

Detecting Distributed Predicates Using Genetic Algorithms

Eslam Al Maghayreh, Computer Sciences Department, Yarmouk University, Irbid, Jordan

Iyad Abu Doush, Computer Sciences Department, Yarmouk University, Irbid, Jordan

Faisal Alkhateeb, Computer Sciences Department, Yarmouk University, Irbid, Jordan

ABSTRACT

Detection of distributed predicates is one of the techniques that have been used in the literature to improve the dependability of distributed programs. This technique (sometimes referred to as runtime verification) is used to verify that a given run of a distributed program satisfies certain properties (specified as predicates). In general, the detection of a distributed predicate can incur significant overhead due to the existence of multiple processes running concurrently. Several techniques have been introduced in the literature to efficiently detect distributed predicates. However, most of these techniques work efficiently for certain classes of predicates, like conjunctive predicates. In this paper, the authors have presented a technique based on genetic algorithms to efficiently detect distributed predicates under the possibly modality. The authors have used JGAP (Java Genetic Algorithms Package) to implement the algorithm and conducted several experiments to demonstrate its effectiveness.

Keywords: *Debugging, Distributed Predicates Detection, Distributed Programs, Runtime Verification, Genetic Algorithms, Testing*

1. INTRODUCTION

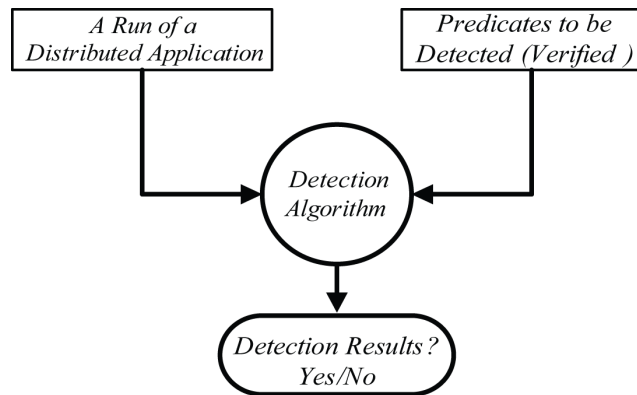
Improving the dependability of distributed programs is not an easy task. Many techniques were introduced in the literature to help in developing dependable distributed programs. A key technique which is used by several researchers is the detection of distributed predicates (Garg, 2002; Dumais & Li, 2002; Freiling & Jhumka, 2007; Chu & Brockmeyer, 2008; Garg & Waldecker, 1994; Garg & Waldecker, 1996). In this technique a specific run of a

given distributed program is verified against certain properties.

The main components of a distributed predicates detection framework are shown in Figure 1. Algorithms to detect distributed predicates can be used to verify that a particular implementation of a given distributed system satisfies some required properties. Other verification techniques, like model checking, work on a model of the system rather than a particular implementation of it. However, for some distributed applications, especially those employed in safety critical environments, it is important

DOI: 10.4018/jiit.2013010104

Figure 1. The main components of a distributed predicates detection framework



to reason about a particular implementation rather than an abstract model of the system. This is considered one of the main advantages of distributed predicates' detection techniques (Garg, 2002; Al Maghayreh, 2010).

The detection of distributed predicates has several applications, including the following:

- **Detection of when a distributed application reaches some stable state:** This includes conditions that once become true they will not turn false anymore. For example, deadlock detection and loss of token in a token ring.
- **Testing and debugging of distributed programs:** For example, when testing a leader election algorithm, we can monitor the system to detect the presence of multiple leaders.
- **Identifying bottlenecks:** In this case we are interested in locating positions during the execution of a distributed application when more than n processes blocked at the same time.
- **Symmetry in Communications:** Distributed predicates' detection techniques can be used to identify when a producer starts generating items faster than what the corresponding consumer can receive and process.

The detection of distributed predicates is a time consuming task due to the concurrency in distributed applications. In order to reduce the time to solve this problem, many methods have been introduced in the literature (Al Maghayreh, 2010; Garg, 2002; Chase & Garg, 1998). A brief review of these methods will be given in Section 4. However, most of these methods work efficiently for only certain classes of predicates. It has been proved that verifying whether a run of a distributed program satisfies certain property (predicate) or not is, in general, an NP-complete problem (Garg, 2002).

In this paper, a genetic algorithm is used to develop a more powerful technique for detecting distributed predicates. This technique can work efficiently for predicates under the possibly modality (a predicate under the possibly modality is evaluated to true if it is true in at least one global state (Garg, 2002)). A genetic algorithm is a popular artificial intelligent search technique that can be used effectively to solve problems with exponential size search space which is the case in distributed predicates' detection. In Section 5, we will give a brief introduction to genetic algorithms.

The remainder of the paper is organized as follows. In Section 2, we present a formal model of a run of a distributed program. Section 3 highlights the difficulties of distributed

13 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/article/detecting-distributed-predicates-using-genetic/75546

Related Content

Design of Efficient VLSI Architecture and Implementation of Power-Optimized Turbo Decoder for LTE Networks

Manjunatha K. N., Raghu N. and Kiran B. (2023). *Role of 6G Wireless Networks in AI and Blockchain-Based Applications* (pp. 131-166).

www.irma-international.org/chapter/design-of-efficient-vlsi-architecture-and-implementation-of-power-optimized-turbo-decoder-for-lte-networks/320329

An Intelligent Traffic Engineering Method over Software Defined Networks for Video Surveillance Systems Based on Artificial Bee Colony

Reza Mohammadi and Reza Javidan (2016). *International Journal of Intelligent Information Technologies* (pp. 45-62).

www.irma-international.org/article/an-intelligent-traffic-engineering-method-over-software-defined-networks-for-video-surveillance-systems-based-on-artificial-bee-colony/171440

A Brownian Agent Model for Analyzing Changes in a Nation's Product Space Structure

Bin Jiang, Chao Yang, Takashi Yamada and Takao Terano (2015). *International Journal of Intelligent Information Technologies* (pp. 52-71).

www.irma-international.org/article/a-brownian-agent-model-for-analyzing-changes-in-a-nations-product-space-structure/128839

Development and Evaluation of a Dataset Generator Tool for Generating Synthetic Log Files Containing Computer Attack Signatures

Stephen O'Shaughnessy and Geraldine Gray (2011). *International Journal of Ambient Computing and Intelligence* (pp. 64-76).

www.irma-international.org/article/development-evaluation-dataset-generator-tool/54448

Customer Satisfaction in Hospitality Marketing From a Technological Perspective

Iva Rani Das, ABM Shahidul Islam and Mohammad Badruddoza Talukder (2024). *Impact of AI and Tech-Driven Solutions in Hospitality and Tourism* (pp. 383-407).

www.irma-international.org/chapter/customer-satisfaction-in-hospitality-marketing-from-a-technological-perspective/354143