Continuous Delivery of Research Application in a Distributed Environment

Literature Review

Sakhile Masoka (851667@students.wits.ac.za)
Witwatersrand University

May 1, 2015

Contents

Abstract

One of the aims of e-Infrastructure is to provide easy access to powerful computational and data platforms, to as many eligible users as possible. While user access is being simplified and made easy, the community of application developers and technical support in scientific collaborations does not yet have an easy way to integrate and manage applications. This gap has been identified and solutions to integrate applications into infrastructure in a fast, flexible, distributed and reproducible way have been developed by a few. This review presents some of the developed software systems to address this gap in e-Infrastructure.

1 Introduction

2 Literature Review

2.1 The Need For Automatic Application Deployment

Many scenarios push for an automated application deployment in a distributed environment. Gos [2004] in an expiriment to develop and implement DistAnt stated two main challenges the software had to solve. One is Grids are managed by different organisations, which commonly implies different resources administration and operating procedures, this complicates application deployment due to different resource acces,user permission, accounts and installation procedures. Secondly, the Grid is has a range of architectures (x86, ia64, sparc, MIPS) and operating systems (varying flavors of Linux, SunOS and IRIX). Technical heterogeneity complicates application deployment due to different build and executable files, library dependencies, file systems and installation procedures.

Hou [2009] added to the challenges that Applications software must be deployed and managed remotely, often without the benefit of interactive login capabilities, with limited resource authority in different autonomous administrative domains. Secondly, manual deployment and management of application software packages usually require expertise about both the underlying system platforms and the application. Lastly, manual deployment and management is error prone and does not scale to large grids.

Slawinski and Sunderam [2014] in making a case for an automated provisioning system, stated that scientists tradictionally execute their applications on computational clusters located at datacenters and rely on expect knowledge if they encounter troube with software deployment and runtime, but advancement in computer science and technology allows for more options into multi/many-core workstations and computational clouds or affordable clusters on-premise if deployment could be made easy. As scientific applications are computational intensive, users should select the best possible target for a specific run of an application to achieve the best performance (e.g., some clouds offer computational accelerators such as GPGPUs, when other may provide high-throughput file systems). However,this flexibility requires simplification in switching between computational platforms, which leads to a need of an autonomous approach in scientific application deployment and improved model of execution.

2.2 Application Deployment Designs

2.3 Features or Advantages

2.4 Scalability

3 Implementation Results

4 Conclusion

References

[Gos 2004] *Distributed Ant: A System to Support Application Deployment in the Grid*, 2004.

[Hou 2009] *ADEM: Automating deployment and management of application software on the Open Science Grid*, 2009.

[Slawinski and Sunderam 2014] J Slawinski and V. Sunderam. Autonomic multi-target deployment of science and engineering hpc applications. pages 180–189, 2014.