Volume 3, Issue 2, February 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Enhanced Reachability Testing Scheme in Game Theoretic Approach for Efficient Communication

S. Preetha
Research Scholar
Karpagam University
Coimbatore, India

Dr. M. Punithavalli
Director
Sri Ramakrishna Engineering College
Coimbatore, India

Abstract- Reachability testing is more dynamic and exhaustive. Recent work does not focus on communication aspect of concurrent programs which is most probably complex in multiple thread execution of parallel program. Due to this reason the problem of determining and reacting to fault detection has received a greater attention in today's era and supports only the homogenous application rather than heterogeneous web application. So, this paper presents a web based reachability testing model with the game theoretic approach encompassing a state of Nash equilibrium for both homogenous and heterogeneous applications. In addition to the proposed model it tracks the synchronization events on the concurrent program with better communication between multiple semaphores generated during the execution. Reachability testing is of two folds. First, reachability testing adopts a dynamic framework in which SYN-sequences are derived automatically and on-the-fly, without creating any static models. Secondly, reachability testing is an interleaving-free approach in which independent events are never totally ordered in a SYN-sequence. The proposal adopts game theoretic approach which supports parallel and diversifies functions by maintaining equilibrium state for contra functions. Simulations are conducted with web and game theoretic approach to evaluate the performance of proposed web based reachability testing model for both homogeneous and heterogeneous applications. The consistency of our proposal model shows reduction in Communication time, increased efficiency, minimal testing time i.e., nearly 13%, and space usage i.e. nearly 15% compared to the existing reachability testing methods.

Keywords: **Reachability Testing, Semaphore, Game Theoretic Approach, Synchronization, Web Application, Heterogeneity.**

I. INTRODUCTION

The classical mode of semaphore based reachability testing refer to reachability testing distributed across different nodes which addresses the communication efficiency between parallel task programs. Usually the states and transitions of the system under consideration are collected in a transition graph, and the only dynamic aspect is the change of state which is then captured by the concept of an execution path or of a multiplication tree. A theory of model-checking over energetic structures is still very much in its beginnings.Reachability testing of concurrent program includes a set of supportive threads or processes multithreaded programs and the threads are coordinated by admitting shared memory dispersed programs. This threads synchronization completed by switching over messages is pervasive in contemporary software development with enhanced resource exploitation by growing computing efficiency to definite problems in the non-deterministic behavior. In this respect, Reachability testing is complementary to specification-based testing, which decide on valid sequences from a requirement and conclude whether the sequences are allowed by the implementation.

We consider graphs as models of communication networks. Such a network is understood to change over time by way of deletion and formation of nodes. The obtainable resources travel during edges to existing nodes, they may at some node Y stores a deleted node Z by moving to it if there was an edge (Y, Z) in the network before the deletion of Z as in Fig. 1.1.

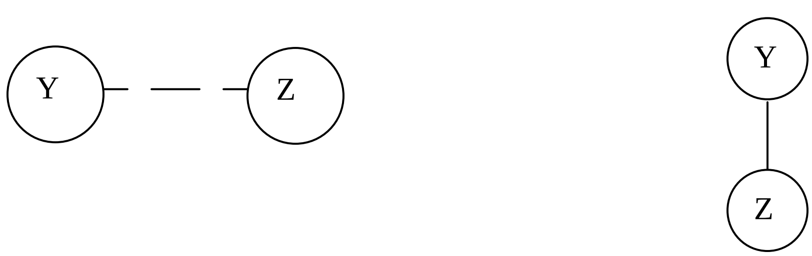


Fig.1.1: Movement of nodes

The similar way the formation of nodes Y1, Y2, Y3 can also be performed by picking out some set of strong nodes by creating a new node P and connect it with each of all the nodes in P by transition as shown in Fig. 1.2.

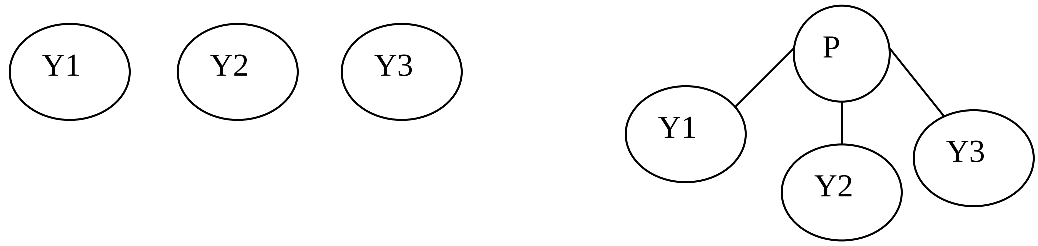
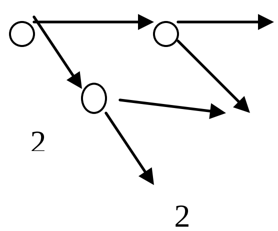


Fig.1.2: Creation of node

A fault is a deviation of the system structure or the system parameters from the nominal situation. This implies that after the occurrence of a fault the system will have a behavior which is different from the nominal one. Hence Fault detection is the property of respond to faults. In meticulous the investigation of fault tolerance consists in establishing if a given system is still able to achieve its tasks after the occurrence of a given fault, while the mixture of fault tolerance resides in providing a given system the tools to react to a given faulty situation.

A game consists of players, a set of moves available to those players and a condition of payoffs for each amalgamation of strategies. The games are represented in different forms. The extensive form (Fig 1.3) formed games with a time series of moves. Games here are played on like a tree structure. Each vertex (or node) tells a point of choice for a player. The nc 1 . (Fig 1.4) game is presented by a matrix which identifies the players, approaches, and pay-offs.



Player A

Player A

4,6	0,0
-1, -1	7,5

Player B

Fig 1.3: Extensive Form of game

Fig 1.4: Normal Form of game

To make the game theoretic approach more effective, in this work we are going to present a technique which accepts the set of functions with different kinds of tasks i.e., from heterogeneous environment. Applying reachability testing for a game theoretic approach in heterogeneous environment which accepts different sets of functions with different kinds of actions needs to be joined with an older version of program without affecting the older set of functions already detached in it.

II. LITERATURE REVIEW

A universal execution model for simultaneous programs that permit reachability testing is applied to some commonly used synchronization constructs. It guarantees the every partially ordered synchronization sequence exactly once with no saving any sequences that already keep fit [1]. An comprehensive CFG consists of a number of CFGs , one per each process, that are connected by their static organization structure. There are two primary problems with this policy [2]. First, paths selected from an extended CFG might be infeasible; Secondly, it may be hard to capture dynamic behaviors in a statically constructed graph. Consistency-based diagnosis [3] considers the faulty behavior as a contradiction between the actual and the nominal behavior of the system. It doesn't require the likely faults to be known, and it takings by reducing the statement on the behavior of each component in turn. If this eliminate the negation, the constituent is measured a candidate for rectification. More newly, applicability of discrete game theory to liability localization and routine repair of programs have been proposed in [4].

A hybrid game is a multiplayer structure where the players have both discrete and continuous moves and the game proceeds in a sequence of rounds. In every round each player chooses either a discrete or a continuous move among the available ones [5]. A hybrid game has been successfully applied to solve the controller synthesis problem for timed, hybrid automata and to the fault diagnosis problem for timed automata [6]. In our setting we model the fault diagnosis problem as a game flanked by two players, the environment and the diagnoser. The alternative strategy for selecting SYN-sequences is to derive the reachability graph of a program and then select test paths from this graph [7]. Every path selected from a reachability graph is certain to be possible, but this strategy suffers from the state explosion problem. State exploration techniques such as VeriSoft [8] are also developed for testing concurrent programs. These techniques are considered as model checking practical to programs, as an alternative of their stipulation or models.

A game theoretic approach [10], which can be used to identify the differences, arises in the game. A game theoretic approach has been applied in different applications [12]. The game theoretic approach is a mathematical model which gives a solution for distributed systems [11]. Game-theoretic approaches for distributed power control [9] in interference relay channels are also be processed under different schemes. Nondeterministic testing is perhaps the simplest approach to testing concurrent programs. The main problem with this approach is that executions are uncontrolled and, thus, some SYN-sequences may be executed many times, whereas others may never be executed. Techniques have been developed to increase the chances of exercising different SYN-sequences and, thus, the chances of finding faults when a program is repeatedly executed [13]. The first technique is actually a family of algorithms that compute persistent sets. This technique needs to identify independent transitions and consider "which operations on which communication objects each thread might execute in the future" [14]. The latter information can be obtained from a static analysis of the program or from the user; otherwise, it must be assumed that anything could happen in the future. The amount of reduction that is obtained depends on the accuracy of this information [15]. State exploration techniques such as VeriSoft [16-17] and others have also been developed for testing concurrent programs.

III. Methodology for Enhancing the Communication in Game Theoretic Approach

The Fig 3.1 depicts the architecture of Web based reachability with the game theoretic approach (WBRGTA) which results in the effective communication with minimized fault detection. The key idea of our test method is to adopt a winning strategy. Semaphore is the special type of shared variable with the non-negative integers. Semaphore produces the concurrent programs for reachability testing. A reachability testing problem is that given a set P and a goal state G, we should find a winning strategy w. A strategy w is a function that during the course of the timed game constantly gives information as to what the player should do in order to win the game. Reachability testing is based on a technique called prefix-based testing which executes a test run deterministically up to a certain point and thereafter allows the test run to proceed nondeterministically.

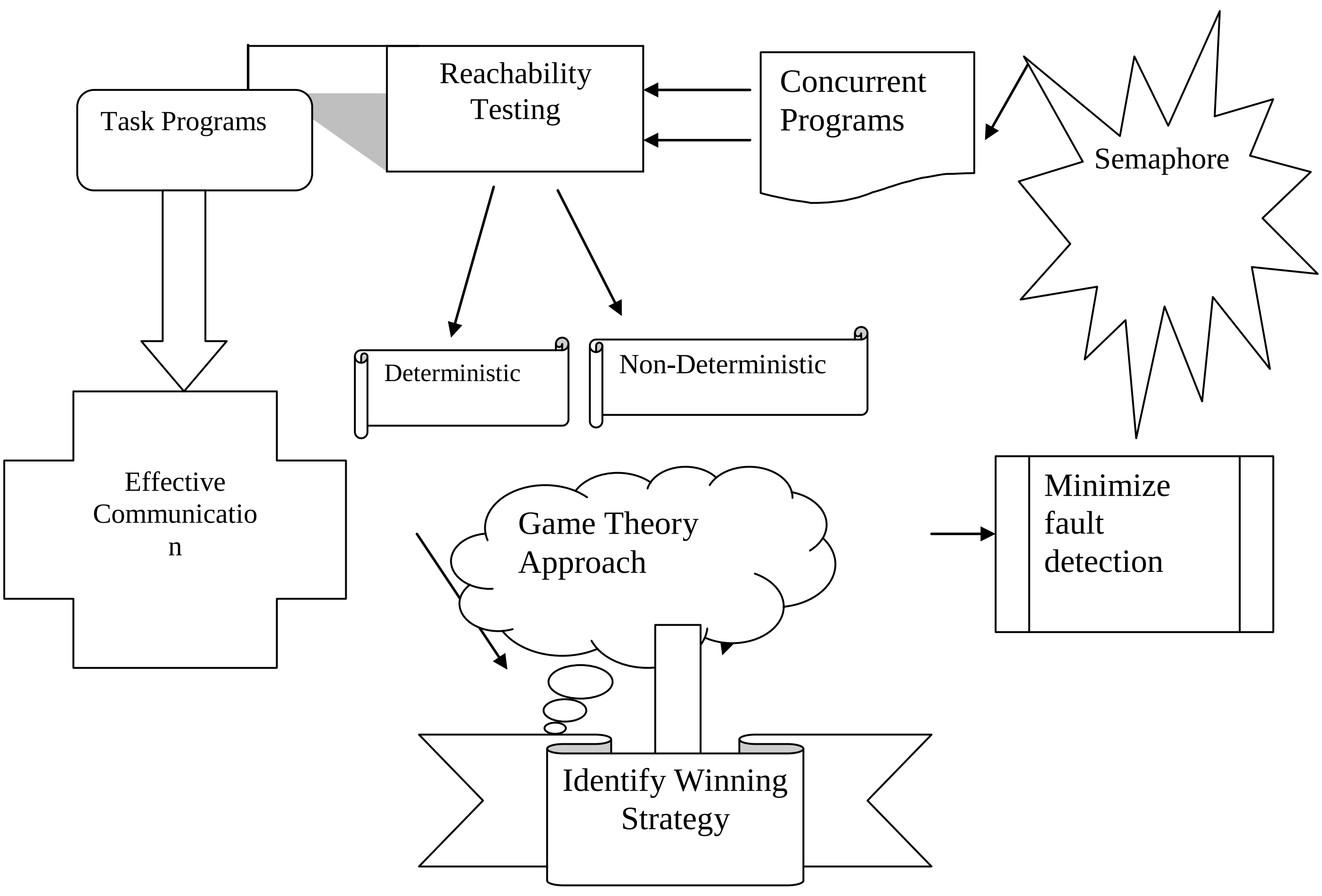


Fig 3.1 Architecture of Web Based Reachability with Game Theoretic Approach (WBRGTA)

The game theoretic approach consists of set of strategies used to find out a winning strategy with minimal faults. A fault is a discrepancy of the system structure from an insignificant situation. At a given state of the run, the player can be guided either to do a particular controllable action or to do nothing at this point in time and just wait.

3.1 Monitoring synchronization on concurrent program by reachability testing

Track synchronization is used to identify devices that are taking a considerable amount of time to complete their synchronization activity. To do this, we can identify the phase where the duration is high. In addition, devices that have exceeded the configured value of the number of days they can work without synchronizing. We propose a general execution model for the synchronization constructs. This model provides sufficient information for replaying an execution and for identifying race conditions in an execution. Replay techniques have already been developed for these constructs. Our execution model contains all the information required by these techniques. In our general execution model, we refer to a send or call event as a sending event, and receive, completion, or entry event as a receiving event. We refer to a semaphore or monitor generally as a synchronization object. If a sending events is synchronized with a receiving event r in an execution (i.e., a sent message is received or a called semaphore operation is completed or a called monitor method is entered), we refer to < s;r > as a synchronization pair and say that s is the sending partner of r and r is the receiving partner of s.

Our general execution model can also be applied to locks, which are another commonly used synchronization construct. Since locks and binary semaphores can be handled similarly, and due to space constraints, locks are not discussed in this paper. Our general model can also be used to handle shared variables by modeling a read (write) operation as receiving (sending) a message from (to) a shared variable. During an execution of a concurrent program, a sequence of synchronization events, called a synchronization sequence, takes place between the concurrent processes. Because of the unpredictable progress of concurrent processes and the use of nondeterministic synchronization constructs, multiple executions of a concurrent program with the same input may exercise different synchronization sequences and even produce different results. It is this nondeterministic execution behavior that creates problems during the testing and debugging cycle of concurrent programs.

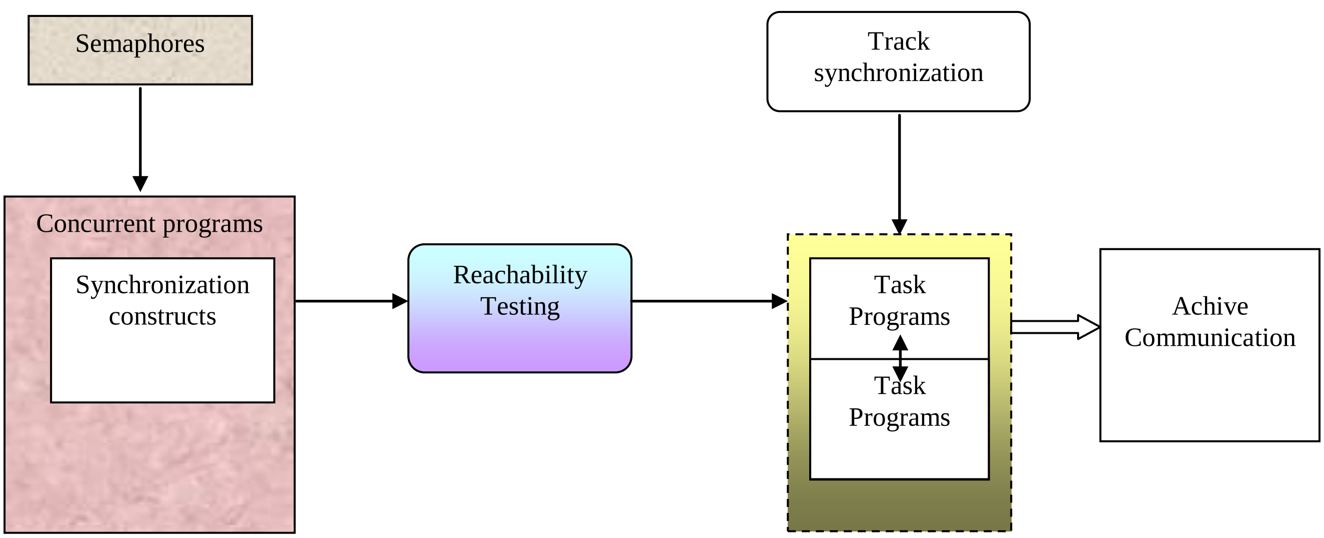


Fig 3.2. Tracking of synchronization on concurrent program by reachability testing

Our approach to solving the problems caused by nondeterministic execution behavior is called deterministic execution debugging and testing. Although we use the well-understood constructs of semaphores and monitors as examples in this article, our approach works for other concurrent constructs. Semaphores are a special type of shared variable. Semaphores are nonnegative integers on which two operations, called P and V operations are defined. In a P operation, the semaphore decrements by 1 until it reach 0, at which point the process that called the P operation is suspended. In a V operation, a process suspended by a previous P operation is awakened; if the process has not been suspended, the semaphore increments by 1. After a semaphore is initialized, only the P and V operations can access and alter its values.

Processes synchronize using P and V operations on semaphores. We assume that V operations wake processes in the order that they were delayed - a semaphore queue is first-in, first-out. Also, for simplicity, we assume that mutually exclusive accesses of shared variables are guaranteed by the use of semaphores, although it is usually impossible to guarantee mutual exclusion in a semaphore-based program with high utilization factor of semaphores. Consider a semaphore-based concurrent program that, with a particular input, exercises a sequence of P and V synchronization operations. You usually must separate the start and completion of a P operation because a process executing a P operation either completes the operation immediately or blocks itself and then later is awakened to complete the operation.

3.2 Reachability testing with game theoretic approach

The fig 4. depicts the architecture of the game theoretic approach which results in the minimized fault detection. The key idea of our test method is to adopt a winning strategy. A reachability testing problem is that given a set S and a goal state K , we should find a winning strategy f. A strategy f is a function that during the course of the timed game constantly gives information as to what the player should do in order to win the game. At a given state of the run, the player can be guided either to do a particular controllable action (i.e., to offer an input to the plant), or to do nothing at this point in time and just wait

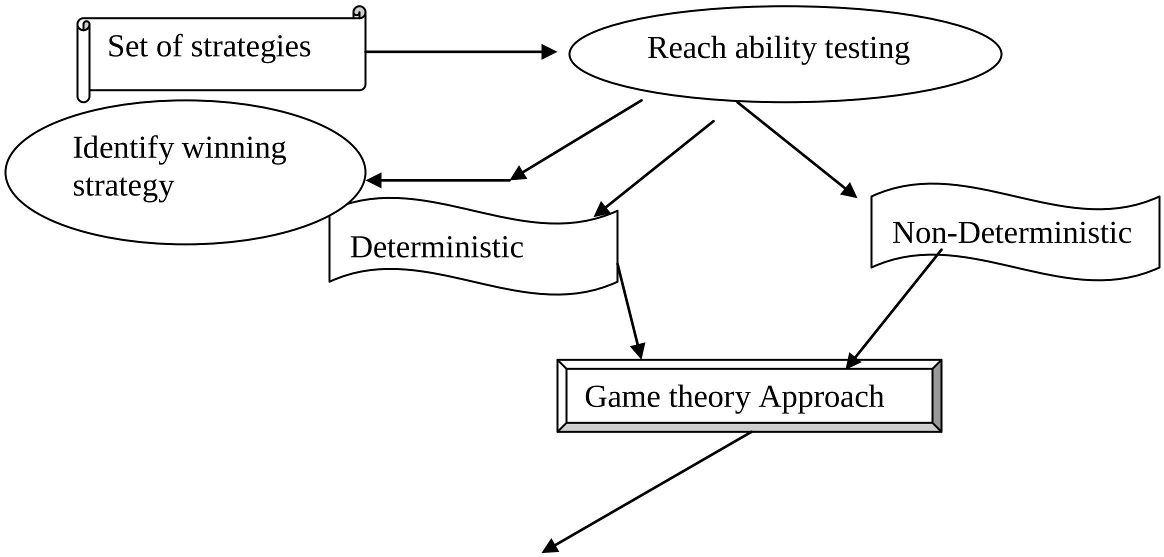


Fig 3.3. Architecture of Game Theoretic Approach

3.3 Game Theory Mathematical Model

The essential principles of game theory is that

1. Each node makes the best possible move.
2. Each node knows that his or her opponent is also making the best possible move

When new features are added into an old program, it is possible to loss the old feature of that program. While using the reachability testing to the game theoretic approach, it greatly avoids the side effects. So, the old program will retain the new program with new features to achieve an equilibrium state and to diminish the error rate using the following formulation:

The considered form of game is defined by three objects:

1) the set N = {1,2,….n} players
2) the sequence $H_1,\ldots.H_n$ of strategy sets
3) the sequence $p_1(h_1,\ldots h_n)$, $p_2(h_2..h_n)$….payoff functions of players

A game is zero sum if

$$\sum_{p_i}(h_1,h_2,...,h_n)=0 \quad \text{………. Eqn 1}$$

Where i = 1,2,..n and $h_1 € H_1$,$h_2 € H_2$,$h_n € H_n$.

For each player i and alternate strategy H we have that

$$p_i(h_1,h_2,...,h_n) \geq p_i'(h_1',h_2',...,h_n') \quad \text{…… eqn 2}$$

Where,

N – Set of Players

H– Strategies used by n player

P_i – Number of i processes

p_i – Pay-offs of strategies

The above formulations are used to retain the new program with new features in a given game theoretic approach to achieve an equilibrium state and to diminish the error rate.

3.4 Nash Equilibrium Mathematical Model

Consider a strategy Q for old program. Let u be the expected value. The expected payoffs for different strategies of new feature add on using game theory will be pQ. The best strategy can be obtained by

$$Q \geq 0 \qquad \text{\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ eqn (3)}$$

$$\sum Q_i = 1 \qquad \text{eqn (4)}$$

$$(pQ)_k \geq u \text{ for all k} \qquad \text{eqn (5)}$$

While performing the reachability testing with game theoretic approach in a specified environment, it accepts a set of functions. The reachability testing is solution concept of game theory specifying optimal strategic choices for all players by reason that none of the players has any motivation to diverge from the reachability testing because one player can not gain greater payoffs by choosing another strategy when all the other players choose the strategies given by the profile. Reflect on the Reachability Testing in Heterogeneous Environment contains a N set of players, Hi as a finite strategy set and p_i as a payoff function.

3.5 Reachability testing with game theoretic approach in heterogeneous environment

While performing the reachability testing with game theoretic approach in a heterogeneous environment, it allows accepting different set of functions in different tasks in a less interval of time.

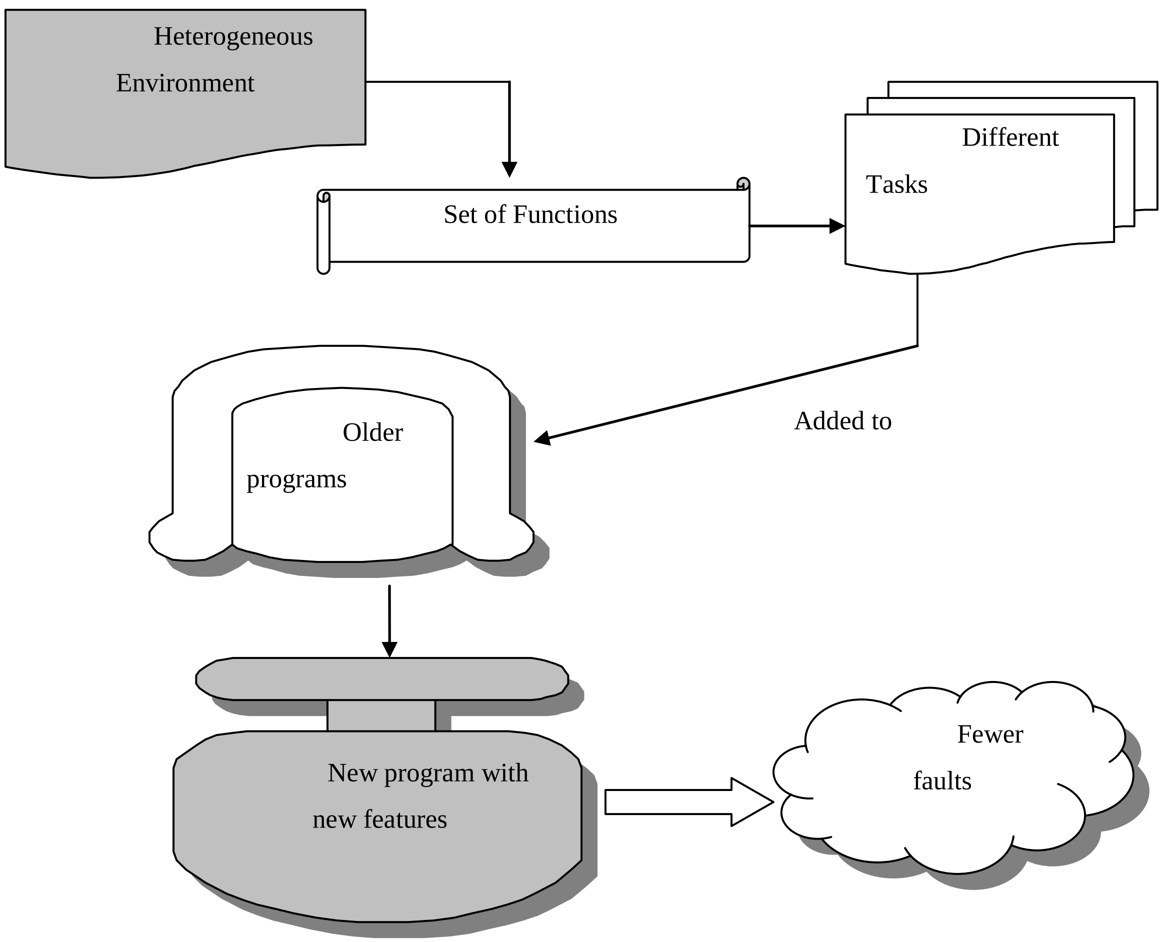


Fig 3.3 Architecture of game theoretic approach in heterogeneous environment

Consider a set of heterogeneous functions as f1, f2,…,fn which has different tasks. Consider an old program is Q and if the set of functions approached the old program Q, then the impact of Q should retain the same ole features without any alteration. While adding up the new set of functions, the old version of program Q is embedded as new version of program M but the old version Q is retained by applying reachability testing. The number of events are identified in the heterogeneous functions fn and the SYNC events, the communication behavior is noticed with pre-defined events to recover the scalability of the web based applications.

It presents a reachability testing algorithm for driving the testing process. In order to keep away from exercising the same SYN-sequence is used additionally; all existing reachability testing algorithms need to save the history of SYN-sequences that have been exercised. Our new algorithm using game theoretic approach saves no SYN-sequences, except immobile pledge that every incompletely prearranged SYN-sequence is exercised exactly once. This considerably diminishes the space and time supplies of reachability testing.

IV. EXPERIMENTAL EVALUATION

Extensive experimental revise is conducted to observe the proposed game theoretic approach using reachability testing in heterogeneous environments. We have implemented the game theoretic approach using reachability testing in Java, and approved out a series of performance experiments in order to monitor the effectiveness of the approaches. The experiments were run on an Intel P-IV machine with 2 GB memory and 3 GHz dual processor CPU. The proposed game theoretic approaches by means of reachability testing are used for real web applications like e-learning system, e-commerce and so on. The proposed work identified accurately the number of multi-threaded events occurred for a particular interval of time. By using reachability test measure, the numbers of SYNC events are identified and the communication behavior is also being noted. We have implemented several replay tools for the semaphore. Another way to speed up reachability testing is to execute multiple instances of RichTest in parallel on multiprocessor/distributed systems. Since prefix-based test runs during reachability testing are independent, interprocess communication takes place only when variants are distributed. To get an initial estimate of the speedup that is possible from utilizing multiple processors, we performed an experiment on the dining philosophers (DP) program. Program DP had six philosophers and 901,752 possible sequences. It took reachability testing 146.5 minutes to exercise these sequences on a single PC. On a cluster of 12 PCs, distributed reachability testing took only 17.5 minutes. When a game theoretic approach is applied with reachability testing using Nash Equilibrium and Fault detection, predefined events and abnormal events are recognized. The concert of the Web based Reachability testing using Nash Equilibrium and Fault detection with game theoretic approach is measured in terms of

(i) Communication Time
(ii) Utilization Factor
(iii) Space Occupied
(iv) Execution Time
(v) Testing Error Rate

Communication Time is the number of changes per second to the media being transmitted. Utilization Factor is a firm that charge the time to the clients. The execution time is the time taken to perform the given event and also it identifies the SYNC events and abnormal events. The consumption space is less for storing the predefined multi-threaded events. The testing error rate is the rate which describes the number of abnormal events raised.

V. RESULTS AND DISCUSSION

The experimental results shows the performance with the existing Reachability Testing (RT) model in terms of communication behavior, and synchronization factor, Execution time, Space occupied and testing error rate.In this work, it explains a different set of strategies written in mainstream languages such as Java. The previous took reachability testing model with Game Theory (GT) approach 20-25 seconds to perform the job whereas the proposed reachability testing model with game theory approach completed the same job in 5 seconds.

Table 5.1 No. of Semaphore vs. Communication Rate

	Communication Rate (sec)	
Semaphore	Proposed RT	Existing RT
10	.5	1
20	.7	1.5
30	.8	2
40	1	2.5
50	1.3	3

The above table (Table 5.1) describes the Communication rate by the number of semaphores taking place in game theoretic approach, when more number of semaphores approached.

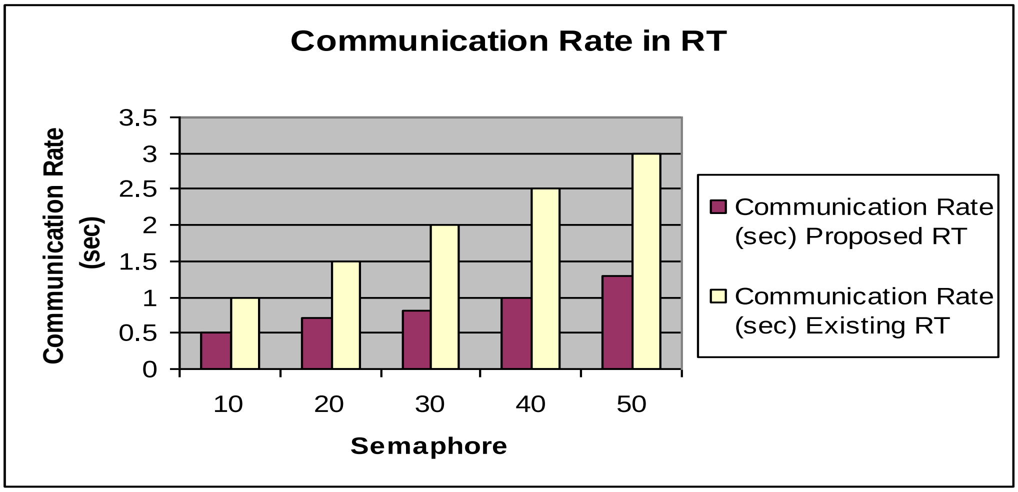


Fig 5.1 No. of Semaphore vs. Communication Rate

Fig 5.1 demonstrates the Communication rate to perform the reachability testing in the Game Theory Approach. Comparison result of the proposed WBRGTA model with game theory approach with the previous work reachability testing model with game theory approach based on Communication Rate consumption variance, measured in terms of seconds(s). When number of Semaphore applications increases in the web, the Communication rate taken for the reachability testing is 40- 50 % less in the proposed reachability testing model with game theory approach contrast to an exisitng reachability testing model with game theory approach.

Table 5.2 No. of Nodes vs. Utilization Factor

	Utilization Factor (%)	
No. of Nodes	Proposed RT	Existing RT
2	40	10
4	55	20
6	65	25
8	78	32
10	85	41

The above table (Table 5.2) describes the Utilization Factor by the number of nodes taking place in game theoretic approach, depending upon the web application.

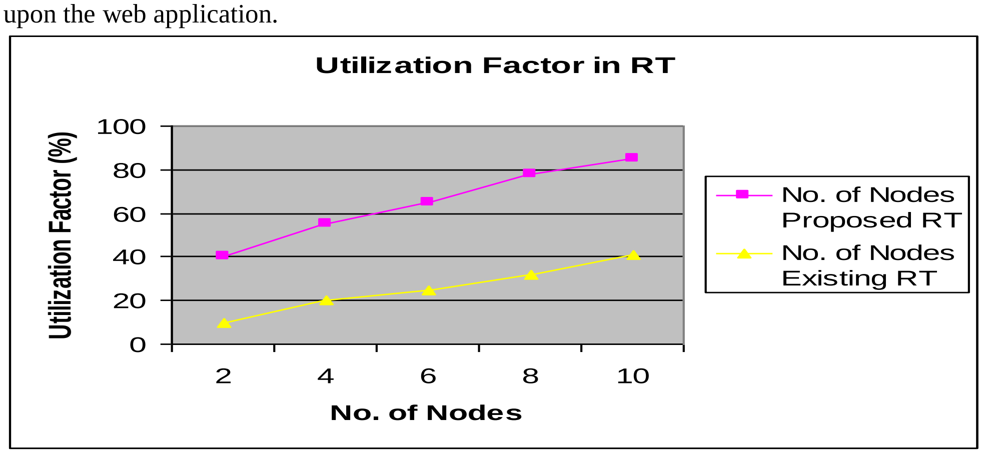


Fig 5.2 No. of Nodes vs. Utilization Factor

Fig 5.5 describes Utilization Factor to perform the reachability testing for the functions occurring in the web environment. The utilization factor of the nodes is determined based on the process accomplished with the given game theoretic approach. Comparison result of the proposed reachability testing model with game theory approach (WBRGTA) with the previous work reachability testing model with game theory approach based onUtilization Factor, measured in terms of rate(%).The variance in the utilization factor for reachability testing would be 55-65% high in the proposed reachability testing model with game theory approach using Nash Equilibrium.

Table 5.3 No. of Events vs. Space Occupied

	Space Occupied (KB)	
No. of Events	Proposed RT	Existing RT
10	20	35
20	25	40
30	35	50
40	40	65
50	42	80

The above table (Table 5.3) describes the amount of space occupied by the number of pre-defined events taking place in game theoretic approach, when more number of functions approached in web applications.

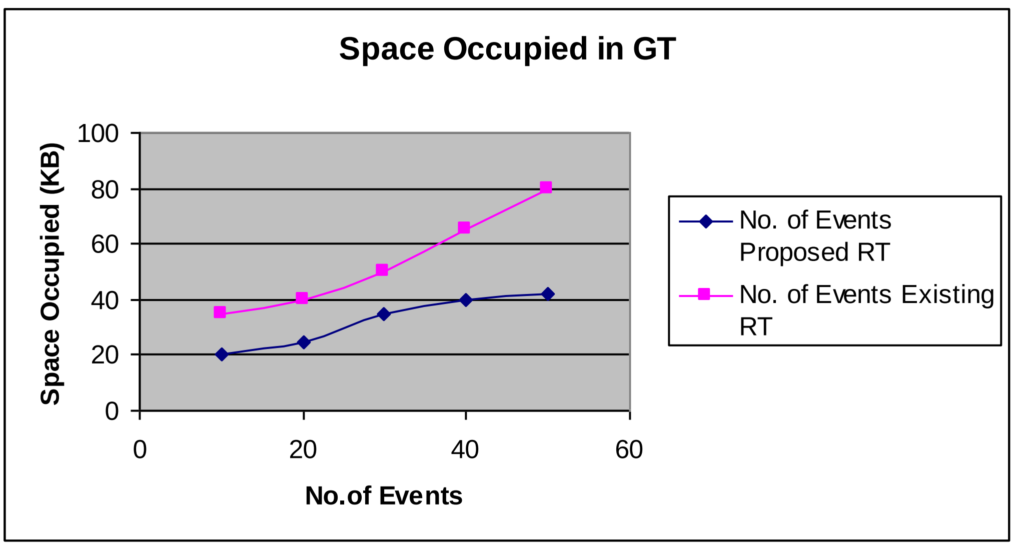


Fig 5.3 No. of Events vs. Space Occupied

Fig 5.3 illustrates the usage of space to perform the reachability testing for the events occurring in the heterogeneous environment. Various number of predefined are used in the experimentation to validate the proposed reachability testing model with game theory approach (WBRGTA) using Nash Equilibrium. Comparison result of the proposed reachability testing model with game theory approach in web environment with the previous work reachability testing model with game theory approach based on memory usage variance, measured in terms of kilobyte(KB). When number of events which has heterogeneous functions increases in the web application, the usage of memory is 10-15% less in the proposed reachability testing model with game theory approach (WBRGTA) using Nash Equilibrium contrast to an exisitng reachability testing model with game theory approach.

Table 5.4 No. of functions vs. Execution time

	Execution time (sec)	
No. of functions (f)	Proposed RT in GT	Existing RT in GT
5	10	25
10	11	26
15	13	30
20	16	28
25	15	29
30	14	30

The above table (Table 5.4) describes the time taken by the reachability testing in game theoretic approach, when more number of functions approached the game players.

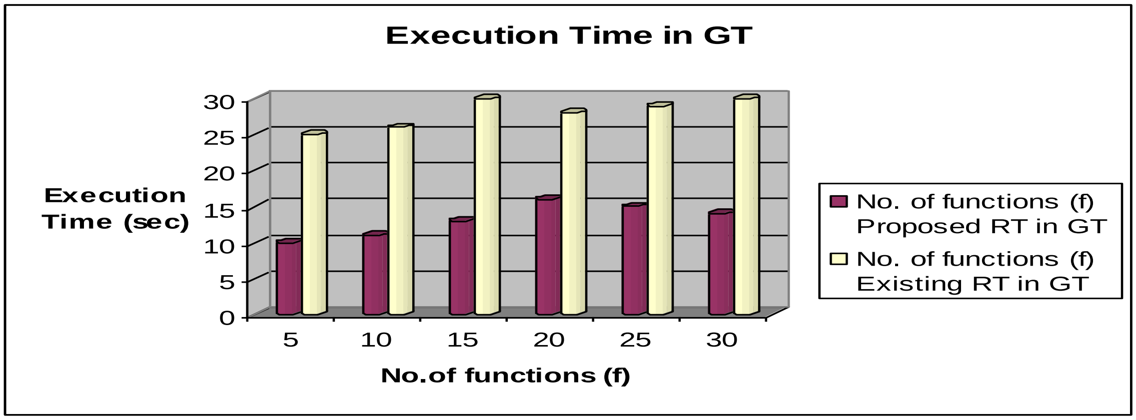


Fig 5.4 No. of functions vs. Execution time

Fig 5.4 demonstrates the time consumption to perform the reachability testing in the Game Theory Approach. The various number of web applications are used in the experimentation to validate the proposed WBRGTA with the existing game theory approach. Comparison result of the proposed WBRGTA model with game theory approach with the previous work reachability testing model with game theory approach based on execution time consumption variance, measured in terms of seconds(s). When number of functions applications increases in the web application, the execution time taken for the reachability testing is 20-25%less in the proposed reachability testing model with game theory approach contrast to an exisitng reachability testing model with game theory approach.

Table 5.5 No. of events vs. Testing Error Rate (%)

	Testing Error Rate (%)	
No. of events (e)	Proposed RT in GT	Existing RT in GT
5	5	10
10	8	12
15	11	15
20	15	18
25	19	22
30	21	25

The above table (Table 5.5) describes the Error Rate of testing by the number of pre-defined events taking place in game theoretic approach, when more number of functions approached in web applications.

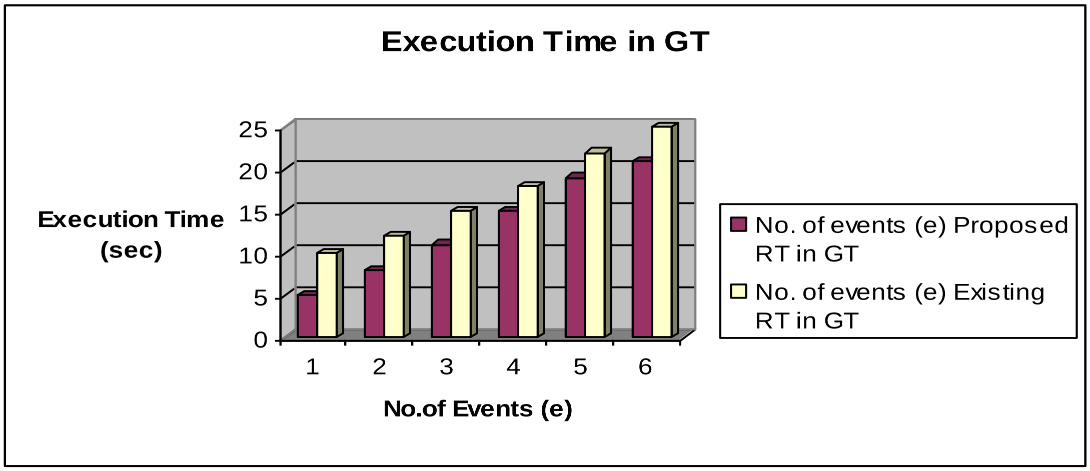


Fig 5.5 No. of events vs. Testing Error Rate (%)

Fig 5.5 describes the Error Rate to perform the reachability testing for the events occurring in the web environment. Comparison result of the proposed reachability testing model with game theory approach (WBRGTA) with the previous work reachability testing model with game theory approach based on error rate testing, measured in terms of rate(%).The variance in the error rate for reachability testing would be 10-13% low in the proposed reachability testing model with game theory approach using Nash Equilibrium.

VI. CONCLUSION

In this paper, we proposed a web based reachability testing using Nash Equilibrium and fault detection with game theoretic Approach by fulfilling the application functionality, scalability, reliability both on web and standalone platforms. The proposed testing model efficiently accepts, both homogenous and heterogeneous environment in the Game Theoretic Approach. The performance of the work is measured in terms Communication rate, Utilization factor, space occupied, efficiency and testing the error rate. Standard web applications are used to conduct the performance evaluation of the proposed reachability testing model (WBRGTA). The result shows that the proposed model in the different set of functional integration is nearly 70% better in accepting the both homogenous and heterogeneous events with multiple services for efficient testing.

REFERENCES

[1] Yu Lei and Richard H. Carver, “Reachability Testing of Concurrent Programs” IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 32, NO. 6, JUNE 2006.

[2] R. Carver and K.C Tai, Modern Multithreading. Wiley, 2005.

[3] G. Fey, S. Staber, R. Bloem & R. Drechsler (2008): Automatic Fault Localization for Property Checking. Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on 27(6), pp. 1138 –1149.

[4] B. Jobstmann, S. Staber, A. Griesmayer & R. Bloem (2011): Finding and Fixing Faults. Journal of Computer and System Sciences (JCSS)

[5] P. Bouyer, T. Brihaye & F. Chevalier (2010): O-Minimal Hybrid Reachability Games. Logical Methods in Computer Science 6(1:1), pp. 1–48.

[6] P. Bouyer, F. Chevalier & D. D’Souza (2005): Fault Diagnosis Using Timed Automata. In: Foundations of Software Science and Computational Structures, Lecture Notes in Computer Science 3441, Springer Berlin / Heidelberg, pp. 219–233, doi:10.1007/978-3-540-31982-5 14.

[7] C. Flanagan and P. Godefroid, “Dynamic Partial Order Reduction for Model Checking Software,” Proc. 32nd Symp. Principles of Programming Languages (POPL), pp. 110-121, 2005.

[8] P. Godefroid, "Software Model Checking: The VeriSoft Approach," Formal Methods in System Design, vol. 26, no. 2, pp. 77- 101, 2005.

[9] Yi Shi; Jiaheng Wang et. Al., "A game-theoretic approach for distributed power control in interference relay channels", IEEE Transactions on Wireless Communications, June 2009

[10] Yong Xiao; Guoan Bi et. Al., 'Game Theoretic Analysis for Spectrum Sharing with Multi-Hop Relaying', IEEE Transactions on Wireless Communications,2011

[11] Niyato, D.; Ping Wang et. Al., "Exploiting Mobility Diversity in Sharing Wireless Access: A Game Theoretic Approach", IEEE Transactions on Wireless Communications, 2010

[12] Zhen Yang; Huadong Ma, "Hotspot Avoidance for P2P Streaming Distribution Application: A Game Theoretic Approach", Parallel and Distributed Systems, IEEE Transactions on 2009.

[13] Yu Lei and Richard H. Carver, "Reachability Testing of Concurrent Programs" IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 32, NO. 6, JUNE 2006.

[14] Y. Lei and R.H. Carver, "Reachability Testing of Semaphore-Based Programs," Proc. 28th Computer Software and Applications Conf. (COMPSAC), pp. 312-317, 2004.

[15] Y. Lei and R. Carver, "Reachability Testing of Monitor-Based Programs," Proc. Int'l Conf. Software Eng. and Applications, pp. 312- 317, 2004.

[16] Y. Lei and R. Carver, "A New Algorithm for Reachability Testing of Concurrent Programs," Proc. Int'l Symp. Software Reliability Eng., pp. 346-355, 2005.

[17] S.D. Stoller, "Testing Concurrent Java Programs Using Randomized Scheduling," Proc. Second Workshop Runtime Verification (RV), 2002.