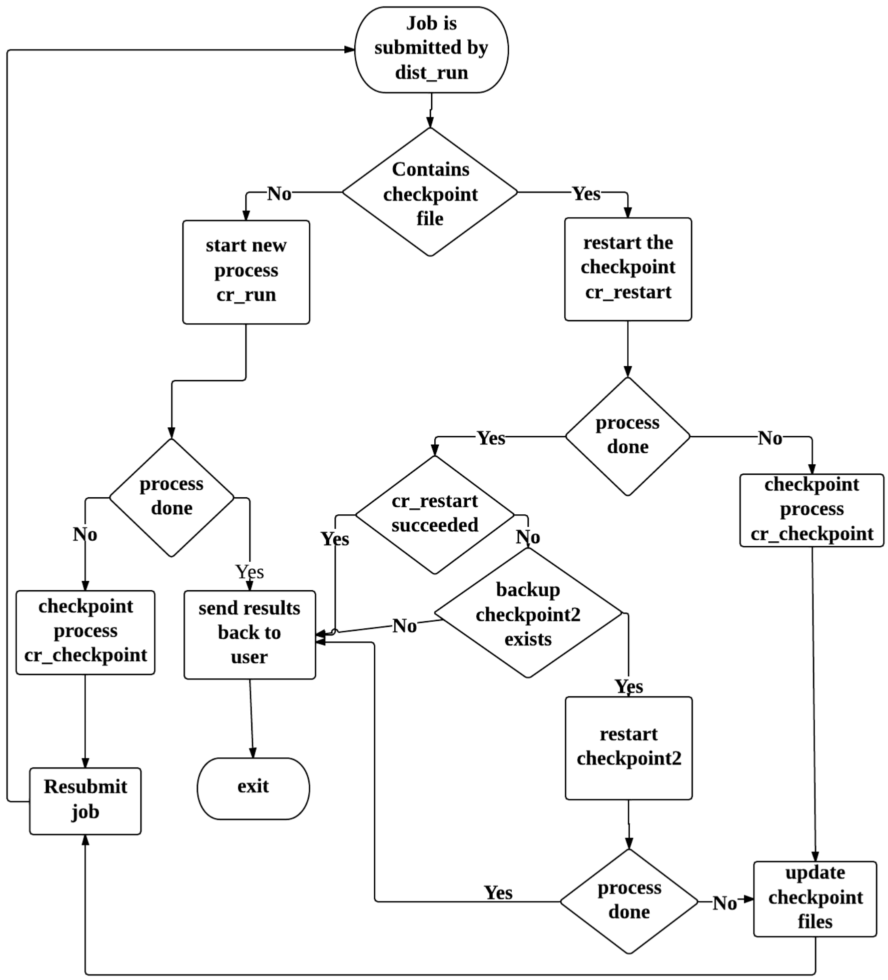

Figure 1: Integration of BLCR with dist_run program

Results and Discussion

Scientists use Avida as an evolutionary computing platform to propagate generations of digital organisms. In this research, checkpoint/restart was successfully implemented for Avida software by integrating BLCR and dist_run program.

Impacts on digital evolution research

As a result of the integration of BLCR and dist_run, Avida jobs can:

- recover from hardware failures
- pause and restart from the point they left off
- start on one hardware platform and finish on another

Impacts on high performance computing center

In this research we were able to overcome the shortcomings of time-restricted computing systems allowing HPCC to:

- provide computing resources for long-running jobs
- utilize resources allocated for short-running jobs
- increase reliability and fault tolerance

Adding checkpoint/restart functionality to Avida has multiple impacts on the research; long-running jobs (weeks or months) are able to run on time restricted computing systems (hours or days). Long-running jobs are more robust to system failures, and interesting future digital evolution research can be conducted, given, availability of computing resources.

Conclusion

This research presented a checkpoint/restart implementation for Avida jobs that has been implemented in dist_run tool using BLCR as the checkpointer. Integrating checkpoint/restart feature with Avida enables researchers working with the software to disregard computational limitations and focus on other complexities of research.

Future Work

We will consider exploring other implementations of checkpoint/restart for Avida. BLCR is system level implementation and it would be beneficial to compare the module with library-based checkpointer. Exploring the following frameworks will grant us a better understanding of different modules.

- Condor
- Score
- CoCheck
- BPRoc

Acknowledgments

The author wishes to express her appreciation to and acknowledge the help and support of Dr. Dirk Colbry, Dr. David Bryson, and Dr. Judi Clarke.

References

[1] Lam/mpi users guide version 7.1.4. http://www.lam-mpi.org/download/files/7.1. 4-user.pdf, July 2007.

[2] Berkeley lab checkpoint/restart (blcr) for linux. http://crd.lbl.gov/groups-depts/ftg/projects/currentprojects/BLCR/, January 2013.

[3] Charles Darwin and William F Bynum. The origin of species by means of natural selection: or, the preservation of favored races in the struggle for life. AL Burt, 2009.

[4] Charles Ofria and Claus O Wilke. Avida: A software platform for research in computational evolutionary biology. Artificial life, 10(2):191-229, 2004.

[5] Eric Roman. A survey of checkpoint/restart implementations. In Lawrence Berkeley National Laboratory, Tech. Citeseer, 2002.

[6] Sriram Sankaran, Jeffrey M Squyres, Brian Barrett, Vishal Sahay, Andrew Lumsdaine, Jason Duell, Paul Hargrove, and Eric Roman. The lam/mpi checkpoint/restart framework: System-initiated checkpointing. International Journal of High Performance Computing Applications, 19(4):479-493, 2005.

[7] Chao Wang, F. Mueller, C. Engelmann, and S.L. Scott. A job pause service under lam/mpi+blcr for transparent fault tolerance. In Parallel and Distributed Processing Symposium, 2007. IPDPS 2007. IEEE International, pages 1-10, 2007.