An Expected Value Prediction of Game Theory for Pre-copy over Cloud Computing

Yao-Chiang Yang[1], Wei-Tsong Lee[1], Hsin-Wen Wei[1], Vooi Voon Yap[2]

[1] Department of Electrical and Computer Engineering, Tamkang University, Taiwan
[2] Faculty of Engineering and Green Technology, Universiti Tunku Abdul Rahman, Malaysia
ychiangy@gmail.com.tw, wtlee@mail.tku.edu.tw, hwwei@mail.tku.edu.tw, yapvv@utar.edu.my

Abstract

Virtual Machine (VM) in a cloud environment has become popular. An operating system (OS) can be installed on a VM platform which emulates a real machine in a cloud environment. In such a setup, the physical host can be taken offline for maintenance and the service continues. This approach relies on Pre-copy live migration technique. With this technique, one of the challenges is how to ensure that the existing memory pages can be securely migrated to the alternative host without wasting unnecessary migrations; having minimum migrations will reduce the total migration time. In this paper, the authors propose a method which uses the expected value of Game Theory over the Gilbert-Elliot (GE) model to predict memory migration. Furthermore, with this method, it is envisaged when the service of the original physical host is halted it can reduce the delay when VM migrates to an alternative host. By the simulations, it is observed that using the Game Theory model is faster than the method in the GE model to obtain the "CUT-OFF" point in varied memory states.

Keywords: Pre-Copy, Gilbert-Elliot (GE) model, Game theory

1 Introduction

As mentioned in the abstract, cloud computing technology has become popular and it provides the community with many conveniences. There are also evidences that more people are utilizing the resources of Cloud Computing to replace the jobs which are originally designed to perform on the desktop, laptop, and smart phones. Cloud computing is able to serve more people because of the virtual machines and it can effectively utilize the resource in the host, and also distribute the jobs to different virtual machines.

A Virtual Machine [1-2] (VM) is a virtualization technology. It can provide a platform where an operating system (OS) can be installed to execute computer programs. This means a VM with an OS can emulate a real computer. A number of VMs can be created on the same host and each VM can host an independent OS.

There are occasions where the host needs to be shut down for maintenance. In such situation, the VM will be migrated to an alternative host. This process of moving a running VM to another physical host withoutdisconnecting the client is as well-known, live migration. With live migration, no service is taken offline and users are always not aware of the migration processes being going. The migration processes can be separated into Non-live Migration and Live Migration. Live migration [3-4] is well-known as a Dynamic Migration. Live Migration can help the host to perform the migration of the existing jobs (or memory page) to the destined host. Figure 1 is an example, which shows the VM migrated to the destined host from the original host. The VM in the destined host can be recovered quickly in a very short time.

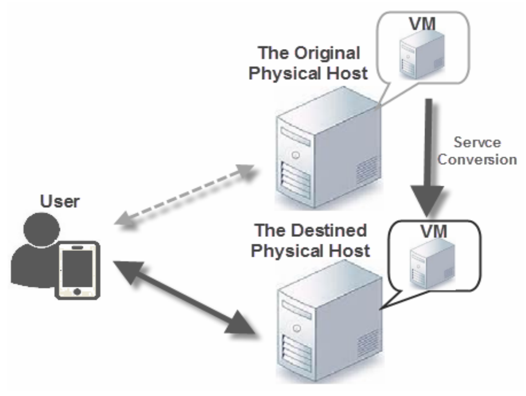


Figure 1. The migration of Virtual Machine (VM)

Live Migration can provide service for the user continuously without any change during the migration process to an alternative host. The process of Live Migration can be divided into three phases, that is, push phase, stop-and-copy phase and pull phase [5-10] as shown in Figure 2. The push phase and the stop-and-copy phase can be combined to form a Pre-copy memory migration. The post-copy memory migration is a combination of the last two phases, that is, stop-and-copy phase and pull phase.

*Corresponding Author: Wei-Tsong Lee; E-mail: wtlee@mail.tku.edu.tw
DOI: 10.3966/160792642020112106003

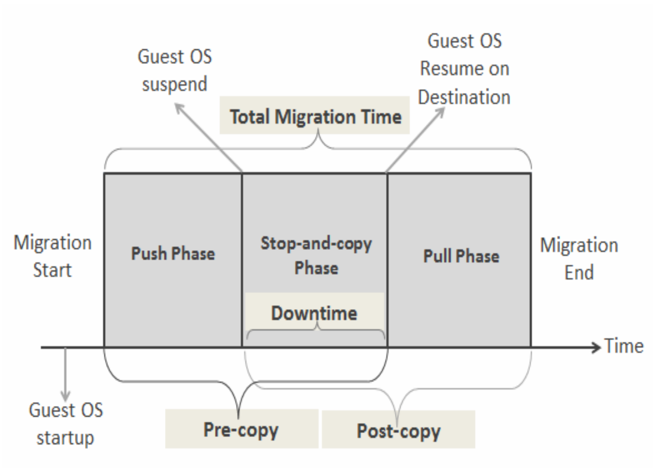


Figure 2. Pre-copy and Post-copy process [10]

Push Phase: To start the migration of the existing memory data working in the original host to the destined host. Memory data will be re-migrated if the memory data has been modified during the migration.
Stop-and-copy Phase: To stop the original virtual machine immediately, the remaining memory data has be migrated from the original host to the destined host.
Pull Phase: To start the VM of the destination, the incomplete memory data will cause a page fault, and the lack of memory data will be re-migrated from the original host.

In order to evaluate the performance of the migration, Total Migration Time and Downtime are used. Total Migration Time is a period with from the start of the migration until all migrations are completed. However, Downtime is defined as a period with the virtual machine of the original host suspended, and the memory data is migrated to the destination immediately until the destined host resumed.

The challenge with the Pre-copy method is how to ensure the existing memory page can be securely migrated to the destined host without wasting the unnecessary migration time. This is important because the migrated memory page needs to be re-migrated as long as the memory page in original host is being marked as "dirty". The destined memory page cannot be utilized anymore. The goal is to obtain an exact cutoff time in order to go the Stop-and-copy Phase.

The rest of this paper is organized as follows. Section 2 provides an overview of the Gilbert-Elliot or GE Model, Game Theory and the technique of memory migrated prediction. Section 3 provides a description of the proposed method of how the expected value by Game Theory is obtained and define a threshold number on whether it should go the migration or not. Section 4 presents the comparison between Game Theory and the GE Model. Finally, Section 5 provides the conclusion.

2 Background and Related Work

2.1 Overview of Gilbert-Elliot (GE) Model

The following figure shows the GE Model [11-12] ;

Figure 3 shows the two states of Markov process [13]. The "Good" state means that the state which is defined as "**READ**" or "0" in transmission, and the "Bad" state means the state which is defined as "**WRITE**" or "1" in transmission. The bit error rates were defined as ε_1 and ε_2, where $\varepsilon_1 \ll \varepsilon_2$. The probability transition is as follows

$$P_{GG} = P_{00} = P(G|G) = P[\text{The } E_j \text{ state of } G \mid \text{The } E_i \text{ state of } G] = 1\text{-}b$$
$$P_{BG} = P_{10} = P(B|G) = P[\text{The } E_j \text{ state of } B \mid \text{The } E_i \text{ state of } G] = b$$
$$P_{GB} = P_{01} = P(G|B) = P[\text{The } E_j \text{ state of } G \mid \text{The } E_i \text{ state of } B] = g$$
$$P_{BB} = P_{11} = P(B|B) = P[\text{The } E_j \text{ state of } B \mid \text{The } E_i \text{ state of } B] = 1\text{-}g$$

$$S = \begin{bmatrix} P_{GG} & P_{BG} \\ P_{GB} & P_{BB} \end{bmatrix} = \begin{bmatrix} P_{00} & P_{10} \\ P_{01} & P_{11} \end{bmatrix} = \begin{bmatrix} 1-b & b \\ g & 1-g \end{bmatrix} \quad \textbf{(1)}$$

Thus, the state transition matrix S can be defined as follows.

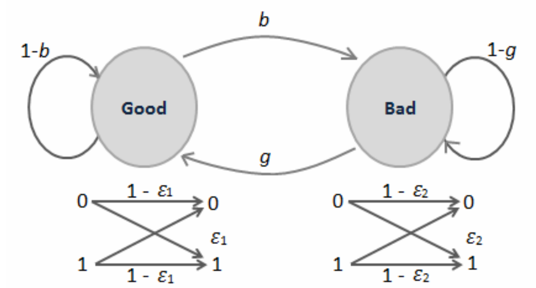


Figure 3. The Gilbert-Elliot(GE) model [12]

Markov process in state "G" and state "B" will be closed to a constant, and $P(G)$ is equal to $1\text{-}P(B)$, where $P(G)$ is a good steady state probability, and $P(B)$ is a bad steady state probability. They can be obtained by the following (2) and (3).

$$P(G) = \frac{g}{b+g} \quad \textbf{(2)}$$

$$P(B) = \frac{b}{b+g} \quad \textbf{(3)}$$

According to the GE model, the average BER (Bit Error Rate) can be calculated as (4).

$$Ave\_BER = P(G) \times \varepsilon_1 + P(B) \times \varepsilon_2 = \frac{g}{b+g} \times \varepsilon_1 + \frac{b}{b+g} \times \varepsilon_2 \quad (4)$$

2.2 Overview of Game Theory

The proof of the initial Game Theory [14-17] was provided by John von Neumann in 1944. The proof considers the existence of mixed-strategy equilibrium in a two-person zero-sum game. Many researchers utilize the method of Game Theory for the strategy analysis in applied science. The aim of Game Theory is to find the best strategy for the participants themselves, and each game can be divided into

Players: There should be at least two players which are known as the competitors. Each competitor will try to get the best interest for themselves.

Rule: Each competitor should understand and follow the rules of the game.

Strategy: Each competitor has to select two or more actions to secure the best interest in the game.

Payoff: To collect all competitors' strategy, to produce a payoff matrix for various results.

Table 1. The strategies of Player: R and Player: C

Player: C / Player: R	Strategy A	Strategy B
Strategy A	a	b
Strategy B	c	d

Table 1, it assumes that there are two players in the game, one is player R, another is player C. Player C is a competitor of R. Each of them tries to come out with some strategies to make sure they can get the best interest for themselves in the game.

The mixed strategy "*P*" of player R is

$$\mathrm{P} = \begin{bmatrix} p & 1-p \end{bmatrix} \quad \mathbf{(5)}$$

The mixed strategy "*Q*" of player C is

$$\mathrm{Q} = \begin{bmatrix} q \\ 1-q \end{bmatrix} \quad \mathbf{(6)}$$

Assuming the payoff matrix "*M*" of two players R and C is

$$\mathrm{M} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} ;\ \mathrm{a + d \neq b + c} \quad \mathbf{(7)}$$

The expected value "*E*(*P*, *Q*)" of player R will be

$$\mathrm{E(P, Q) = PMQ} = \begin{bmatrix} p & 1-p \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} q \\ 1-q \end{bmatrix} \quad \mathbf{(8)}$$

2.3 The Prediction of Pre-copy Migration

In [18], Sun et al defined the memory pages with "NOT MODIFIED" as "0", which means no memory modification. The "READ" operation is defined as "1", which means the memory page has been read. The "WRITE" operation defined as "2", which means the memory page has been modified. Sun et al also designed a predicted method which is based on the request of memory pages prediction to define seven tables in their paper. The seven tables are "Change_table", "Dirty_table", "Probability_table0", "Probability_table", "Send_table", "Skip_table" and "Fix_table". All functions were defined as follows:

Change_table: To record the states in memory pages during migration.

Dirty_table: To record the current dirty memory pages.

Probability_table0: To copy the current dirty pages from Dirty_table into Probability_table0.

Probability_table: To calculate the dirty probability of each memory page when going into the next migration iteration.

Send_table: During the calculation, the smaller dirty probabilities in memory pages will be picked up and ready to migration.

Skip_table: To record the newest dirty memory pages during the probability calculation.

Fix_table: To record the remaining memory pages which are set to migrate during the Stop-and-copy Phase.

And nine steps are designed in Figure 4;

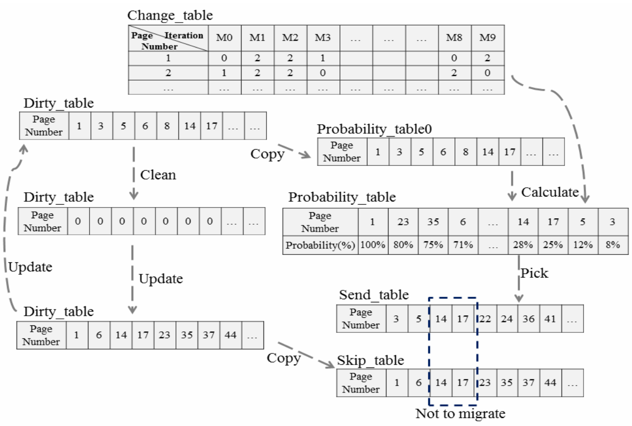


Figure 4. The flowchart of predicted method [18]

Step1: After VM obtains the migration message, the VM starts to monitor each state in memory pages and writing each memory state into the Change_table accordingly.

Step2: To update the Change_table then go to Step 4 if migration iteration has begun.

Step3: To update the Dirty_table and copy Dirty_table into the Probability_table0 directly, then the Dirty_table will be cleared.

Step4: Based on the Change_table, calculate each dirty memory probability from Probability_table0; calculate each dirty memory probability from Change_table directly if the migration iteration has been performed.

Step5: After calculation, save the probability result into the Probability_table.

Step6: Set a threshold "Probmax"; if the dirty memory probabilities are smaller than "Probmax", copy them directly into the Send_table.

Step7: Update the Dirty_table again and copy

Dirty_table directly into the Skip_table.

Step8: Compare the Send_table with the Skip_table, the memory pages in the Send_table will not be migrated if the same memory pages can be found in the Skip_table.

Step9: Diagnose if the migration iterations can be stopped to go into the Stop-and-copy Phase, if not, go back to Step 2.

In [8], Wu et al designed a predicted method which is based on two-memory states.

In Figure 5, a "0" is defined to express both "NOT MODIFIED" and "READ" in the memory page since for memory modification, there is no difference between them. The "1" is defined to express the "WRITE", which means memory has been modified in the memory page. Wu et al utilized the Markov process[12] to obtain

$$P(E_i E_j) = P(E_j \mid E_i) = P_{ij} \quad \textbf{(9)}$$

Where *Ei* is defined as the existing state of memory and the *Ej* is defined as the next state of memory. However, there are only two states in memory, that is, "**READ**" and "**WRITE**". The two states situation is proper in explanation of the memory states with "**READ**" as the E_0 and "**WRITE**" as the E_1.

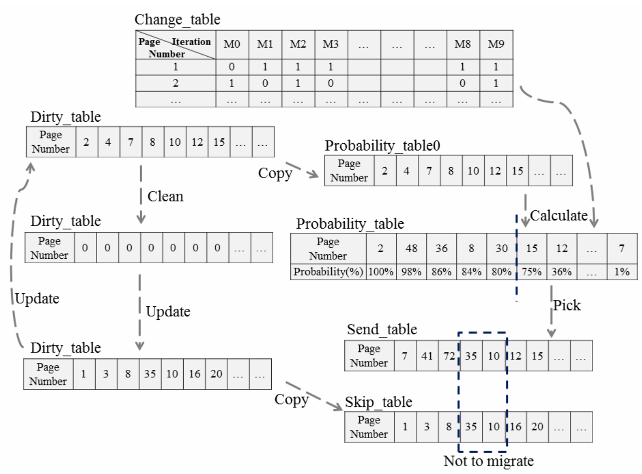


Figure 5. The flowchart of memory prediction [8]

3 Problem Statement and Proposed Method

3.1 The Problem Statement

In [8], Wu et al proposed a method that utilizing the Markov process in prediction of Pre-copy migration. They calculated the subsequent probability prediction which is based on the existing dirty memory page. However in [9], Lin proposed which using the GE model to improve the defect of the method with Markov process. The rational for this approach is because in [8], Wu et al consider the case of "Correct Memory Page" →"Dirty Memory Page", but the case of "Dirty Memory Page" → "Correct Memory Page" was not considered in the method. Thus, in [9], Lin proposed the GE model in the prediction of Pre-copy migration. The proposed method has the benefit of covering the case of "Dirty Memory Page" → "Correct Memory Page". However, a lot of iterations have to be performed in preparation for collecting the information of dirty memory pages. This method would take much computational time because of the iterative actions.

3.2 The Expected Value of Game Theory

We propose using the expected value of Game Theory in the prediction calculation over the GE model. The goal is to attempt to compute a "CUT-OFF" point under variable memory states which can stop the iteration and go directly into the Stop-and-copy Phase. The definition of the variable memory states may cause memory variation during the iteration stage. It is not easy to obtain the stable memory state to discover the "CUT-OFF" point. Selecting Game Theory, it is expected to gain the "CUT-OFF" point as sooner than GE model. It is because the theory has a predicting property that it can obtain the expected value through every calculation. Thus, to define an expected value EN_{th} as the reference value at $M_{N_{th}}$ iteration. N_{th} is defined as an integer of 0~9. Similarly, the expected value $E_{(N+1)_{th}}$ at $M_{(N+1)_{th}}$ iteration can be obtained. Finally, the proposed method define a number as a convergent level that can be obtained through the calculation of $|E_{(N+1)_{th}} - EN_{th}|$. This equation is used to evaluate the convergent condition at every iteration. In addition, setting this convergent number to be the "CUT-OFF" point can help to make a decision to stop the existing migration and go directly into the Stop-and-copy Phase. The detailed flowchart is shown as in Figure 6.

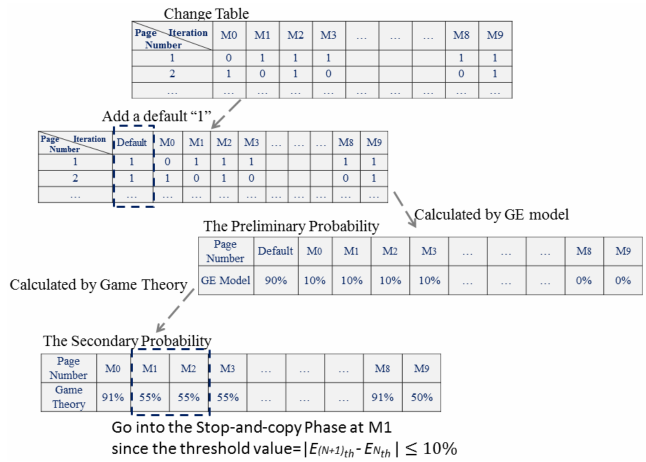


Figure 6. The flowchart of utilizing Game Theory

Figure 6 is the flowchart as the expected value of Game Theory over the GE model. It notes that from the flowchart, the default value "1" is added for reference. This is located at "Default" column next to M_0. M_0 means the 0_{th} iteration of N_{th} iterations. This action can help to calculate the probability of the dirty memory page easily. Secondly, we put each memory page number into the GE model for the calculation. From the calculations (1) to (4), the preliminary probability M0 to M9 can be obtained. Thirdly, following the Game Theory by inserting the existing Preliminary Probabilities M0 to M9 into the Table 1 of Game Theory, $M_{N_{th}}$ iteration into Table 1 is set as Player C. Similarly, $M_{(N+1)_{th}}$ iteration is inserted into Table 1 and set as Player R. Thus, forming Table 2 below:

Table 2. The probability of the $M(N+1)_{th}$ and MN_{th} iteration

$M_{(N+1)_{th}}$ \ MN_{th}	Migration	Non- Migration
Migration	a	b
Non-Migration	c	d

The payoff matrix of two players $M(N+1)_{th}$ and MN_{th} is

$$Payoff\ Matrix = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \textbf{(10)}$$

Then following (5), "*p*" is set as the migration of the $M_{(N+1)_{th}}$ iteration. Where "*p*" is the probability to perform the migration at $M_{(N+1)_{th}}$ iteration. So that "1-*p*" is another probability which is relative to "*p*" to express not to perform the migration at $M_{(N+1)_{th}}$ iteration. Therefore, the mixed strategy "P" can be written

$$ap+c(1-p) = bp+d(1-p) \quad \textbf{(11)}$$

$$p = \frac{d-c}{a-b-c+d} \quad \textbf{(12)}$$

$$P = \begin{bmatrix} p & 1-p \end{bmatrix} = \begin{bmatrix} \frac{d-c}{a-b-c+d} & 1-\frac{d-c}{a-b-c+d} \end{bmatrix} \quad \textbf{(13)}$$

Similarly, following (6), "*q"* is set as the migration at $M_{N_{th}}$ iteration. Where "*q*" is the probability of the migration at $M_{N_{th}}$ iteration. So that "1- *q*" is another probability which is relative to "*q*", which does not perform migration at $M_{N_{th}}$ iteration. Therefore, the mixed strategy "*Q*" can be written

$$Aq + b(1-q) = cq + d(1-q) \quad \textbf{(14)}$$

$$q = \frac{d-b}{a-b-c+d} \quad \textbf{(15)}$$

$$Q = \begin{bmatrix} q & 1-q \end{bmatrix} = \begin{bmatrix} \frac{d-b}{a-b-c+d} & 1-\frac{d-b}{a-b-c+d} \end{bmatrix} \quad \textbf{(16)}$$

It would be easy to set "*a*" and "*d*" as 100% since there are no actions over them. "*b*" can be set as the preliminary probability of the $M_{(N+1)_{th}}$ iteration which is obtained through the calculation from the GE model. Similarly, "*c*" can be set as the preliminary probability of the $M_{N_{th}}$ iteration through the calculation by GE model as well. Using the results from (10) to (16), the secondary probability is obtained by putting the results back into (8). As defined earlier, $E_{N_{th}}$ is the reference value at $M_{N_{th}}$ iteration. Where N_{th} is defined as an integer of 0~8, and $(N+1)_{th}$ is defined as an integer of 0~9. Thus, per calculation, the secondary probabilities *E0~E9* can be obtained. Where *E0~E9* stand for each reference value at M0~M9. In addition, the equation $|E_{(N+1)_{th}} - E_{N_{th}}|$ is used to evaluate the convergent condition at every iteration. The threshold value of 10% is being set as the "CUT-OFF" point. The threshold value of 10% is used because of the rate of variation is very small and convergent. If the result of $|E_{(N+1)_{th}} - E_{N_{th}}|$ is larger than 10%, that means the migration of Pre-copy is not useful.

The following section explains how the "CUT-OFF" point is obtained from the Game Theory. Table 3 is a memory pages list at M0~M9. Note, only 10 memory pages are used to simplify the explanation. In reality the memory page should be generally more than 10 pages.

Table 3. The list of memory page state

Number \ Page Iteration	M0	M1	M2	M3	M4	M5	M6	M7	M8	M9
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	1	1	1
3	1	0	0	0	0	0	0	0	0	0
4	1	1	0	0	0	0	0	0	0	0
5	1	1	1	1	1	0	0	0	0	0
6	0	0	0	0	0	0	0	1	1	1
7	1	1	1	1	1	1	1	1	1	1
8	1	1	1	0	0	0	0	1	1	1
9	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	0	1	1	1

Table 3 is a list of memory page state based on the two-memory states. A "0" in Table 3 is defined to express both "NOT MODIFIED" and "READ" in corresponding memory pages. A "1" is defined as "WRITE", which means memory has been modified in corresponding memory pages.

Then, each default value "1" is set and placed in front of M0, where M0 means the first 0th iteration of iterations. The default value is set to "1" because it is assumed all of memory pages have performed "WRITE" initially. Setting the default value to "1" can make the Game Theory work. As discussed in Section 2.2, the payoff matrix of Game theory has been created. There are at least two players in this game. In order to obtain the expected value at M0~M9, especially at M0, the default value "1" must be inserted in front of M0. Pleast note the default value "1" does not have real significance; the only purpose is for calculation.

As Table 4 shown, each default value "1" has been inserted in front of M0. Using the GE model, that is, "0 → 1" and "1 → 0" it is observed that there is no occurrence of "0 → 1" for M0. It is also further observed that "1 → 0" occurs 3 times in 10 page numbers.

$$P_{BG} = P_{10} = P(B \mid G) = b = 0 \tag{17}$$

$$P_{BG} = P_{01} = P(B \mid G) = g = 30\% \tag{18}$$

Table 4. The list of memory state with default value "1"

Number \ Page Iteration	M0	M1	M2	M3	M4	M5	M6	M7	M8	M9
1	1	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	1	1	1
3	1	1	0	0	0	0	0	0	0	0
4	1	1	1	0	0	0	0	0	0	0
5	1	1	1	1	1	0	0	0	0	0
6	1	0	0	0	0	0	0	1	1	1
7	1	1	1	1	1	1	0	0	0	0
8	1	1	1	1	0	0	0	1	1	1
9	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	0	1	1	1

Using equation (4), the preliminary probability at M0 is

$$P(G)\times \varepsilon_1 + P(B)\times \varepsilon_2 = \frac{g}{b+g}\times \varepsilon_1 + \frac{b}{b+g}\times \varepsilon_2$$

Assuming $\varepsilon_1 = 0.1$ and $\varepsilon_2 = 0.9$.

The preliminary probability can be obtained when each page number is inserted into the GE model that is shown in Table 5.

Table 5. The preliminary probability list

Number \ Page Iteration	M0	M1	M2	M3	M4	M5	M6	M7	M8	M9
GE Model	90%	10%	10%	10%	10%	10%	10%	90%	0%	0%

Using the data from Table 2, the preliminary probability of M0 and Default iteration is inserted into the calculation of the expected value of the Game Theory shown below in Table 6.

Table 6. The probability list of Mo and Default

Mo \ Default	Migration	Non-Migration
Migration	100%	10%
Non-Migration	90%	100%

The mixed strategy "*P*" is

$$P = \begin{bmatrix} p & 1-p \end{bmatrix} = \begin{bmatrix} 10\% & 90\% \end{bmatrix} \tag{19}$$

The mixed strategy "*Q*" is

$$Q = \begin{bmatrix} q \\ 1-q \end{bmatrix} = \begin{bmatrix} 90\% \\ 10\% \end{bmatrix} \tag{20}$$

And the payoff matrix of M0 and Default is

$$\text{Payoff Matrix} = \begin{bmatrix} 100\% & 10\% \\ 90\% & 100\% \end{bmatrix} \tag{21}$$

Thus, the secondary probability "*E0*" of M0 will be

$$E_0 = \begin{bmatrix} p & 1-p \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} q \\ 1-q \end{bmatrix} = 91\% \tag{22}$$

Then utilizing the same approach, the remaining secondary probability of M1-M9 can be obtained. The list is as Table 7.

Table 7. The secondary probability list

Number \ Page Iteration	M0	M1	M2	M3	M4	M5	M6	M7	M8	M9
GE Theory	91%	55%	55%	55%	55%	55%	55%	91%	91%	50%

Finally, the threshold value of 10% is being set to decide whether the migration of Pre-Copy should run continuously or halt the original host immediately. The threshold value of 10% is being set is because the rate of variation is very small, probably wasting unnecessary time. If performing the migration in 10%, the rest of memory would become “dirty pages” easily since the basic operation in host needs the support of a few memory sizes. A 10% means that there is no further assistance on the migration of the Pre-copy in Push Phase. Going into the Stop-and-copy Phase would be constructive.

$$\text{Threshold value} = |E_{(N+1)_{th}} - E_{N_{th}}| \quad \textbf{(23)}$$

From table 7, it can be observed that $|E2 - E1|$ at M1 is less than 10%, which means keeping the migration running at M2 in the Push Phase will not be necessary, i.e. the law of diminishing returns. The original host should be stopped immediately at M1, then move on to the Stop-and-copy Phase.

4 Results and Analysis

In this section, we will evaluate which migration method is more efficient to obtain the “CUT-OFF” point to stop the host and going into the Stop-and-copy Phase. This work will compare simulation results that using the Game Theory from the GE model.

4.1 Random Memory State under $\varepsilon_1 = 0.1$

4.1.1 Parameters Definition

In this simulation, the parameter values are set as shown in Table 8. The memory state is set according to the definition stated in [7] which is “0” for “**READ**” and “1” for “**WRITE**”. In addition, the memory states are generated by programming the random process and the number of memory pages. With Following above, the migration varies under $\varepsilon_1 = 0.1$ with $\varepsilon_2 = 0.1, 0.3, 0.5, 0.7$ and 0.9 are shown as below, where ε_1 and ε_2 are bit error rates in the GE model.

Table 8. Parameter list for the simulation under $\varepsilon_1 = 0.1$

Parameter	Value
Memory State	0 or 1
Memory Statue Generated	Random
Number of Migration Iterations	9
Number of Memory Pages	1000
ε_1 of Bit Error Rates	0.1
ε_2 of Bit Error Rates	0.1, 0.3, 0.5, 0.7, 0.9

4.1.2 Simulation Result

Following the parameters setting shown in Table 8, a comparison is made with the method in Game Theory and the GE Model. These are shown in Figure 7 to Figure 11. Note, the preliminary probability and secondary probability respectively are also shown in the figures.

It is deemed necessary to compare the benefit that using the Game Theory with the GE model. It is necessary to use the equation (23) to calculate the convergent level. In other words, the variation of the preliminary probability of the $M_{(N+1)th}$ and M_{Nth} is calculated. The variation threshold is set as defined in equation (23), that is 10%. Therefore, the calculated convergent value in the Game Theory and the calculated variation value in the GE model are at the same level. Both values are used in the comparison to determine which method is prompt and quick to stop the original host, and then move on to the Stop-and-copy Phase.

In Figure 7 and Figure 8, the default value in first column is not being considered. This is because this work only focuses on the calculation between the probability values in $M_{(N+1)th}$ and M_{Nth}, and the default value is not adopted. As Figure 7 to Figure 8 shown, if the convergent value of Game Theory is calculated, it can obtain the same result as the variation calculation of the GE model at M0 is less than 10%. Which means the original host is going to be stopped, and moves on to the Stop-and-copy Phase at M0.

Figure 7. The probability under $\varepsilon_1 = 0.1$ and $\varepsilon_2 = 0.1$

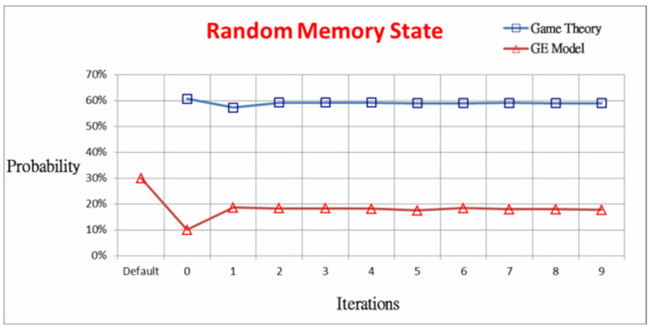


Figure 8. The probability under $\varepsilon_1 = 0.1$ and $\varepsilon_2 = 0.3$

As shown in Figure 9 and Figure 10, the deviation of probabilities between M0 and M1 is larger than 10%. The host maintains the migration of Pre-copy. However, when the migration goes into M1, performing the calculations of convergent value and variation value, the host is stopped at M1. The reason for this is because the calculations of the probabilities at M1 and M2 in Figure 9 and 10 are less than 10%.

The host is stopped at M_1.

Figure 9. The probability under $\varepsilon_1 = 0.1$ and $\varepsilon_2 = 0.5$

Figure 10. The probability under $\varepsilon_1 = 0.1$ and $\varepsilon_2 = 0.7$

As shown in Figure 11 the ε_2 of bit error rates is larger, and most of "1s" has changed to "0s". This means that the host is performing more "**READ**" than "**WRITE**" at the memory pages. It is observed that the method used in the GE model is stopping the host earlier than the method used in the Game Theory. The result indicates that the host has stopped at M_0, and the method used in the Game theory has stopped the host later at M_1.

Figure 11. The probability under $\varepsilon_1 = 0.1$ and $\varepsilon_2 = 0.9$

Based on the results shown in Figure 7 Figure 11 above, it can be observed that there is no difference on the method used in the Game Theory and the method used in the GE model for the condition $\varepsilon_1 = 0.1$. The method used in the Game Theory will not cause the host move on to the Stop-and-copy Phase later than the method used in the GE model. However, for $\varepsilon_2 = 0.9$, it is noticed that the method used in the GE model respond as faster than in the Game Theory. From Figure 11 it is shown that most of the actions performed are "**READ**" actions. This indicates that most of the memory states are as "0s". The "CUT-OFF" point for stopping the host, using the method with the GE model will take less time than the method with the Game Theory. Except this case, it is also noticed that there is no difference under $\varepsilon_1 = 0.1$ condition.

4.2 Random Memory State under $\varepsilon_1 = 0.5$

4.2.1 Parameters Definition

In this simulation, the parameter setting shown in Table 9. The memory states are same used in previous settings that are, "0" for "**READ**" operation and "1" for "**WRITE**" operation. In addition, the memory states are generated by programming the random process and the number of memory pages. As following the condition above, the migration varies for the condition $\varepsilon_1 = 0.5$, $\varepsilon_2 = 0.1$, 0.3, 0.5, 0.7 and 0.9 are shown as below, where ε_1 and ε_2 are bit error rates in the GE model.

Table 9. Parameter list for the simulation under $\varepsilon_1 = 0.5$

Parameter	Value
Memory State	0 or 1
Memory Statue Generated	Random
Number of Migration Iterations	9
Number of Memory Pages	1000
ε_1 of Bit Error Rates	0.5
ε_2 of Bit Error Rates	0.1, 0.3, 0.5, 0.7, 0.9

4.2.2 Simulation Result

Following the parameter setting in Table 9, the method used in the Game Theory is compared with the method used in the GE Model shown in Figure 12 to Figure 16. The preliminary probability and secondary probability respectively are also shown in the figures.

As shown in Figure 12 and 13, the default value is not being considered. The "CUT-OFF" point is also visible in both Figure 12 and 13. However, when ε_1 reaches to 0.5, the amount used is large than the previous, that is, $\varepsilon_1 = 0.1$. The "CUT-OFF" point is at M_1 for the GE model and it occurs later compared with the method used in the Game Theory at M0. The method used in the Game Theory can be used to stop the host and move into the Stop-and-copy Phase.

Figure 12. The probability under $\varepsilon_1 = 0.5$ and $\varepsilon_2 = 0.1$

Figure 13. The probability under $\varepsilon_1 = 0.5$ and $\varepsilon_2 = 0.3$

As shown in Figure 14 and Figure 15, the condition $\varepsilon_1 = 0.5$ is larger than previous value which is $\varepsilon_1 = 0.1$. However, as increases, both the methods used in the Game Theory and the GE model have the same iteration M0 to stop the original host and go directly into the Stop-and-copy Phase. This situation indicates as long as ε_2 is going to be larger, which means when some of "1s" in memory pages are changed to the "0s", the Game Theory method can obtain the same "CUT-OFF" point during the iterations.

Figure 14. The probability under $\varepsilon_1 = 0.5$ and $\varepsilon_2 = 0.5$

Figure 15. The probability under $\varepsilon_1 = 0.5$ and $\varepsilon_2 = 0.7$

With the reference to Figure 16, it is observed when reaches ε_2 to 0.9, the "CUT-OFF" point is obtained at M1, no matter what the method is being used.

Based on the results shown above, Figure 12 and Figure 13 indicates that the "CUT-OFF" point can be calculated faster by using Game Theory. This justifies using the Game Theory method, assuming that is going to increase, the Game Theory method is still a better choice for predicting the "CUT-OFF" point. This is because as increases, it is still possible to obtain the same "CUT-OFF" point. The host should be stopped

Figure 16. The probability under $\varepsilon_1 = 0.5$ and $\varepsilon_2 = 0.9$

and go directly into the Stop-and-copy Phase. However, there is an exception in Figure 11. When the memory states are "0" for "**READ**", only few memory states are "1" for "**WRITE**". The iteration of the method in the Game Theory is larger than the method in the GE model only once.

4.3 Random Memory State under $\varepsilon_1 = 0.9$

4.3.1 Parameters Definition

In this simulation, the parameter values used are shown as Table 10. The memory state set is defined in [7] which "0" or "**READ**" and "1" for "**WRITE**". As with the previous examples, the memory states are generated by programming the random process and the number of memory pages. Following the condition above, the migration varies for $\varepsilon_1 = 0.9$ with $\varepsilon_2 = 0.1$, 0.3, 0.5, 0.7 and 0.9 are shown as below, where ε_1 and ε_2 are bit error rates in the GE model.

Table 10. Parameter list for the simulation under $\varepsilon_1 = 0.9$

Parameter	Value
Memory State	0 or 1
Memory Statue Generated	Random
Number of Migration Iterations	9
Number of Memory Pages	1000
ε_1 of Bit Error Rates	0.9
ε_2 of Bit Error Rates	0.1, 0.3, 0.5, 0.7, 0.9

4.3.2 Simulation Result

Following the parameter setting in shown Table 10, it is possible to compare the method which used in the Game Theory with the method used in the GE Model. The results are shown in Figure 17 to Figure 21, where the preliminary probability and secondary probability are also shown in the figures.

In Figure 17 Figure 20 it can be seen when the bit error rate ε_1 is larger than the value of ε_2, the variation used the method in the Game Theory is smaller than the method in the GE model initially. The method used in the GE model cannot converge at the preliminary probabilities between M0 and M1. If the GE model is used, the "CUT-OFF" point at M1 can be obtained, rather than at M0. However, if the method is changed

Figure 17. The probability under $\varepsilon_1 = 0.9$ and $\varepsilon_2 = 0.1$

Figure 18. The probability under $\varepsilon_1 = 0.9$ and $\varepsilon_2 = 0.3$

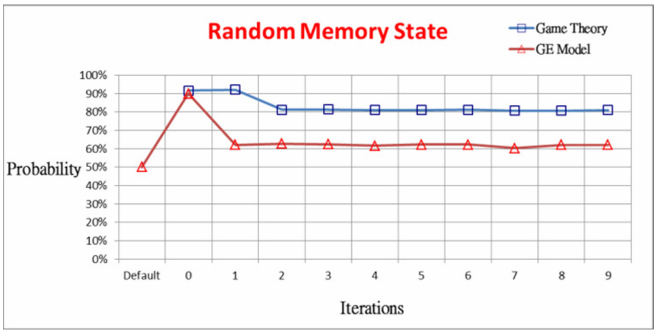


Figure 19. The probability under $\varepsilon_1 = 0.9$ and $\varepsilon_2 = 0.5$

Figure 20. The probability under $\varepsilon_1 = 0.9$ and $\varepsilon_2 = 0.7$

to use Game theory, the iterations between M0 and M1 converge. With this method, the "CUT-OFF" point can be obtained at M0, and stopping the host and going directly into the Stop-and-copy Phase.

As shown in Figure 21 when ε_2 reaches 0.9, this is, equal to ε_1, the preliminary probabilities trend to be convergent values. The method used in the Game Theory at this stage can obtain the same "CUT-OFF' point as in the GE model at M0.

Figure 21. The probability under $\varepsilon_1 = 0.9$ and $\varepsilon_2 = 0.9$

Based on the results above, if the condition as ε_1 increases, which means most of the memory states from original "0" for "**READ**" operation are being changed to be "1" for "**WRITE**" operation, and only a few memory states are presented as "0s". Using the method in the Game Theory is faster than the method used in the GE model to obtain the "CUT-OFF' point. This result justified using the Game Theory model for the memory prediction with Pre-copy. With memory states, sometimes the memory states are presented as "0s", and sometimes the memory are presented as "1s". If most of the memory states are presented as "1s", the memory states "0s" presented are few. Most cases, the method used in the Game Theory is *faster* than using the GE model to stop the host and move into the Stop-and-copy Phase. If the memory states are presented as "0s" then they are larger than the memory "1" states. In most cases, the Game Theory will obtain the same "CUT-OFF" point as used the method in the GE model. Except the only one case when $\varepsilon_1 = 0.1$ and $\varepsilon_2 = 0.9$, the iteration in the Game Theory occurs later than the GE model to obtain the "CUT-OFF' point to stop the original host.

4.4 The Predicted M_{th} of Special Case 1

4.4.1 Parameters Definition

In this simulation, the parameter used is shown in Table11. All of memory states are set to "1" as listed in Table12 and the memory pages can be reduced to 10 pages. This is enough for the simulation since the memory states are all "1" in the memory pages. Since the memory states are not generated by random process, ε_1 and ε_2 and the relative differences are not listed. ε_1 and ε_2 are fixed as the default value, where $\varepsilon_1 = 0.1$ and $\varepsilon_2 = 0.9$ and $\varepsilon_1 << \varepsilon_2$ as defined by the GE model.

4.4.2 Simulation Result

Following the parameter settings shown in Table 11 and 12, a comparison is made with the method which used in the Game Theory and the GE Model. The comparison is shown in Figure 22. Note, the preliminary probability and secondary probability are included in the figure.

Table 11. Parameter list for the simulation of special case 1

Parameter	Value
Memory State	All 1
Memory Statue Generated	Fixed as Table 12
Number of Migration Iterations	9
Number of Memory Pages	10

Table 12. The memory list of special case 1

Page Iteration / Number	M0	M1	M2	M3	M4	M5	M6	M7	M8	M9
1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1	1

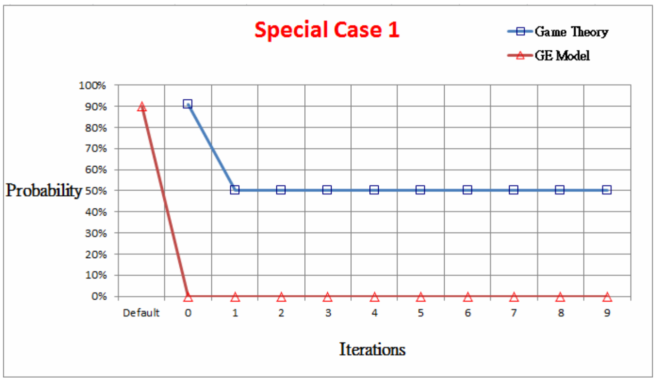


Figure 22. The probability of special case 1

With the reference to Figure 22, note that all memory states are set to "1" which indicate all the memory states are doing "WRITE" operation. The preliminary probability in the GE model shows that there is no change at each probability after the M0 iteration. This shows that the GE model can obtain 0% after the M0 iteration and the secondary probability set as a fixed value after M1. From Figure 22 it can be seen that the GE model can obtain the "CUT-OFF' point at M0, and the Game Theory can obtain the "CUT-OFF' point at M1.

Based on the result above, it is noticed that the GE model is faster than the Game Theory in computing the "CUT-OFF" point, which is used to stop the host and go directly into the Stop-and-copy Phase. As shown in Table 12 shown all actions performed are "**WRITE**" operation, all of memory states are set to "1s". It is also observed that the GE model takes less time to compute the "CUT-OFF" point when compared to using the Game Theory model.

4.5 The Predicted M_{th} of Special Case 2

4.5.1 Parameters Definition

In this simulation, the parameter used is shown in Table13. The memory list is shown in Table14. The memory pages is reduced to 10 pages that is enough for the simulation since the memory states are all fixed in the memory pages. Since the memory states are not generated by random process, the ε_1 and ε_2 and the relative differences are not listed. The ε_1 and ε_2 are fixed as the default value, where ε_1 = 0.1 and ε_2 =0.9 with $\varepsilon_1 << \varepsilon_2$. This is set according to the GE model.

4.5.2 Simulation Result

Following the parameter setting shown in Table 13 and Table 14, a comparison is made with the method used in the Game Theory and the GE Model. The preliminary probability and secondary probability are shown in the Figure 23.

Table 13. Parameter list for the simulation of special case 2

Parameter	Value
Memory State	0 or 1
Memory Statue Generated	Fixed as Table 14
Number of Migration Iterations	9
Number of Memory Pages	10

Using the memory states in Table 14, with the reference to Figure 23, it is observed that the Game Theory model can obtain the "CUT-OFF" point at M0, which is used to stop the host and move on to the Stop-and-copy Phase. However, using the GE model, the "CUT-OFF" point cannot be obtained. The result

Table 14. The memory list of special case 2

Page Iteration / Number	M0	M1	M2	M3	M4	M5	M6	M7	M8	M9
1	0	1	0	1	0	1	0	1	0	1
2	0	1	0	1	0	1	0	1	0	1
3	0	1	0	1	0	1	0	1	0	1
4	0	1	0	1	0	1	0	1	0	1
5	0	1	0	1	0	1	0	1	0	1
6	0	1	0	1	0	1	0	1	0	1
7	0	1	0	1	0	1	0	1	0	1
8	0	1	0	1	0	1	0	1	0	1
9	0	1	0	1	0	1	0	1	0	1
10	0	1	0	1	0	1	0	1	0	1

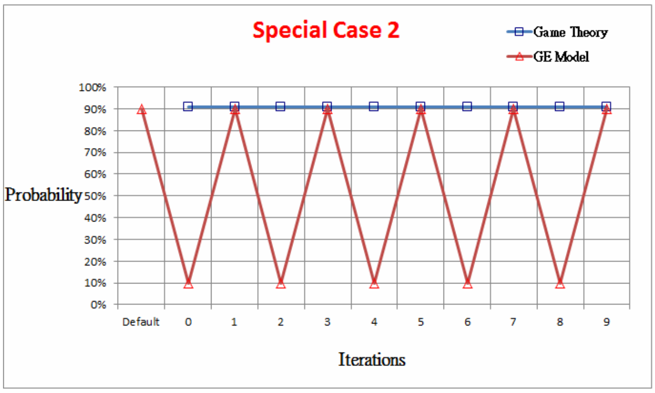


Figure 23. The probability of special case 2

shows that the GE model causes the probability increase to 90% at the default point. At M0, the probability goes down to 10%, and then it repeats again. The results show that the GE model cannot be used to obtain the "CUT-OFF" point in this special case.

Based on the result above, it is observed that using the Game Theory model is faster than using the GE model to obtain the "CUT-OFF" point. The reason is because using the GE model cannot get the probabilities to converge, which means the "CUT-OFF" point cannot be found by using the GE model. This special case also explains the benefit of using the Game Theory model. When the memory states are presented as changed in migration iterations, using the Game Theory model is more suitable than using the GE model.

4.6 The Predicted M_{th} of Special Case 3

4.6.1 Parameters Definition

In this simulation, parameter values are shown in Table15. The memory list is shown in Table16. The memory pages can be reduced to 10 pages which is adequate for the simulation since the memory states are all fixed in the memory pages. Since the memory states are not generated by random process, the ε_1 and ε_2 and the relative differences are not listed. The ε_1 and ε_2 can be fixed as the default value, where $\varepsilon_1 = 0.1$ and $\varepsilon_2 = 0.9$ as defined in the GE model with $\varepsilon_1 << \varepsilon_2$.

4.6.2 Simulation Result

Following the parameter setting shown in Table 15 and Table 16. The method used in the Game Theory and the GE Model are compared and the results are shown in Figure 24. The preliminary probability and secondary probability are included in the figure.

Table 15. Parameter list for the simulation of special case 3

Parameter	Value
Memory State	0 or 1
Memory Statue Generated	Fixed as Table 16
Number of Migration Iterations	9
Number of Memory Pages	10

Table 16. The memory list of special case 3

Page Iteration / Number	M0	M1	M2	M3	M4	M5	M6	M7	M8	M9
1	0	1	0	1	0	1	0	1	0	1
2	1	0	1	0	1	0	1	0	1	0
3	0	1	0	1	0	1	0	1	0	1
4	1	0	1	0	1	0	1	0	1	0
5	0	1	0	1	0	1	0	1	0	1
6	1	0	1	0	1	0	1	0	1	0
7	0	1	0	1	0	1	0	1	0	1
8	1	0	1	0	1	0	1	0	1	0
9	0	1	0	1	0	1	0	1	0	1
10	1	0	1	0	1	0	1	0	1	0

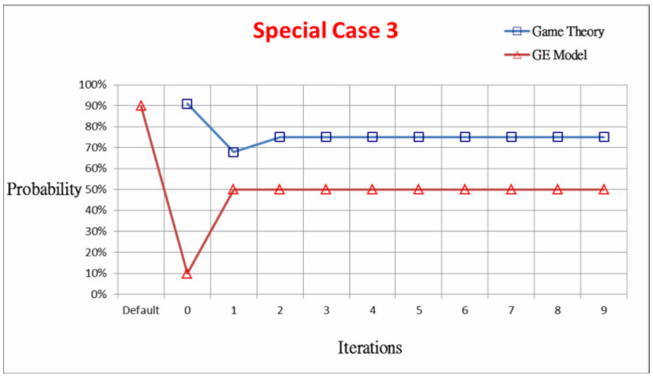


Figure 24. The probability of special case 3

With the reference to Figure 24, it can be seen that when the memory states are continuously varied at the different iterations in Table 16, useing both methods can obtain the same "CUT-OFF" point at M1 then stop the host and go directly into the Stop-and-copy Phase. Note that this result is not same as in special case 2. In this simulation, the memory state changes in the different columns. In addition, the different page numbers, and the memory states are being changed as well.

Based on the result above, it can be seen that the memory states are not only being changed at the different iterations, but also being changed at the different page numbers. The "CUT-OFF" point can be obtained at M1. However, special case 2 shows that the Game Theory model is faster than the GE model. The reason for this is that in special case 2 and 3, the "0" and "1" are respectively different. However, special case 2 shows all "0" or all "1" are listed in the same column, and special case 3 does not. Special case 3 has different "0s" and "1s" listed in the same column, which means each column that contains the same number of "0s" and "1s". The preliminary probability can obtain the convergent value. This explains why there are two different results.

5 Conclusion and Future Work

In the Pre-copy, the challenge is how to ensure the existing memory page can be securely migrated to the alternative host without wasting unnecessary migration time. This is because the migrated memory page will need to be re-migrated as long as the memory page in original host is verified as dirty. The destined memory page cannot be employed anymore. Thus, the goal is to obtain an exact "CUT-OFF" time in order to go the Stop-and-copy Phase.

In [8], it is previously proposed to use the GE model for improving the defect of utilizing the method of Markov process. That is because in [7], Lin considered the case of "Correct Memory Page" →"Dirty Memory Page", however, the case of "Dirty Memory Page" → "Correct Memory Page" was not considered in the method. In [8], many iterations are needed in preparing the collection of the dirtied page information. This method takes up unnecessary time due to the iterative actions.

In this paper, a method is proposed that adding the expected value of Game theory into the predicted calculation. The goal is to try to compute a "CUT-OFF" point fast under variable memory states which can stop the iteration and go directly into the Stop-and-copy Phase. The definition of variable memory state is the memory state that causes higher memory variation during the iteration.

In the simulation results presented, it is observed that the Game Theory model is faster than the GE model when the CUT-OFF point is computed. Besides, with special case 2, it is noticed that using the GE model, the probabilities cannot be convergent and it cannot compute the "CUT-OFF" point. However the "CUT_OFF" can only be computed by the Game Theory. This proves that the Game Theory model is suitable with cases of varied memory states. On the stable memory states, when the memory of original host are stable to perform "0" or "1" in the memory states, used with the method in the Game theory will be later than the method in the GE model one iteration. Except the stable memory cases, the Game Theory model is suitable for the prediction of Pre-copy. Thus, it is more practical to use the Game Theory model with the varied memory states to compute the "CUT-OFF" point than using the GE model.

The method proposed in this paper uses the Game Theory model to predict the "CUT-OFF" point, and uses this to determine when to stop the host and go directly into the Stop-and-copy Phase. However, using the Game Theory with stable memory is presented as slower than using the GE model. For future work, this paper suggests to use a new method that can cover both cases of stable memory states and varied memory states.

Acknowledgements

This work is supported partially by the Ministry of Science and Technology, Taiwan, R.O.C., under the grant No. MOST 108-2221-E-032-022.

References

[1] F. Rastegar, M. D. T. Fooladi, Online Virtual Machine Assignment Using Multi-Armed Bandit in Cloud Computing, *International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, Faridabad, India, 2019, pp. 40-46.

[2] M. Lee, S. Park, Y. Song, Y.-I. Eom, Introspection of Virtual Machine Memory Resource in the Virtualized Systems, *IEEE International Conference on Big Data and Smart Computing (BigComp)*, Kyoto, Japan, 2019, pp. 1-4.

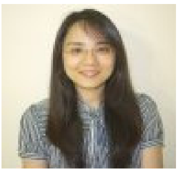
[3] P. P. Thakre, V. N. Sahare, VM Live Migration Time

Reduction Using NAS Based Algorithm during VM Live Migration, *2017 Third International Conference on Sensing, Signal Processing and Security (ICSSS)*, Chennai, India, 2017, pp. 242-246.
[4] K. Chanchio, P. Thaenkaew, Performance Comparisons and Data Compression of Time-bound Live Migration and Pre-copy Live Migration of Virtual Machines, *2017 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, Kanazawa, Japan, 2017, pp. 363-368.
[5] Z. Shan, J. Qiao, S. Lin, Fix Page Fault in Post-copy Live Migration with RemotePF Page Table Assistant, *2018 17th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)*, Wuxi, China, 2018, pp. 40-43.
[6] D. Fernando, J. Terner, K. Gopalan, P. Yang, Live Migration Ate My VM: Recovering a Virtual Machine after Failure of Post-Copy Live Migration, *IEEE INFOCOM-IEEE Conference on Computer Communications*, Paris, France, 2019, pp. 343-351.
[7] K. Chanchio, J. Yaothanee, Efficient Pre-Copy Live Migration of Virtual Machines for High Performance Computing in Cloud Computing Environments, *2018 3rd International Conference on Computer and Communication Systems (ICCCS)*, Nagoya, Japan, 2018, pp. 497-501.
[8] T.-Y. Wu, W.-T. Lee, J.-S. Huang, C.-Y. Duan, T.-W. Suen, An Enhanced Approach for Estimating Writable Working Set by Pre-copy and Prediction, *The International Conference on Information Networking (ICOIN)*, Bangkok, Thailand, 2013, pp. 227-231.
[9] Y.-W. Lin, *The Improvement of Predicted Probability with Cloud Computing Memory Live Migration*, Master's Thesis, Department of Electrical Engineering, Tamkang University, New Taipei City, Taiwan, 2013.
[10] H.-H. Wang, *Improving the Live Migration of Virtual Machines with Hybrid Copy Architecture*, Master's Thesis, Tatung University, Taipei, Taiwan, 2012.
[11] R. Meshram, Power Control Over Gilbert-Elliot Channel With No Observable States, *2016 Twenty Second National Conference on Communication (NCC)*, Guwahati, India, 2016, pp. 1-6.
[12] C.-M. Yeh, *Robust Vector Quantization for Burst Error Channels*, Master's Thesis, Department of Electrical Engineering, Chung Yuan Christian University, Taoyuan, 2002.
[13] D. T. Gillespie, Markov *Processes: An Introduction for Physical Scientists*, Academic Press, 1992.
[14] W.-C. Wong, W.-T. Lee, H.-W. Wei, Y.-C. Yang, V.-V. Yap, A Novel Bilateral Incentive Mechanism Based on Social Relation and Evolutionary Game Theory, *Journal of Internet Technology*, Vol. 20, No. 5, pp. 1441-1452, September, 2019.
[15] C. F. Camerer, Progress in Behavioral Game Theory, *The Journal of Economic Perspectives*, Vol. 11, No. 4, pp. 167-188, Fall, 1997.
[16] W.-J. Luo, K.-C. Wu, S.-Y. Tsau, Gender Stereotype of Male Nurse in a Virtual Reality Game: Exploring the Effect of MBTI in Decision-making Process through Game Theory, *2018 IEEE International Conference on Applied System Invention (ICASI)*, Chiba, Japan, 2018, pp. 418-421.
[17] M. M. Esfahani, A. Hariri, O. A. Mohammed, Game-theory-based Real-Time Inter-Microgrid Market Design Using Hierarchical Optimization Algorithm, *2018 IEEE Power & Energy Society General Meeting (PESGM)*, Portland, OR, USA, 2018, pp. 1-5.
[18] G.-F. Sun, J.-H. Gu, J.-H. Hu, T.-H. Zhao, Improvement of Live Memory Migration Mechanism for Virtual Machine Based on Pre-copy, *Computer Engineering*, Vol. 37, No. 13, pp. 36-39, July, 2011.

Biographies

Yao-Chiang Yang received his MS degree in Electrical Engineering from Tamkang University, Taiwan. He is currently a Ph.D. student in the Department of Electrical and Computer Engineering, Tamkang University, New Taipei City, Taiwan, His research interests include wireless networking, and computer networks.

Wei-Tsong Lee received his B.S., M.S. and Ph.D. degrees in Electrical Engineering from National Cheng Kung University, Tainan, Taiwan. He is currently a professor in the Department of Electrical and Computer Engineering, Tamkang University. His research interests include computer architecture, micro-processor interface and computer networks.



Hsin-Wen Wei received the Ph.D. degree in Computer Science from National Tsing Hua University, Hsinchu, Taiwan in 2008. She is currently an Associate Professor in the Department of Electrical and Computer Engineering, Tamkang University. Her research interests include storage systems, cloud computing, wireless networking, and real-time systems.

Vooi Voon Yap is currently an Associate Professor and Dean of Faculty of Engineering and Green Technology at Universiti Tunku Abdul Rahman (UTAR). He received his Ph.D. in Wavelet-based Image Compression for Mobile Devices from Middlesex University, London. He has over 30 years teaching experience in colleges and universities both in the United Kingdom and Malaysia, with special interest in embedded systems and video surveillance. He is also a Fellow of the Institution of Engineering and Technology (FIET) and a Charter Engineer (CEng).