University College London
MSc Data Communications, Networks & Distributed Systems

Deployment in Computational Distributed Grids

Executive Summary

supervisor
Wolfgang Emmerich, Ph. D.

group members

Daisy Kong
Vesso Novov
Dimitrios Tsalikis
Stefanos Koukoulas
Thomas Karampaxoglou

06 September 2004

Executive Summary

Deployment in Computational Distributed Grids

Overview

Grid service deployment is a complex process that covers all post-development activities, such as installation, configuration and ignition of the service as well as of the infrastructure and services it depends on, de-installation, etc. The fact that the financial application[1], which we will be dealing with in this project, is computationally demanding implies the deployment of the grid services, which perform the Monte Carlo simulations, on a large collection of computational resources. However, the manual deployment is time-consuming, cumbersome and error-prone.

Motivation

What would it mean to an organization to be able to speed up the deployment of grid services while maintaining the process reliable and reducing the risk of misconfiguration due to inconsistencies in configuration files or human error? Moreover, what if failures during the deployment process were detected and appropriate actions were taken automatically? It is apparent that the human resources needed and the time spent to carry out all the deployment activities would have been minimized dramatically. As would the down time of the services, the disruption of the business process, and the amount of computational and communication resources used. All these are welcomed prospects by organizations.

Project aims and scope

This project aims to investigate the use of the SmartFrog deployment framework in conjunction with the emerging Open Grid Services Architecture (OGSA) to facilitate the rapid deployment of computationally intensive applications on compute clusters. The final product will enable the automatic deployment not only of a financial grid application on a large collection of computational resources, but also of the

[1] Developed by Charaka Goonatilake as part requirement for the BSC Degree in Computer Science.

infrastructure it depends on. Moreover, it will make it possible to perform various management tasks, such as re- and un-deployment, on the deployed services and infrastructure, as well as detect failures during the process and take the appropriate actions. Finally, an assessment will be made as to whether the increase, if any, in deployment speed is significant enough to offset the added time and effort needed to learn the Smart Frog technology.

The solution

The group has developed an application that enables the user to give as input, with the aid of a GUI, the name of the grid service(s) to be deployed and the physical locations of the hosts participating in the computational grid. The application displays the available resources of particular interest, such as CPU load and free hard disk space, of each and every host so that the user can determine whether they are sufficient. Given the sufficiency of resources on a host, deployment can begin with the click of a button. In the case that the grid infrastructure (GT3, Tomcat, etc) is not in place already, the user has the option to deploy it first. Special care is taken that infrastructure dependencies are available on that particular host. Then, the grid service(s) is fetched in the form of a GAR[2] file and is deployed on Tomcat with the aid of Ant. It should be noted that all the configuration activities are carried out automatically by the SmartFrog deployment framework with configuration parameters provided by the user through a description file. The application also makes it possible to perform various management tasks on the deployed services and infrastructure, such as re- and un-deployment. Moreover, it displays to the user failures that occurred during the deployment process and it is programmed to take appropriate actions.

Results

It is concluded that the SmartFrog framework provides all the essential facilities for building applications that enable the rapid deployment of grid services on computational resources. Moreover, one can acquire knowledge on SmartFrog with a relatively steep learning curve, which means that the time invested in learning the technology is much lower than the deployment acceleration gained in the long term.

[2] A GAR file contains the service(s) implementation, deployment description (wsdd), and the service interface definition (wsdl).