อัลกอริทึมการแทนที่หน่วยความจำที่ถูกใช้น้อยที่สุดแบบไดนามิกสำหรับเว็บแคชชิ่ง

Dynamic Least-Recently-used Memory Replacement Algorithm for Web Caching

สุรพงษ์ รามัญจิตต์[1*] และ ดวงอาทิตย์ ศรีมูล[2]

Surapong Ramanjit[1*] and Duang-arthit Srimoon[2]

[1]นักศึกษาปริญญาโท หลักสูตรวิศวกรรมศาสตรมหาบัณฑิต วิทยาลัยวิศวกรรมศาสตร์ มหาวิทยาลัยรังสิต*
ถนนพหลโยธิน ตำบลหลักหก อำเภอเมือง จังหวัดปทุมธานี 12000
[2]อาจารย์ประจำ หลักสูตรวิศวกรรมศาสตรมหาบัณฑิต มหาวิทยาลัยรังสิต
ถนนพหลโยธิน ตำบลหลักหก อำเภอเมือง จังหวัดปทุมธานี 12000
[1]Graduate student in Master of Engineering (Electrical and Computer Engineering) of College of Engineering, Rangsit University,*
Phahonyothin Rd., Lak-hok, Patumtanee, Thailand 12000
[2]Lecturer in Master of Engineering (Electrical and Computer Engineering) of College of Engineering, Rangsit University,
Phahonyothin Rd., Lak-hok, Patumtanee, Thailand 12000
[*]Corresponding author, E mail: surapong_ram@hotmail.com

บทคัดย่อ

วัตถุประสงค์ของงานวิจัยนี้คือการเพิ่มอัตราการพบของหน่วยความจำเว็บ โดยที่งานวิจัยดังกล่าวได้มีการพัฒนาอัลกอริทึมการสับเปลี่ยนแบบใช้งานน้อยที่สุดออกก่อนไดนามิก (Dynamic Least Recently Used :DLRU) โดยการใช้หลักการการหาคำตอบที่ดีที่สุดด้วยวิธีการคณิตศาสตร์และสถิติ ซึ่งได้แก่ การประมาณค่าคาดหวัง การหาค่าในช่วง และคิวบิกส์สไปล์น ยิ่งไปกว่านั้น อัลกอริทึมนี้ยังสามารถตัดสินใจการเลือกเว็บต่าง ๆ ซึ่งคาดว่าน่าจะถูกร้องขอในอนาคตได้ อัลกอริทึมนี้ได้ถูกทดสอบด้วยชุดข้อมูลจากมหาวิทยาลัยแห่งหนึ่งในประเทศไทย ผลการทดลองของงานวิจัยนี้สามารถอธิบายได้ว่า อัลกอริทึม DLRU ให้ค่าอัตราการพบสูงสุดที่ 72.56 เปอร์เซ็นต์ ในขณะที่อัลกอริทึม LRU ให้ค่าอัตราการพบสูงสุดที่ 53.34 เปอร์เซ็นต์ อย่างไรก็ตามค่าเฉลี่ยของอัตราการพบของอัลกอริทึม DLRU อยู่ที่ 53.8 เปอร์เซ็นต์ แต่ค่าเฉลี่ยของอัลกอริทึม LRU อยู่ที่ 12.99 เปอร์เซ็นต์

คำสำคัญ: *อัลกอริทึมการแทนที่ข้อมูล, หน่วยความจำเว็บแคชชิ่ง, การใช้งานเว็บ, การสับเปลี่ยนแบบใช้งานน้อยที่สุดออกก่อน*

Abstract

The objective of this research is to increase the hit ratio of the web cache memory. This research investigates the Dynamic Least Recently Used *(*DLRU*)* by using the optimization with mathematical and statistical methods, which are as follows*:* estimated value, interpolation and cubic spline. In addition, this algorithm can decide to select web objects, which will be requested in the future. This algorithm is tested with the datasets from a university in Thailand. The experimental results of this research can apprize that the DLRU algorithm gives the maximum hit rate at 72.56 percent while the LRU algorithm offers the maximum hit rate at 53.34 percent. However, the average hit ratio of the DLRU algorithm is at 53.8 percent while the average hit ratio of the LRU algorithm amounts to about 12.99 percent.

Keywords: *Replacement algorithm, web caching memory, web usage, dynamic least recently used*

1. บทนำ

อินเทอร์เน็ตกำลังได้รับความนิยมอย่างมากในปัจจุบัน เทคโนโลยีที่ทันสมัยได้ถูกพัฒนาเพื่อเพิ่มประสิทธิภาพให้กับเครือข่ายเครือข่าย ยิ่งไปกว่านั้นเน็ตเวิร์คต่างๆ เหล่านี้ยังสามารถช่วยเพิ่มช่องทางการทำธุรกิจได้อีกด้วย ในหลาย ๆ ประเทศได้มีการทำธุรกิจผ่านทางอินเทอร์เน็ตแล้ว หรือแม้แต่มีการเรียนการสอนผ่านเครือข่ายนี้ด้วย ดังนั้นช่องสัญญาณของเครือข่ายจึงได้รับผลกระทบที่เกิดจากปริมาณการใช้งานเว็บตามไปด้วย

จำนวนการดาวน์โหลดเว็บไซต์ต่าง ๆ ในแต่ละวันทำให้เกิดการคับคั่งของช่องจราจรเครือข่าย ซึ่งเรียกว่าปัญหาคอขวด และบ่อยครั้งที่ผู้ใช้งานต่างๆ มักจะดาวน์โหลดเว็บไซต์ในเวลาเดียวกัน ซึ่งการดาวน์โหลดเว็บไซต์ในแต่ละครั้งนั้นจะมีค่าใช้จ่ายและเวลาในการดาวน์โหลดไปยังเครื่องที่ร้องขอทั้งสิ้น

หน่วยความจำเว็บแคชถูกใช้สำหรับลดจำนวนการดาวน์โหลดเว็บไซต์จากเซิร์ฟเวอร์ต่างๆ (Nguyen, 2008), (Cover, 2009) หน่วยความจำชนิดนี้สามารถที่จะบันทึกเว็บไซต์ต่างๆ ที่ดาวน์โหลดมาเก็บไว้ในหน่วยความจำได้ แต่อย่างไรก็ตามหน่วยความจำชนิดนี้มีเนื้อที่การจัดเก็บไม่มาก ดังนั้นหน่วยความจำแคชจึงจำเป็นต้องมีการย้ายเว็บไซต์บางเว็บไซต์ออกเมื่อเนื้อที่หน่วยความจำเต็ม เทคนิคต่าง ๆ ถูกนำมาใช้เพื่อการตัดสินใจว่าจะให้เว็บไซต์ใดต้องถูกกำจัดออกไป ซึ่งเทคนิคหนึ่งที่ได้รับความนิยมอยู่ในปัจจุบันคืออัลกอริทึม LRU เนื่องมาจากเป็นอัลกอริทึมที่มีลักษณะการทำงานไม่สลับซับซ้อน แต่อย่างไรก็ตามอัลกอริทึม LRU ก็ยังให้ค่าอัตราการพบไม่สูงมากนักโดยเฉลี่ยอยู่ที่ 40 เปอร์เซ็นต์

งานวิจัยนี้ได้พัฒนาอัลกอริทึมการแทนที่ข้อมูลของหน่วยความจำเว็บแคชซึ่งบนพื้นฐานของการใช้งานเว็บสำหรับเพิ่มอัตราการพบ (Hit rate) ของหน่วยความจำเว็บแคช โดยมีการนำหลักการทางคณิตศาสตร์และสถิติมาใช้ในอัลกอริทึมนี้ได้แก่ การประมาณค่าคาดหวัง การประมาณค่าในช่วง และคิวบิกส์สไปล์น โดยนำข้อมูลที่มีการเรียกใช้งานเว็บไซต์ในอดีตมาใช้เป็นชุดข้อมูล (Datasets) เพื่อใช้ตัดสินใจเลือกเว็บไซต์ที่คาดว่าน่าจะมีการเรียกใช้มากที่สุดในอนาคต หลังจากนั้นงานวิจัยนี้ก็จะทำการเปรียบเทียบอัตราการพบระหว่างอัลกอริทึม DLRU กับอัลกอริทึม LRU ในลำดับถัดไปจะเป็นการบรรยายเกี่ยวกับทฤษฎีและงานวิจัยที่เกี่ยวข้องต่าง ๆ

2. วัตถุประสงค์

บทความมีวัตถุประสงค์ต่าง ๆ ดังนี้

1. เพื่อเพิ่มอัตราการพบของหน่วยความจำเว็บแคชด้วยการพัฒนาอัลกอริทึมไดนามิกโปรแกรมมิ่งการสับเปลี่ยนแบบใช้งานน้อยที่สุดออกก่อน

(Dynamic Least Recently Used : DLRU) ขึ้นมาประยุกต์ใช้

2. เปรียบเทียบประสิทธิภาพระหว่างอัลกอริทึมที่ได้พัฒนาขึ้นกับอัลกอริทึมทั่วไป สำหรับการเข้าไปใช้งานเว็บไซต์

3. วิธีดำเนินการวิจัย

ก่อนอื่นจะขอกล่าวถึงทฤษฎีและงานวิจัยที่เกี่ยวข้องกับการทำวิจัยนี้ก่อน ซึ่งก่อนที่จะมีการอธิบายถึงการทำงานของอัลกอริทึม DLRU นั้น เพื่อให้ผู้อ่านได้เห็นภาพทฤษฎีต่างๆ ที่ต้องนำมาใช้ในการพัฒนาอัลกอริทึมดังกล่าวนั้น จึงมีความจำเป็นต้องอธิบายให้ทราบรายละเอียดของเทคนิคต่างๆ ซึ่งทฤษฎี หรือเทคนิค หรือวิธีการที่นำมาใช้ได้แก่

3.1 การประมาณค่าคาดหวัง

การประมาณค่าคาดหวัง (estimated value) ใช้สำหรับคำนวณหาค่าเฉลี่ยซึ่งหากเป็นการประมาณค่าคาดหวังด้วยตัวแปรสุ่มนั้นสามารถใช้วิธีการได้ในงานวิจัยของ Gerhard (Gerhard, 2010) การประมาณค่าคาดหวังมีได้กัน 2 รูปแบบ ได้แก่ การประมาณค่าคาดหวังจากตัวแปรสุ่มแบบไม่ต่อเนื่อง (Discrete random variable) และตัวแปรสุ่มแบบต่อเนื่อง (Continuous random variable) อย่างไรก็ตามงานวิจัยนี้เป็นการใช้ตัวแปรสุ่มแบบต่อเนื่องซึ่งสามารถแสดงได้สมการที่ 1

$$E(x) = \int_{1}^{\infty} xf(x)dx \qquad (1)$$

โดยที่ $E(x)$ คือค่าประมาณค่าคาดหวังตัวแปรแบบสุ่มของ x และ $f(x)$ คือฟังก์ชันความน่าจะเป็นของตัวแปรสุ่มของ x

3.2 การประมาณค่าในช่วงด้วยฟังก์ชันคิวบิกส์สไปลน์

การประมาณค่าในช่วง (interpolation) เป็นเทคนิคหนึ่งที่นำมาใช้ทำนายค่าบางค่าซึ่งเป็นค่าที่ขาดหายไป หรือค่าที่ไม่ได้มีการเก็บข้อมูล ณ ช่วงใดช่วงหนึ่ง (Gharles, 2013) นอกจากนี้การประมาณค่าในช่วงยังสามารถประมาณค่าบางค่าที่เท่ากับศูนย์ หรือว่าสามารถปรับเป็นค่าใดๆ ที่คาดว่ามีความเหมาะสมมากกว่า อย่างไรก็ตามการประมาณค่าในช่วงนั้นมีหลายฟังก์ชันที่ใช้สำหรับการหาค่าประมาณในช่วง แต่งานวิจัยนี้ใช้ฟังก์ชันคิวบิกส์สไปล์น (Cubic Spline) มาใช้สำหรับค้นหาค่าดังกล่าว

ฟังก์ชันคิวบิกส์สไปล์นเป็นฟังก์ชันหนึ่งที่สามารถเชื่อมโยงข้อมูลระหว่างข้อมูลไดเป็นอย่างดี ซึ่งฟังก์ชันนี้สามารถปรับค่าเส้นโค้งได้ด้วยการยกกำลังสาม (the third power) โดยที่คิวบิกส์สไปล์นนั้นสามารถเชื่อมต่อเส้นโค้งนี้จากจุดหนึ่งไปอย่างจุดหนึ่งได้ใกล้เคียงกับค่าความเป็นจริงมากที่สุดสำหรับข้อมูลของงานวิจัยนี้ (Akila, 2013) การทำให้เส้นกราฟมีความโค้งนี้นักวิทยาศาสตร์บางท่านเรียกฟังก์ชันหรือเทคนิคนี้คิวบิกส์ไปล์ธรรมชาติ (Natural Cubic Spline) ซึ่งสมการของฟังก์ชันแสดงไว้ในสมการที่ 2

$$f_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i \qquad (2)$$

โดยที่ $i = 1, 2, 3, \ldots, n$

3.3 การวัดอัตราการพบของหน่วยความจำเว็บแคช

การประเมินประสิทธิภาพของการแทนที่ข้อมูล (replacement evaluation) ของหน่วยความจำเว็บแคช มักจะใช้ฟังก์ชันอัตราการพบ (hit rate function) มาเป็นตัวพิจารณาประสิทธิภาพอัลกอริทึมต่างๆ ถ้าหากอัตราการพบมีเปอร์เซ็นต์ที่สูง นั่นหมายความว่าประสิทธิภาพของอัลกอริทึมนั้นมีประสิทธิภาพที่ดีสำหรับฟังก์ชันอัตราการพบที่ใช้โดยทั่วไปและงานวิจัยนี้ถูกแสดงไว้ในสมการที่ 3

$$HR = \frac{\sum Hi}{\sum Ri} \quad (3)$$

โดยที่ Hi คือจำนวนการร้องขอการใช้เว็บไซต์ทั้งหมด ณ เวลาหนึ่ง ซึ่งเว็บไซต์ที่ร้องขอนั้นอยู่ในหน่วยความจำเว็บแคช และ Ri คือจำนวนการร้องขอเว็บไซต์ทั้งหมดไม่ว่าจะพบหรือไม่พบในหน่วยความเว็บแคชหรือไม่

3.4 งานวิจัยที่สัมพันธ์

ผู้วิจัยได้มีการทบทวนวรรณกรรมจากงานวิจัยต่างๆ ที่เกี่ยวข้องกับเทคนิคการแทนที่ข้อมูลของหน่วยความจำเว็บแคช ซึ่งงานวิจัยจำนวนมากเป็นการประยุกต์ใช้อัลกอริทึม LRU เช่น งานวิจัยของ Fang, C. และคณะ (Fang, 2013) ได้ทำวิจัยการประยุกต์ LRU สำหรับบริหารจัดการหน่วยความจำขนาดใหญ่ที่ใช้เก็บเว็บไซต์ต่างๆ โดยที่การทำงานของวิธีการนี้จะเลือกเก็บเว็บไซต์ที่มีการเรียกใช้ ณ ปัจจุบันก่อน ข้อดีของเทคนิคนี้คือหากหน่วยความจำมีพื้นที่จัดเก็บปริมาณมาก นอกจากนี้ยังมีงานวิจัยเช่น William และคณะ (William, 2014) ได้พัฒนาการผสมผสานระหว่างหลายอัลกอริทึมและประยุกต์ใช้กับ GAM (Waleed, 2011) งานวิจัยของ Waleed และ Siti ได้พัฒนาเทคนิค ICWCS และสำหรับ Walled, A. และคณะ (Walled, 2012) ได้พัฒนาอัลกอริทึมโดยใช้ Naïve Bay ซึ่งงานวิจัยต่างๆ เหล่านี้ให้ค่าอัตราการพบโดยประมาณไม่เกิน 40 เปอร์เซ็นต์

งานวิจัยนี้ได้มีการเปรียบเทียบอัตราการพบกับอัลกอริทึม LRU โดยที่หลักการการทำงานของอัลกอริทึมนี้คือจะเลือกเว็บไซต์ที่ไม่ได้ถูกเรียกใช้งานนานที่สุดออกจากหน่วยความจำเว็บแคชก่อน หากหน่วยความจำเว็บแคชนั้นมีพื้นที่เต็มและมีการเรียกใช้งานเว็บไซต์อื่น ณ เวลาปัจจุบัน

3.5 การออกแบบอัลกอริทึมการสับเปลี่ยนแบบใช้งานน้อยที่สุดออกก่อนไดนามิก (DLRU อัลกอริทึม)

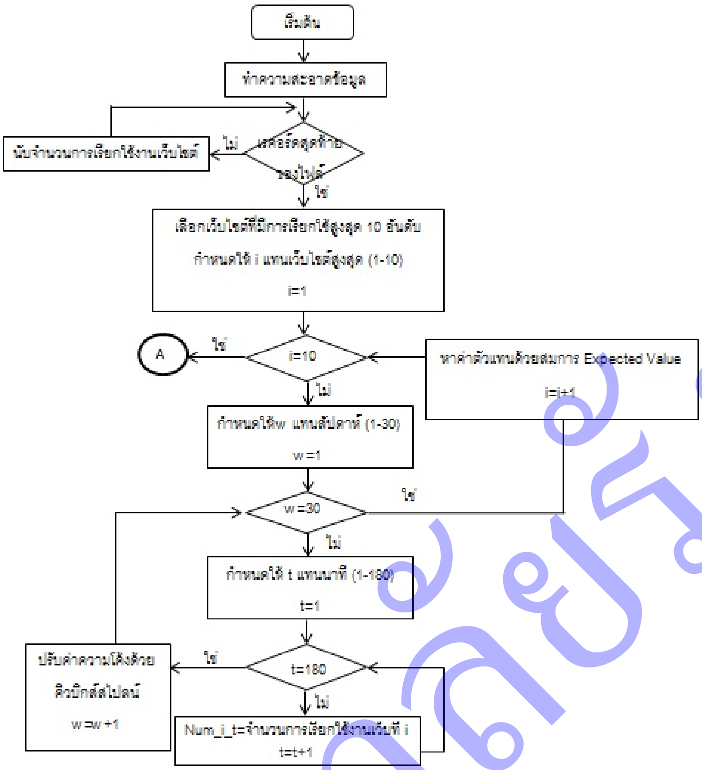
จากวัตถุประสงค์ของงานวิจัยนี้คือการเพิ่มอัตราการพบของหน่วยความจำเว็บแคช ซึ่งส่วนนี้เป็นการอธิบาย 2 ส่วนด้วยกัน ได้แก่ ประเภทข้อมูลที่นำมาใช้ในงานวิจัย และส่วนประกอบต่าง ๆ ของอัลกอริทึม DLRU

3.5.1 ประเภทของข้อมูลที่นำมาใช้ในงานวิจัย

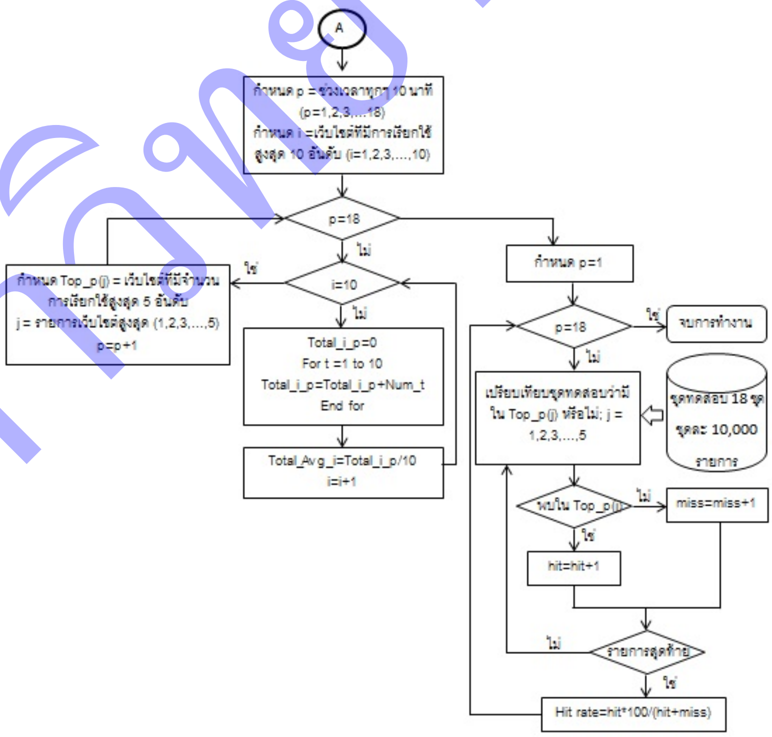
ชุดข้อมูลต่างๆ ของการเรียกใช้งานเว็บถูกรวบรวมด้วยโปรแกรมสควิด (squid program) ซึ่งโปรแกรมดังกล่าวเป็นซอฟต์แวร์ที่ใช้สำหรับบันทึกการเรียกใช้งานเว็บต่างๆ จากภายนอกของผู้ใช้งานทั้งหมดภายในองค์กร โดยโปรแกรมนี้จะบันทึกเว็บที่เรียกใช้ วันที่ เวลา และเบอร์ไอพีของเครื่องไคลเอนต์ ซึ่งโปรแกรมสควิดจะถูกติดตั้งอยู่ที่เซิร์ฟเวอร์กลางของมหาวิทยาลัยแห่งหนึ่งในประเทศไทย และงานวิจัยนี้ก็ได้นำชุดข้อมูลดังกล่าวมาใช้ในการทดลองนี้ด้วย อย่างไรก็ตามชุดข้อมูลเหล่านี้จะต้องผ่านการทำความสะอาดข้อมูลก่อนที่จะนำมาใช้งานกับอัลกอริทึม DLRU

3.5.2 อัลกอริทึมการสับเปลี่ยนแบบใช้งานน้อยที่สุดออกก่อนไดนามิก (DLRU)

อัลกอริทึมการแทนที่ข้อมูลของหน่วยความจำเว็บแคชซึ่งบนพื้นฐานของการใช้งานเว็บถูกพัฒนาขึ้นจากงานวิจัยนี้ อัลกอริทึม DLRU ได้ถูกออกแบบเพื่อเพิ่มประสิทธิภาพของอัตราการพบและยังใช้สามารถประเมินว่าเว็บไซต์ใดที่คาดว่าจะถูกร้องขอในอนาคตได้อีกด้วย ถ้า DLRU สามารถคาดการณ์การร้องขอที่คาดว่าน่าจะเกิดขึ้นในอนาคตได้อย่างถูกต้องแม่นยำ นั่นหมายความว่าอัตราการพบก็จะเพิ่มสูงขึ้นและการดาวน์โหลดเว็บไซต์จากเซิร์ฟเวอร์ต่างๆ ก็จะลดลงไปด้วย



(ก)



(ข)

รูปที่ 1 กระบวนการทำงานของอัลกอริทึม DLRU

โมดูลการทำงานของอัลกอริทึม DLRU ประกอบด้วย 3 โมดูล ได้แก่ 1) โมดูลการจัดเตรียมข้อมูล (The data preparation module) 2) โมดูลการจัดเตรียมการทำงานแคช (The cache feature preparation module และ 3) โมดูลการแทนที่ข้อมูลของแคช (The cache replacement module)

1) โมดูลการจัดเตรียมข้อมูลคือกระบวนการทำความสะอาดข้อมูลก่อนที่จะนำข้อมูลเหล่านั้นมาประมวลด้วยอัลกอริทึม DLRU

ขั้นตอนแรกของโมดูลนี้จะทำการแปลงข้อมูลจากไฟล์ log ต่างๆ ด้วยภาษา Perl เพื่อแปลงโค้ดให้เป็นข้อมูลที่เข้าใจได้ เช่น วัน เวลา และเว็บไซต์ที่มีการร้องขอ และหากข้อมูลไม่มีความสมบูรณ์บางรายการก็จะถูกกำจัดออกในขั้นตอนนี้

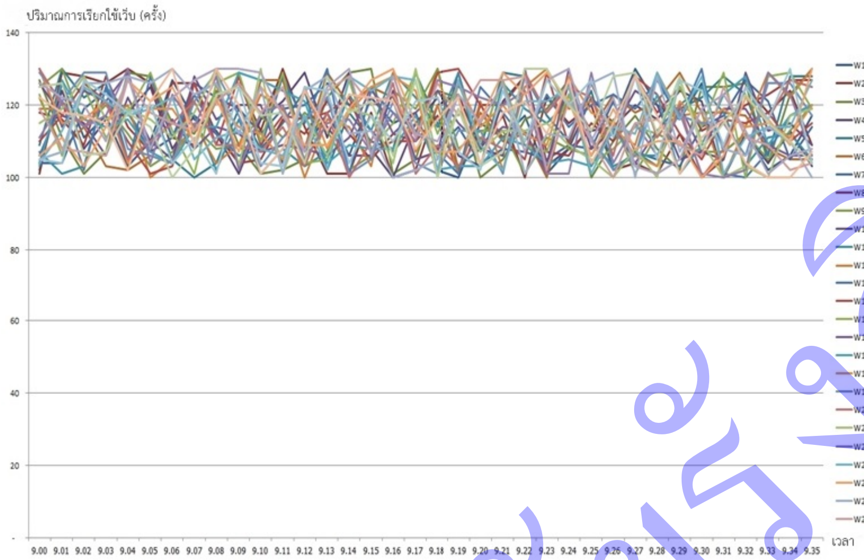
ขั้นตอนที่ 2 คือการคำนวณจำนวนปริมาณการเรียกใช้เว็บไซต์ในแต่ละเว็บไซต์ในช่วงวันจันทร์ถึงวันศุกร์ระหว่างเวลา 9.00 น. – 12.00 น. เป็นระยะเวลา 30 สัปดาห์ สาเหตุที่เลือกช่วงระยะเวลา 9.00-12.00 น. เนื่องจากมีปริมาณการใช้งานเว็บไซต์ในแต่ละวันมากที่สุด และช่วงเวลาอื่นๆ นั้นมีการเรียกใช้งานเว็บจำนวนน้อย เมื่อสิ้นสุดในขั้นตอนนี้จะทำให้ทราบว่าปริมาณการเรียกใช้งานเว็บไซต์มากที่สุด 10 อันดับแรกคือเว็บไซต์ใดบ้าง ซึ่งจะเลือกเว็บไซต์ที่ต้องการคาดการณ์จากข้อมูลการเรียกใช้งานเว็บไซต์มากที่สุด 10 อันดับในทุกๆ การคาดการณ์ต่อไป

ขั้นตอนที่ 3 ทำการหาปริมาณการใช้งานเว็บไซต์แต่ละเว็บไซต์ของ 10 อันดับการใช้งานมากที่สุด โดยให้หาปริมาณการใช้ทุก 1 นาที เริ่มตั้งแต่ 9.00 น. – 12.00 น. ซึ่งจะได้ปริมาณการเรียกใช้งานเว็บดังตัวอย่างในตารางที่ 1

ตารางที่ 1 ตัวอย่างปริมาณการใช้งานเว็บไซต์หนึ่งจากเว็บไซต์ 10 อันดับแรกที่มีการเรียกใช้งานมากที่สุด

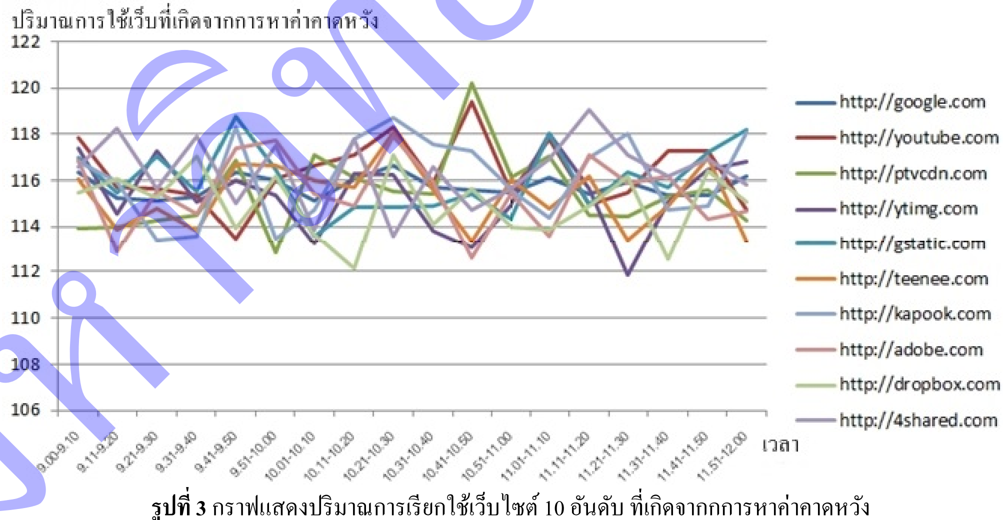
สัปดาห์ที่	**จำนวนการเรียกใช้งานเว็บระหว่างเวลา (ครั้ง) 9.00 น. – 12.00 น.**					
	9.00	**9.01**	**9.02**	**9.03**	**.**	**12.00**
1	1,200	1,204	1,204	1,265	.	0
2	1,234	1,303	1,303	1,324	.	45
3	1,505	2,500	2,500	2,304	.	1,003
.	.	.	.	.	.	1,723
30	1,709	1,754	1,776	4,743	.	4,743

ตารางที่ 1 แสดงข้อมูลบางส่วนของเว็บไซต์หนึ่งจากเว็บไซต์ 10 อันดับแรกที่มีผู้ใช้งานเรียกใช้เว็บไซต์มากที่สุด ซึ่งหากนำมาพล็อตด้วยกราฟเส้นจะได้ดังรูปที่ 2



รูปที่ 2 กราฟแสดงปริมาณการเรียกใช้เว็บไซต์หนึ่งจาก 10 อันดับเว็บไซต์

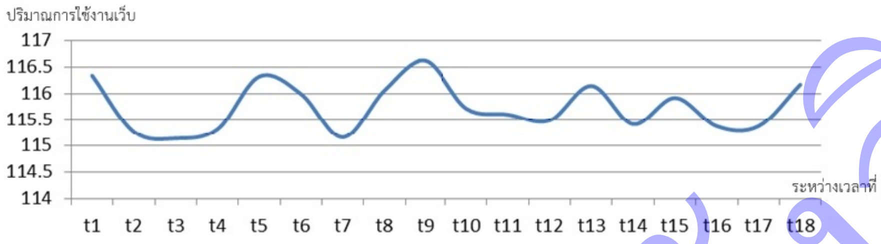
รูปที่ 2 คือกราฟเส้นที่ถูกทำจากจำนวนปริมาณการใช้งานเว็บหนึ่งจาก 10 อันดับที่มีปริมาณการใช้เว็บมากที่สุดจำนวน 30 สัปดาห์ ลำดับถัดไปจะทำการหาค่าตัวแทนจาก 30 สัปดาห์ด้วยฟังก์ชันการประเมินค่าคาดหวัง (Expected value) ดังนั้นหลังจากสิ้นสุดการทำงานนี้จะได้ค่าตัวแทนจากเว็บไซต์ 10 อันดับสูงสุด ดังแสดงในรูปที่ 3



รูปที่ 3 กราฟแสดงปริมาณการเรียกใช้เว็บไซต์ 10 อันดับ ที่เกิดจากกการหาค่าคาดหวัง

จากรูปที่ 3 แสดงปริมาณการใช้งานเว็บของเว็บที่มีการใช้งานมากที่สุด 10 อันดับ โดยที่เส้นกราฟ 1 เส้น แทนค่าปริมาณการใช้เว็บจำนวน 30 สัปดาห์ ซึ่งงานวิจัยนี้ได้ใช้ฟังก์ชันค่าคาดหวังมาทำการหาค่าตัวแทนเหล่านี้

ขั้นตอนที่ 4 งานวิจัยนี้ได้ใช้อัลกอริทึมคิวบิกส์สไปล์น เพื่อทำการปรับค่าตัวแทนให้เป็นเส้นโค้ง (Fitting curve) อย่างไรก็ตามเทคนิคนี้ให้ค่าที่ใกล้เคียงกับความเป็นจริงมากที่สุด ซึ่งตัวอย่างกราฟเส้นที่เกิดจากการใช้อัลกอริทึมนี้แสดงได้ดังรูปที่ 4



รูปที่ 4 ตัวอย่างกราฟเส้นที่เกิดจากการใช้อัลกอริทึมคิวบิกส์สไปล์น

2) โมดูลการเตรียมหน่วยความจำแคช โมดูลนี้มีกระบวนการทำงานที่สำคัญเพราะจะส่งผลต่อการตัดสินใจในการเลือกเว็บใดๆ เข้ามาเก็บไว้ในหน่วยความจำเว็บ สำหรับขั้นตอนการทำงานสามารถอธิบายได้ดังต่อไปนี้

ขั้นตอนที่ 1 คำนวณปริมาณการใช้งานเว็บเฉลี่ยทุก 10 นาที งานวิจัยนี้จะมีการคาดการณ์ปริมาณการใช้งานเว็บทุก 10 นาที ดังนั้นจึงทำการหาค่าเฉลี่ยทุกๆ 10 นาทีด้วย ในแต่เว็บไซต์ ซึ่งสมการในการหาค่าเฉลี่ยแสดงไว้ในสมการที่ 4

$$AVG(J) = \frac{\sum_{i}^{i+9} j_i}{i+9} \qquad (4)$$

โดยที่ i คือจำนวนช่วงระยะเวลา เช่น $i = 1, 11, 21, 31, \ldots, 171$ ส่วน j คือ จำนวนการใช้งานเว็บในแต่ละนาที

ขั้นตอนที่ 2 เปรียบเทียบค่าเฉลี่ยที่ได้จากเว็บไซต์สูงสุด 10 อันดับ แล้วเลือกค่าเฉลี่ยที่มีปริมาณมากที่สุด 5 อันดับ เพื่อเลือกรายการที่นำมาเก็บไว้ในหน่วยความจำเว็บแคชต่อไป

3) โมดูลการแทนที่ข้อมูลของแคช สำหรับการทำงานของโมดูลนี้จะทำการเลือกโหลดเว็บไซต์ 5 อันดับสูงสุด ที่ได้มาจากโมดูลก่อนหน้า ทุกๆ 10 นาที หลังจากนั้นก็จะทำการทดสอบปริมาณการใช้เว็บไซต์โดยใช้ชุดข้อมูลปริมาณการใช้งานเว็บชุดใหม่ในช่วงระยะเวลาทุก 10 นาที เพื่อทดสอบอัตราการพบ (Hit rate) ว่ามีค่าอัตราการพบมากน้อยเพียงใด

4. ผลการวิจัย

ในส่วนนี้เป็นการอภิปรายผลการทดลองที่เกิดจากการพัฒนาอัลกอริทึม DLRU ซึ่งสามารถอภิปรายผลได้ดังต่อไปนี้

จากตารางที่ 2 แสดงปริมาณการใช้งานเว็บสูงสุด 10 อันดับ ซึ่งจะนำข้อมูลเหล่านี้มาทำการคัดเลือกปริมาณการใช้งานเว็บสูงสุด 5 อันดับในทุกๆ 10 นาที เพื่อคาดการณ์ปริมาณการใช้งานเว็บไซต์ที่คาดว่าจะเกิดขึ้นหรือการหาค่าอัตราการพบนั่นเอง

ตารางที่ 2 การเรียกใช้งานเว็บมากที่สุด 10 อันดับแรก

ลำดับ	เว็บไซต์ที่มีปริมาณมากที่สุด 10 อันดับ	จำนวนครั้งที่เรียกใช้ภายใน 30 สัปดาห์	ร้อยละ
1	http://google.com	618,039	38.42
2	http://youtube.com	344,744	21.43
3	http://ptvcdn.com	204,118	12.69
4	http://ytimg.com	135,749	8.44
5	http://gstatic.com	73,394	4.56
6	http://teenee.com	68,053	4.23

7	http://kapook.com	51,448	3.20
8	http://adobe.com	45,196	2.81
9	http://dropbox.com	32,633	2.03
10	http://4shared.com	35,302	2.19
	จำนวนรวม	**1,608,676**	**100.00**

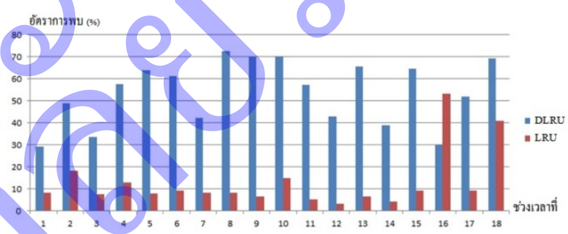
ตารางที่ 3 อัตราการพบทุก ๆ 10 นาทีของ DLRU

ช่วงเวลาที่	รายการที่ 1	รายการที่ 2	รายการที่ 3	รายการที่ 4	รายการที่ 5	จำนวนครั้ง	อัตราการพบ (%)
1	google	youtub	ptvcdn	yting	gstatic	3,276	29.15
2	google	youtub	ptvcdn	yting	gstatic	3,123	48.90
3	google	youtub	ptvcdn	adobe	dropbox	6,243	33.67
4	google	youtub	ptvcdn	gstatic	adobe	13,200	57.55
5	youtube	ptvcdn	ytimg	gstatic	dropbox	5,568	63.83
6	google	youtub	ptvcdn	ytimg	4shared	6,294	61.11
7	google	youtub	ptvcdn	adobe	4shared	3,621	42.34
8	google	youtub	ptvcdn	ytimg	4shared	4,329	72.56
9	google	youtub	ptvcdn	ytimg	dropbox	4,788	69.80
10	google	youtub	ptvcdn	ytimg	4shared	3,552	70.35
11	google	youtub	ptvcdn	ytimg	dropbox	3,513	57.30
12	google	youtub	ptvcdn	4shared	dropbox	3,534	42.95
13	google	youtub	ptvcdn	ytimg	gstatic	3,357	65.50
14	google	youtub	ptvcdn	ytimg	dropbox	3,768	38.77
15	google	youtub	ptvcdn	ytimg	dropbox	4,113	64.62
16	google	youtub	ptvcdn	ytimg	kapook	3,186	30.04
17	google	youtub	ptvcdn	ytimg	teenee	3,084	51.95
18	google	youtub	gstatic	ytimg	teenee	2,205	69.12

ตารางที่ 4 อัตราการพบทุก ๆ 10 นาทีของ LRU

ช่วงเวลาที่	จำนวนครั้งที่พบ	อัตราการพบ (%)	อัตราการพบ	จำนวนครั้งที่พบ	อัตราการพบ(%)
1	924	8.21	10	744	14.72
2	1,166	18.25	11	325	5.30
3	1,421	7.65	12	267	3.24
4	2,929	12.76	13	343	6.70
5	698	8.00	14	424	4.35
6	958	9.30	15	593	9.31
7	695	8.12	16	5,66	53.34
8	485	8.12	17	541	9.10
9	460	6.70	18	1,30	40.76

จากตารางที่ 3 และ 4 แสดงอัตราการพบของอัลกอริทึม DLRU และ LRU ซึ่งพบว่าอัลกอริทึมในค่าอัตราพบสูงสุดที่ 72.56 เปอร์เซ็นต์ ในขณะที่อัลกอริทึม LRU ให้อัตราการพบสูงสุดที่ 53.34 เปอร์เซ็นต์ และอัตราการพบเฉลี่ยของ DLRU มีค่าเฉลี่ยเท่ากับ 53.8 เปอร์เซ็นต์ ในขณะที่ LRU มีค่าเฉลี่ยเท่ากับ 12.99 เปอร์เซ็นต์ อย่างไรก็ตามอัตราการพบระหว่างอัลกอริทึม DLRU กับ LRU ในทุกช่วงเวลา (ทุกๆ 10 นาที) ได้แสดงไว้ในรูปที่ 5



รูปที่ 5 การเปรียบเทียบอัตราการพบระหว่าง DLRU กับ DLRU

5. การอภิปรายผล

งานวิจัยนี้ได้พัฒนาอัลกอริทึม DLRU ซึ่งเป็นการเพิ่มประสิทธิภาพการแทนที่ข้อมูลในหน่วยความจำเว็บแคช ซึ่งจากผลการทดลองสามารถสรุปได้ว่าอัลกอริทึม DLRU สามารถเพิ่มประสิทธิภาพการแทนที่ข้อมูลได้เป็นอย่างดี และมีประสิทธิภาพดีกว่าอัลกอริทึม LRU อย่างไรก็ตามอัลกอริทึมนี้เป็นการคัดเลือกเว็บไซต์ที่คาดการณ์ว่าจะมีปริมาณการเรียกใช้งานเว็บไซต์ที่คาดว่าจะเกิดขึ้นในอนาคตจากการนำข้อมูลการเรียกใช้งานเว็บในอดีต ซึ่งมีขั้นตอนการคัดเลือกเว็บไซต์ที่ค่อนข้างยุ่งยากและใช้เวลานานพอสมควร นอกจากนี้อัลกอริทึม DLRU จะทำให้อัตราการพบ (Hit rate) มีเปอร์เซ็นต์สูงกว่าอัลกอริทึม LRU หากผู้ใช้งานมีการเรียกใช้งานเว็บไซต์ในแต่ละช่วงเวลาที่แตกต่างกันไป แต่ถ้าผู้ใช้งานไม่ได้มีการ

เปลี่ยนแปลงเว็บไซต์ใด ๆ เป็นระยะเวลานาน ๆ จะทำให้อัตราการพบของ DLRU มีเปอร์เซ็นต์ต่ำกว่า LRU

6. บทสรุป

อัลกอริทึม DLRU เป็นอัลกอริทึมที่ถูกนำมาใช้ในการเพิ่มประสิทธิภาพเทคนิคการแทนที่ข้อมูลเว็บ (Web data replacement) ภายในหน่วยความจำเว็บแคช ซึ่งเป็นหน่วยความจำที่อยู่ในเครื่องเซิร์ฟเวอร์ อย่างไรก็ตามงานวิจัยนี้ได้สร้างแบบจำลอง เพื่อทดสอบอัตราการพบ (Hit rate) อัลกอริทึม DLRU นอกจากนี้แบบจำลองยังสามารถเปรียบเทียบประสิทธิภาพอัลกอริทึม DLRU กับอัลกอริทึม LRU อีกด้วย ซึ่งงานวิจัยนี้พบว่าอัลกอริทึม DLRU มีประสิทธิภาพเทคนิคการแทนที่ข้อมูลเว็บดีกว่าอัลกอริทึม LRU โดยที่ค่าเฉลี่ยของอัตราการพบของอัลกอริทึม DLRU เท่ากับ 53.8 เปอร์เซ็นต์ ในขณะที่อัลกอริทึม LRU มีค่าเฉลี่ยของอัตราการพบเท่ากับ 12.99 เปอร์เซ็นต์ ซึ่งสามารถสรุปได้ว่าอัลกอริทึม DLRU มีอัตราการพบเฉลี่ยสูงกว่าอัลกอริทึม LRU ถึง 4 เท่า

อัลกอริทึม DLRU และ LRU ได้มีการพัฒนาโมเดลและทดสอบอัตราการพบโดยการเขียนแอพลิเคชันเฉพาะขึ้นมา ซึ่งแอพลิเคชันดังกล่าวถูกพัฒนาด้วยภาษา C++ ซึ่งแอพลิเคชันนี้จะแสดงจำนวนครั้งที่พบและอัตราการพบให้ทุกๆ 10 นาที

7. กิตติกรรมประกาศ

งานวิจัยนี้สำเร็จลงได้ด้วยดี เนื่องจากได้รับความกรุณาอย่างสูงจาก มหาวิทยาลัยราชภัฏบ้านสมเด็จเจ้าพระยา ที่กรุณาให้ข้อมูลการใช้งานเว็บไซต์ของบุคลากรภายในมหาวิทยาลัยฯ เพื่อนำมาใช้พัฒนาแบบจำลอง และทดสอบอัลกอริทึม DLRU

อนึ่งผู้วิจัยหวังว่างานวิจัยฉบับนี้จะมีประโยชน์อยู่ไม่น้อย จึงขอมอบส่วนดีทั้งหมดนี้ให้แก่อาจารย์ที่ปรึกษาวิทยานิพนธ์ ตลอดจนเหล่าคณาจารย์ที่ได้ประสิทธิประสาทวิชาจนทำให้ผลงานวิจัยเป็นประโยชน์ต่อผู้ที่เกี่ยวข้อง สำหรับข้อบกพร่องต่างๆ ที่อาจจะเกิดขึ้นนั้น ผู้วิจัยขอน้อมรับผิดเพียงผู้เดียว และยินดีที่จะรับฟังคำแนะนำจากทุกท่านที่ได้เข้ามาศึกษาเพื่อเป็นประโยชน์ในการพัฒนางานวิจัยต่อไป

8. เอกสารอ้างอิง

Anthony McGregor, Mark Hall, Perry Lorier and James Brunskill (2004), “Flow clustering using machine learning techniques,” in Proc. of 5th Int. Workshop on Passive and Active Network Measurement, pp.205-214, June 2-7.

A. Waleed, M.S. Siti (2011), “A survey of web caching and prefetching,” in Proc. of ICSRS: International conference on scientific research and studies, vol. 3, no. 1, March.

A. Walled, M.S. Siti, S.I. Abdul (2012), “Intelligent naïve bayes-based approaches for web proxy aching,” Knowledge-based systems, vol. 31, pp. 162-175, July.

Fang, T. Huang, L. Jian, C. Jian-ya, L, Yun-jie (2013), “ Fast convergence caching replacement algorithm based on dynamic classification for content centric networks,” The Journal of China Universities of Posts and Telecommunications, vol. 20, no, 5, pp. 45-50, October.

Gerhard, Z. Günter (2010), Introduction to statistics and data analysis for physicists, Germany

Siegen University, Hamburg, Germany.

M.G. Gharles, S.J. Laurie S.J. (2013), Introduction to probability, 2nd Edition, American Mathematical Society, USA, 1997.[9] W. Akila, Mathematics for economics, Wellesley College, United States.

S.R. William, Z.F. Alain, S.J. Kristynn (2014), "Using generalized additive (mixed) models to analyze single case designs," School Psychology, vol. 52, no. 2, pp. 149-178, April.

Thuy Nguyen and Grenville Armitage (2008), "A survey of techniques for Internet traffic classification using machine learning," IEEE Communications Surveys and Tutorials, vol. 10, no. 4, pp. 56-76, November.

T. M. Cover and J. A. Thomas (2009), Elements of Information Theory, 2nd Edition, Wiley, New York.

W. Akila (2013), Mathematics for economics, Wellesley College, United States.