A FRAMEWORK FOR THE DYNAMIC RECONFIGURATION OF SCIENTIFIC APPLICATIONS IN GRID ENVIRONMENTS

By

Kaoutar El Maghraoui

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. Carlos A. Varela, Thesis Adviser
Dr. Joseph E. Flaherty, Member
Dr. Ian Foster, Member
Dr. Franklin Luk, Member
Dr. Boleslaw K. Szymanski, Member
Dr. James D. Teresco, Member

Rensselaer Polytechnic Institute
Troy, New York

April 2007
(For Graduation May 2007)

ABSTRACT

Advances in hardware technologies are constantly pushing the limits of processing, networking, and storage resources, yet there are always applications whose computational demands exceed even the fastest technologies available. It has become critical to look into ways to efficiently aggregate distributed resources to benefit a single application. Achieving this vision requires the ability to run applications on dynamic and heterogeneous environments such as grids and shared clusters. New challenges emerge in such environments, where performance variability is the rule and not the exception, and where the availability of the resources can change anytime. Therefore, applications require the ability to dynamically reconfigure to adjust to the dynamics of the underlying resources.

To realize this vision, we have developed the Internet Operating System (IOS), a framework for middleware-driven application reconfiguration in dynamic execution environments. Its goal is to provide high performance to individual applications in dynamic settings and to provide the necessary tools to facilitate the way in which scientific and engineering applications interact with dynamic environments and reconfigure themselves as needed. Reconfiguration in IOS is triggered by a set of decentralized agents that form a virtual network topology. IOS is built modularly to allow the use of different algorithms for agents' coordination, resource profiling, and reconfiguration. IOS exposes generic APIs to high-level applications to allow for interoperability with a wide range of applications. We investigated two representative virtual topologies for inter-agent coordination: a *peer-to-peer* and a *cluster-to-cluster* topology. As opposed to existing approaches, where application reconfiguration has

mainly been done at a coarse granularity (e.g., application-level), IOS focuses on migration at a fine granularity (e.g., process-level) and introduces a novel reconfiguration paradigm, *malleability*, to dynamically change the granularity of an application's entities. Combining migration and malleability enables more effective, flexible, and scalable reconfiguration.

IOS has been used to reconfigure actor-oriented applications implemented using the SALSA programming language and iterative process-oriented applications that follow the Message Passing Interface (MPI) model. To benefit from IOS reconfiguration capabilities, applications need to be amenable to entity migration or malleability. This issue has been addressed in iterative MPI applications by designing and building a library for process checkpointing, migration, and malleability (PCM) and integrating it with IOS. Performance results show that adaptive middleware can be an effective approach to reconfiguring distributed applications with various ratios of communication to computation in order to improve their performance, and more effectively utilize dynamic resources. We have measured the middleware overhead in static environments demonstrating that it is less than 7% on average, yet reconfiguration on dynamic environments can lead to significant improvement in application's execution time. Performance results also show that taking into consideration the application's communication topology in the reconfiguration decision improves throughput by almost an order of magnitude in benchmark applications with sparse inter-process connectivity.