Real-time Fault Tolerant Deployment and Configuration Framework for Cyber Physical Systems

Subhav Pradhan, Aniruddha Gokhale, William R. Otte, Gabor Karsai
Dept of Electrical Engineering and Computer Science
Vanderbilt University
Nashville, TN 37235, USA
{subhav.m.pradhan, a.gokhale, w.otte, gabor.karsai}@vanderbilt.edu

ABSTRACT

This paper describes ongoing work on making the deployment and configuration functionality for cyber physical systems reliable and tolerant to failures, while also supporting predictable and incremental online redeployment and reconfiguration of application functionality. Our work is currently designed and evaluated in the context of a system of fractionated spacecrafts, which is a representative CPS system.

Categories and Subject Descriptors

C.4 [**Computer-Communication Networks**]: Distributed Systems—*Fault tolerant software deployment*

General Terms

Reliability, Software Deployment

Keywords

CPS, deployment and configuration, reliability

1. INTRODUCTION

In cyber physical systems (CPS), such as air traffic management, distributed smart grid, and fractionated spacecrafts, deployment and configuration (D&C) of application components to physical resources of the system must happen in a timely and predictable manner. Although timely and predictable D&C is an important requirement for mission-critical CPS, at least two additional key requirements are expected of any D&C capabilities for CPS. First, these CPS applications require that the D&C capabilities themselves be reliable and tolerant to failures where recovery from failures is timely and predictable, and minimizes adverse impact on CPS quality of service (QoS) requirements. Second, due to changes in the mode of operation or fluctuations in resource availability, mission-critical CPS often must undergo redeployment and reconfiguration including incremental D&C, which must be supported predictably within any D&C capability designed for the CPS.

Our prior work on a D&C engine called LE-DAnCE (Locality Enabled Deployment and Configuration Engine) [3, 1] has focused on making application deployment and configuration predictable and timely for large-scale mission-critical cyber applications. However, our prior work did not consider reliability and predictable fault tolerance as well as incremental runtime (re)deployment and (re)configuration. Moreover, our prior work also did not consider extending the functionality to support CPS applications.

2. ONGOING WORK AND CHALLENGES

Our ongoing work focuses on supporting these key requirements within the LE-DAnCE framework and is evaluated in the context of fractionated spacecrafts [2], which is a representative CPS that brings new challenges to an already hard set of problems. For example, in our prior work on LE-DAnCE we had assumed stable networks. For fractionated spacecrafts, the available network bandwidth changes with changing distance between spacecrafts and also the positioning of the spacecrafts and their orbits. Furthermore, space hardware is inherently constrained in resources, which means spacecrafts will comprise limited computing power and memory in addition to the need to conserve power. Thus, significant optimizations are needed to existing D&C schemes to address these limitations.

Currently we are focusing on implementing a pluggable verification mechanism within the LE-DAnCE framework. These plugins are decoupled from the common framework, and help to ascertain if correctness and QoS criteria for the specific application is satisfied.

3. REFERENCES

[1] G. Deng, J. Balasubramanian, W. Otte, D. Schmidt, and A. Gokhale. Dance: A qos-enabled component deployment and configuration engine. *Component Deployment*, pages 67–82, 2005.

[2] A. Dubey, W. Emfinger, A. Gokhale, G. Karsai, W. Otte, J. Parsons, C. Szabó, A. Coglio, E. Smith, and P. Bose. A software platform for fractionated spacecraft. In *Aerospace Conference, 2012 IEEE*, pages 1–20. IEEE, 2012.

[3] W. Otte, A. Gokhale, and D. Schmidt. Predictable deployment in component-based enterprise distributed real-time and embedded systems. In *Proceedings of the 14th international ACM Sigsoft symposium on Component based software engineering*, pages 21–30. ACM, 2011.