Model-driven Generative Framework for Automated OMG DDS Performance Testing in the Cloud

Kyoungho An, Takayuki Kuroda and Aniruddha Gokhale
ISIS, Vanderbilt University
{kyoungho, kuroda, gokhale}@isis.vanderbilt.edu

Sumant Tambe and Andrea Sorbini
Real-Time Innovations
{sumant, sorbini}@rti.com

Motivation

- The Object Management Group's (OMG) Data Distribution Service (DDS) provides many configurable policies which determine end-to-end quality of service (QoS) of applications

Solution

- To overcome this problem
  - Design-time formal methods
  - But ... lack of sufficient accuracy in prediction, tool support, and understanding of formalism
- A promising approach
  - To emulate system behavior and gather data on the QoS parameters of interest by experimentation
- Automated performance testing framework called AUTOMATIC (AUTOmated Middleware Analysis and Testing In the Cloud)
- Activity Domains
  - User - Modeling, Monitoring
  - Test Automation System - Test Planning, Test Deployment
  - Cloud Infrastructure - Test Emulation

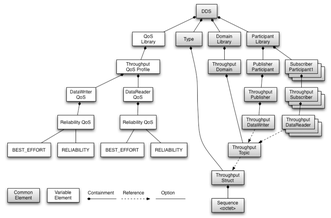


Figure 3. XML-based DDS Application Tree

Test Deployment

- To deploy the XML-based DDS testing applications, specifications related to the deployment are generated by the model interpreter
- Our deployment tool
  - Deploys the XML-based DDS testing applications in a cloud platform
  - Executes remotely the applications with a emulation tool called RTI Prototyper

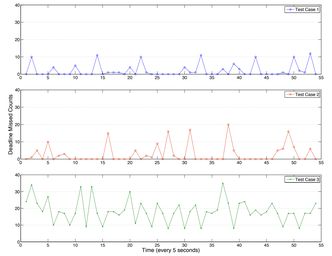


Figure 6. Deadline Miss Counts for Different Reliability QoS Settings

DDS QoS

QoS Policy	QoS Policy	
DURABILITY	USER DATA	User QoS
HISTORY	TOPIC DATA	
READER DATA LIFECYCLE	GROUP DATA	
WRITER DATA LIFECYCLE	PARTITION	Presentation
LIFESPAN	PRESENTATION	
ENTITY FACTORY	DESTINATION ORDER	Redundancy
RESOURCE LIMITS	OWNERSHIP	
RELIABILITY	OWNERSHIP STRENGTH	Transport
TIME BASED FILTER	LIVELINESS	
DEADLINE	LATENCY BUDGET	
CONTENT FILTERS	TRANSPORT PRIORITY	

Volatility, Infrastructure, Delivery

Reference from http://www.omg.org/news/meetings/workshops/RT-2007/00-TS_Hunt-revised.pdf

- To realize this approach ...
  - Model-based automatic performance testing framework with generative capabilities
  - Reduce manual efforts in generating a large number of relevant QoS configurations and deploying and testing applications on a cloud platform

Domain-Specific Modeling Language

- We developed a DSML using the Generic Modeling Environment (GME)
- Modeling a DDS application for emulation and testing its performance for various combinations of DDS QoS policies

Figure 2. Example Domain-Specific Model of DDS Throughput Testing Application

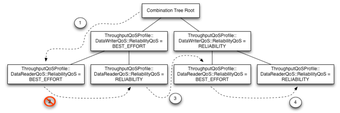
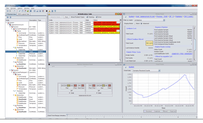


Figure 4.Variable Element Combination Tree

Test Monitoring

- We employed a tool to visualize monitoring data of applications called RTI Monitor
- It helps users to understand DDS systems easily via graphical interfaces and to verity behaviors of entities as expected



Reference from http://www.rti.com/company/news/rti-monitor.html

Conclusion

- Our work combines model-driven engineering (MDE) and generative programming techniques to provide a tool called AUTOMATIC
- Current artifacts in AUTOMATIC are available for download from www. dre.vanderbilt.edu/ ~kyoungho/AUTOMATIC

Challenge

- It is challenging
  - To predict the system's performance in terms of latencies, throughput, and resource usage
- Because ...
  - Diverse combinations of QoS configurations influence QoS of applications

Framework Architecture

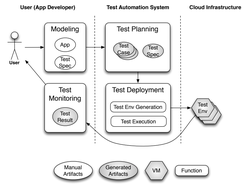


Figure 1. Framework Architecture

Test Plan Generation

- The Test Planning function
  - Traverses modeled elements in a model instance via a model interpreter
  - To generate executable applications and related test specification files

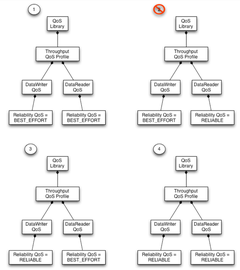


Figure 5.Variable Element Tree

Experience

- The experiment evaluates performance of an example DDS application by combining the RELIABLITY, HISTORY, and DEADLINE QoS policies
- Test Environment
  - OpenStack based cloud test best employing KVM as a hypervisor
  - Each VM type has 1 vCPU and 512 MB
  - DEADLINE QoS: 1millisecond
  - HISTORY QoS: KEEP_ALL

References

Patrick Th. Eugster, Pascal A. Felber, Rachid Guerraoui, and Anne- Marie Kermarrec. The many faces of publish/subscribe. *ACM Com- puter Survey*, 35:114–131, June 2003.

1. Joe Hoffert, Douglas Schmidt, and Aniruddha Gokhale. A QoS Policy Configuration Modeling Language for Publish/Subscribe Middleware Platforms. In *Proceedings of International Conference on Distributed Event-Based Systems (DEBS)*, pages 140–145, Toronto, Canada, June 2007.
2. D. Jayasinghe, G. Swint, S. Malkowski, J. Li, Qingyang Wang, Junhee Park, and C. Pu. Expertus: A Generator Approach to Automate Perfor- mance Testing in IaaS Clouds. In *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*, pages 115–122, 2012.
3. Object Management Group. Data Distribution Service for Real-time Systems Specification, 1.2 edition, January 2007.