PARALLEL I/O OPTIMIZATIONS FOR LARGE-SCALE SCIENTIFIC APPLICATIONS

By

Jing Fu

An Abstract of a Thesis Submitted to the Graduate

Faculty of Rensselaer Polytechnic Institute

in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Major Subject: COMPUTER SCIENCE

The original of the complete thesis is on file
in the Rensselaer Polytechnic Institute Library

Examining Committee:

Dr. Christopher D. Carothers, Thesis Adviser
Dr. Mark Shephard, Member
Dr. Kenneth Jansen, Member
Dr. Peter Fox, Member
Dr. Robert Ross, Member

Rensselaer Polytechnic Institute
Troy, New York

July 2012
(For Graduation August 2012)

ABSTRACT

Highly efficient and accurate modeling on advanced computing platforms will enable the relevant science and engineering communities to advance their understanding of complex systems that are too costly or risky for experimental study. However, as the number of processors increases to hundreds of thousands in the recent parallel computer architecture, the failure probability raises correspondingly, making fault tolerance a highly important yet challenging task.

Application-level checkpointing is one of the most popular techniques used to deal with unexpected failures proactively (as well as keeping data for post processing), due to its portability and flexibility. During the checkpoint phase, the local states of the computation spread across thousands of processors are saved to stable storage. Unfortunately, this approach results in heavy I/O load and can cause an I/O bottleneck in a massively parallel system.

In this dissertation, we examine a few parallel I/O approaches for two scientific applications: a computational fluid dynamics solver called *PHASTA*, and a massively parallel electromagnetics solver system called *NekCEM*. Both applications scale very well on IBM Blue Gene/L, Blue Gene/P and Cray XK6 supercomputers. We discuss a MPI-IO collective approach (coIO), an application-level I/O staging approach, called "reduced-blocking I/O" (rbIO) and threaded version of rbIO. We demonstrate their respective performance advantages over the traditional "1 POSIX file per processor" approach that is still being used by many scientific applications. We perform scaling tests on PHASTA and NekCEM, and our study shows that rbIO and coIO result in 100× improvement over previous 1PFPP approaches on up to 65,536 processors of the Blue Gene/P using the GPFS. Our study also demonstrates a 25× production performance improvement for NekCEM. We also show how to tune various parameters for those parallel I/O approaches and how to validate performance analysis using I/O profiling techniques.

In addition, we discuss the use of threaded rbIO to achieve near-asynchronous I/O for *NekCEM* and greatly mitigate the blocking nature of checkpoint. Our ex-

periments on Blue Gene/P and Cray XK6 shows significant production performance improvement of NekCEM over the collective MPI-IO approach. We also discuss the factors that affect the speedup and expect those approaches to maintain their performance advantage on upcoming supercomputers (such as Mira and Blue Waters) as well as future exascale systems.