Distributed Systems Testing

Anna Liu\(^1\), Paddy Nixon\(^2\)

\(^1\) Advanced Distributed Software Architecture and Technologies Group, CMIS, CSIRO, Sydney Australia, ph.(+61 2) 9325 3237, fax.(+61 2) 9325 3200, Anna.Liu@cmis.csiro.au

\(^2\) Department of Computer Science, Trinity College, Dublin 2, Ireland, ph.(+353 01) 608 2666, fax.(+353 01) 677 2204, Paddy.Nixon@cs.tcd.ie

1. Motivation

With existing robust middle-ware and powerful multi-processor desktop machines, more distributed and multi-threaded programs are being written to exploit the networked multi-processing power. Although component-based software engineering has been widely employed to handle this complexity, testing such a system is still difficult. In addition to the traditional testing problems, in the distributed application domain, there are also many complex issues relating to concurrency that needs to be considered. Phenomena such as resource and communication deadlock, live-lock, starvation and race conditions must be thoroughly tested for, before the system can be launched into production.

Distributed systems testing is an important and emerging area, and there is a need for an appropriate forum to foster the exchange of ideas and experiences as well as the publication of new research results. While industry is racing towards adopting distribution as integral to their needs and products, there is a dearth of products in the distributed systems testing arena. Further, current testing strategies used for distributed systems are black box based and do not address the intricacies of integration. They assume the no-side-affect rule of testing which does not apply to distributed systems. Take for instance the use of Netscape, or any product that relies on a stream based protocol for communication, in an environment which support 802.11 ethernet (wireless ethernet). Connections can arbitrarily drop and reform. Applications do not cater for this and the testing regimes do not test for this.

The theme of this mini-track is Distributed Systems Testing. The mini-track aims to provide a forum for the exchange of information and publication of the latest technological and theoretical advances in testing for distributed systems. This mini-track focuses on the problems that are unique to the software engineer and tester developing and testing distributed systems. Further, this mini-track explores the key technical aspects of software testing methods and tools in the distributed systems domain, both in research and in practice.

2. Contributions

Five papers have been selected for this minitrack. They are printed here in the order of oral presentation at HICSS-33. These papers have been chosen through a rigorous reviewing process, and as a whole present the many challenges and new advances in the field of distributed systems testing.

Specifically, these selected papers make contributions in the following areas:

- Testing tools and environment
- Large scale testing experiences
- Formal methods
- Design for testing

The testing of complex distributed systems is clearly a difficult and important problem, as evidenced by the number of quality papers submitted to this minitrack. The five papers presented here make significant contributions to alleviating these problems, and provide a strong foundation from which future quality research in this area may germinate.