Self-Deployment and Self-Configuration of Pervasive Network Services

Huoping Chen, Salim Hariri, Byoung Kim, Yeliang Zhang
Autonomic Computing Laboratory
ECE Department, University of Arizona
{hpchen, hariri, byoung, zhang}@ece.arizona.edu

Mazin Yousif
Intel Corporation, USA
mzain.s.yousif@intel.com

Abstract

In this paper, we present a policy-based approach to achieve self-deployment and self-configuration of pervasive systems and services. In our approach, we decompose applications into a set of interconnected components. With each component, we specify the autonomic attributes required to self-configure and self-manage the operations of the components in Autonomic Component Architecture (ACA) such that the overall objectives of the applications are maintained at runtime. We use the self-deployment and configuration engines of our Autonomic Computing Environment, Autonomia, to demonstrate and validate our approach. We have demonstrated how Autonomia to automatically deploy and configure the distributed computing application of Linear Equation Solver.

1. Introduction

The control and management of large-scale applications over a heterogeneous and dynamic execution environment present challenging research problems. A possible solution can be autonomic computing [1].

2. Autonomic Component Architecture (ACA)

The Autonomic Component Architecture (ACA) defines the component architecture that can deal with complexity, dynamism, and heterogeneity. Furthermore, ACA model enables user to express the control policies and the optimization algorithms to make these components autonomic.

Autonomic Component Data Structure:

$AC = <C_{spec}, C_{port}, C_{stat}, C_{act}, C_{pol}>$

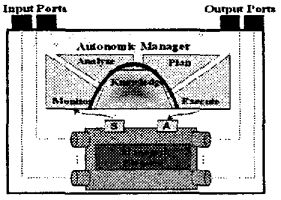

Figure 1: Autonomic Component Model

3. Autonomia – An ACA Framework

Autonomia environment [2] provides application developers with all the tools required to specify the appropriate control and management schemes, the services to deploy and configure the required software and hardware resources, and to run applications.

4. Self-Deployment & Self-Management Services

4.1 Policy Engine

The Autonomia policy is expressed in ***IF-condition-THEN*** format. For example, "*IF CPU load is HIGH, THEN stop running the component.*

4.2 Self-Deployment Service

Self-Deployment service is to automate deployment of applications. The application components and how they interact are described using a workflow and it can be expressed into Autonomia self-deployment policies.

4.3 Self-Management Algorithms

Self-Management is to maintain the application requirements at runtime. To achieve this, we have several autonomous runtime engines that each is assigned to control and manage one property (self-configuration, self-heal, self-optimization, etc.).

5. Experiment

We demonstrated how to automate the deployment and execution of the Linear Equation Solver (LES) application in Autonomia environment. The problem size is 1024×1024 and it runs on a cluster of high-performance computers at the University of Arizona.

6. Conclusion

We described an Autonomic Component Architecture that can support autonomic behavior of the components and the applications composed of these components. We have successfully implemented self-deployment and self-configuration services and demonstrated a distributed computing application example in our ACA framework, Autonomia.

References:

[1]. http://www.research.ibm.com/autonomic/
[2]. Hariri, S. etc., Autonomia: an autonomic computing environment, Conference Proceedings of the 2003 IEEE IPCCC, 9-11 April 2003.