การลดค่าใช้จ่ายของพื้นที่เก็บข้อมูลบนคลาวด์ด้วยคลาวด์แคชชิ่งแบบท้องถิ่น

นายสุเมธ สิริเรืองอำไพ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2556
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)
เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ ที่ส่งผ่านทางบัณฑิตวิทยาลัย
The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository (CUIR)
are the thesis authors' files submitted through the University Graduate School.

LOWER CLOUD STORAGE COST WITH LOCAL CLOUD CACHING

Mr. Sumet Siriruangampai

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Engineering Program in Computer Engineering

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2013

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การลดค่าใช้จ่ายของพื้นที่เก็บข้อมูลบนคลาวด์ด้วยคลาวด์แคชชิ่งแบบท้องถิ่น
โดย	นายสุเมธ สิริเรืองอำไพ
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ผู้ช่วยศาสตราจารย์ ดร.ณัฐวุฒิ หนูไพโรจน์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

.....คณบดีคณะวิศวกรรมศาสตร์

(ศาสตราจารย์ ดร.บัณฑิต เอื้ออาภรณ์)

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.วีระ เหมืองสิน)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

(ผู้ช่วยศาสตราจารย์ ดร.ณัฐวุฒิ หนูไพโรจน์)

.....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.เกริก ภิรมย์โสภา)

.....กรรมการภายนอกมหาวิทยาลัย

(ผู้ช่วยศาสตราจารย์ ดร.ภุชงค์ อุทโยภาศ)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สุเมธ สิริเรืองอำไพ : การลดค่าใช้จ่ายของพื้นที่เก็บข้อมูลบนคลาวด์ด้วยคลาวด์แคชชิ่งแบบท้องถิ่น. (LOWER CLOUD STORAGE COST WITH LOCAL CLOUD CACHING) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: ผศ. ดร.ณัฐวุฒิ หนูไพโรจน์, 48 หน้า.

พื้นที่เก็บข้อมูลบนคลาวด์ถือเป็นหนึ่งในบริการที่ประสบความสำเร็จมากที่สุดของคลาวด์คอมพิวติงทั้งในแง่ของการใช้งานส่วนบุคคลและในระดับองค์กร โดยหนึ่งในปัจจัยสำคัญที่ส่งผลต่อความสำเร็จของพื้นที่เก็บข้อมูลบนคลาวด์ในระดับองค์กรนั้นได้แก่ เรื่องค่าใช้จ่ายที่ต่ำกว่ามากหากเทียบกับการก่อตั้งหรือขยายขนาดศูนย์ข้อมูล (Data Center) ของตัวเอง อย่างไรก็ตาม การคิดค่าใช้จ่ายของพื้นที่เก็บข้อมูลบนคลาวด์ในระดับองค์กร จำเป็นจะต้องพิจารณาอย่างรอบคอบเพราะค่าบริการที่เกิดขึ้นจากการถ่ายโอนข้อมูลอาจจะทำให้ค่าใช้จ่ายรวมมีมูลค่าสูงกว่าที่คาดการณ์ไว้ ซึ่งหนึ่งในวิธีที่มีการนำมาใช้แก้ไขปัญหานี้ก็คือ การนำแคชเข้ามาช่วยลดปริมาณการถ่ายโอนข้อมูลระหว่างผู้ใช้กับผู้ให้บริการ แต่ในปัจจุบันวิธีการวางทับข้อมูลที่ถูกออกแบบมาสำหรับพื้นที่เก็บข้อมูลบนคลาวด์นั้นยังมีไม่มากนัก

งานวิจัยนี้นำเสนอวิธีการวางทับข้อมูลแบบใหม่ ที่มีจุดประสงค์เพื่อลดค่าใช้จ่ายจากการถ่ายโอนข้อมูลและลดความล่าช้าของระบบ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต .....

สาขาวิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก .....

ปีการศึกษา 2556

5370512121 : MAJOR COMPUTER ENGINEERING
KEYWORDS: CLOUD STORAGE / DATA CACHING / CACHE REPLACEMENT POLICY

SUMET SIRIRUANGAMPAI: LOWER CLOUD STORAGE COST WITH LOCAL CLOUD CACHING. ADVISOR: ASST. PROF. NATAWUT NUPAIROJ, Ph.D., 48 pp.

Cloud storage has become one of the most successful cloud computing services for personal and corporate use. One of the important factors contributing to the success of Cloud Storage at corporation level is much lower expense when compared to building or expanding their own data center. However, careful consideration must be given when it comes to expense calculation at corporation level because data transfer fees may lead to higher expense than expected. One of the solutions that available for this problem is to use cache in order to reduce the traffic from transfer data between users and cloud providers but nowadays there are a few cache replacement policies that all designed for cloud storage.

In this thesis, we present a new cache replacement policy that aims to reduce the cost of data transfer as well as the latency and it has been proved to work effectively.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Department: Computer Engineering Student's Signature .....
Field of Study: Computer Engineering Advisor's Signature .....
Academic Year: 2013

กิตติกรรมประกาศ

ขอกราบขอบพระคุณผู้ช่วยศาสตราจารย์ ดร. ณัฐวุฒิ หนูไพโรจน์ซึ่งเป็นอาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ท่านได้กรุณาสละเวลาดูแลการวิจัยจนสำเร็จลุล่วงไปด้วยดี โดยท่านคอยให้คำแนะนำเกี่ยวกับการทำวิจัย และช่วยชี้แนะการแก้ไขปัญหาต่างๆที่เกิดขึ้นระหว่างการทำวิจัย

ขอกราบขอบพระคุณคณะกรรมสอบวิทยานิพนธ์ทุกท่านเป็นอย่างสูง ได้แก่ ผู้ช่วยศาสตราจารย์ ดร.วีระ เหมืองสิน ผู้ช่วยศาสตราจารย์ ดร.เกริก ภิรมย์โสภา และผู้ช่วยศาสตราจารย์ ดร.ภุชงค์ อุทโยภาศในการตรวจแก้ ให้ข้อคิดและคำแนะนำอันเป็นประโยชน์ยิ่งต่องานวิจัย

ขอขอบพระคุณนายสราวุธ ภู่มะณี และนายเลิศพงษ์ เลิศไพศาลวงศ์ เจ้าหน้าประจำหน่วยงานเทคโนโลยีสารสนเทศทางวิศวกรรม ที่คอยให้คำแนะนำและช่วยเหลือทั้งทางด้านเทคนิคและอุปกรณ์ในการเก็บข้อมูลสำหรับนำมาใช้ในงานวิจัย อีกทั้งยังคอยให้คำปรึกษา และกำลังใจตลอดมา

ขอกราบขอบพระคุณบิดามารดาที่คอยให้กำลังใจ และความห่วงใยจนวิทยานิพนธ์สำเร็จลุล่วงไปด้วยดี

ขอขอบคุณนางสาวชนิกา คำเจริญ ผู้เป็นภรรยาที่คอยแบ่งเบาภาระ และให้กำลังใจตลอดระยะเวลาที่ศึกษาต่อ

ขอขอบคุณนางสาวสุพพัต รุ่งเรืองศิลป์ ผู้เป็นเพื่อนทั้งที่มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ และจุฬาลงกรณ์มหาวิทยาลัยแห่งนี้ ที่คอยช่วยเหลือในด้านต่างๆ ทั้งเรื่องเอกสารและการนำเสนอผลงานจนทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดีทุกประการ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง

บทคัดย่อภาษาอังกฤษ ..... จ

กิตติกรรมประกาศ ..... ฉ

สารบัญ ..... ช

สารบัญตาราง ..... ญ

สารบัญภาพ ..... ฎ

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 วัตถุประสงค์ของการวิจัย 2

1.3 ขอบเขตของการวิจัย 2

1.4 ประโยชน์ที่คาดว่าจะได้รับ 3

1.5 วิธีดำเนินการวิจัย 4

1.6 ผลงานตีพิมพ์ 4

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 5

2.1 คลาวด์คอมพิวติง 5

2.1.1 ประเภทของคลาวด์คอมพิวติง 6

2.1.2 พื้นที่เก็บข้อมูลบนคลาวด์ 7

2.2 แคช 8

2.2.1 หลักการทำงานเบื้องต้นของแคช 9

2.2.2 การจัดเก็บข้อมูลลงแคชแบบแบ่งบล็อก (Block-Based Data Caching) 10

2.2.3 อัลกอริธึมการวางทับข้อมูลในแคช 10

2.2.3.1 การวางทับข้อมูลโดยการพิจารณาความทันสมัย (Recency-based) 11

2.2.3.2 การวางทับข้อมูลโดยการพิจารณาขนาด (Size-based) 12

2.2.3.3 การวางทับข้อมูลโดยการพิจารณาความถี่ (Frequency-based) 13

2.2.3.4 การวางทับข้อมูลโดยการกำหนดฟังก์ชัน (Function-based) 14

2.2.3.5 การวางทับข้อมูลโดยการสุ่ม (Randomized) 16

2.2.4 การเขียนข้อมูลลงสู่แคช (Write Caching) 17

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

หน้า

2.2.4.1 การเขียนทั้งหมด 17
2.2.4.2 การเขียนทีหลัง 17
2.3 งานวิจัยที่เกี่ยวข้อง 18
2.3.1 ผลกระทบจากการถ่ายโอนข้อมูลในพื้นที่เก็บข้อมูลบนคลาวด์ 18
2.3.2 เว็บแคช 18
2.3.3 พฤติกรรมของพื้นที่เก็บข้อมูลบนเครือข่าย (Network Storage) 19
2.3.4 คลาวด์แคช 19
บทที่ 3 การออกแบบอัลกอริธึมการวางทับข้อมูลโดยการเปรียบเทียบมูลค่า 21
3.1 ภาพรวมของแคช 21
3.1.1 โครงสร้างของแคช 21
3.1.2 การทำงานของแคช 21
3.2 การแบ่งข้อมูลออกเป็นบล็อก 22
3.3 การจัดเก็บข้อมูลลงแคช 25
3.4 การเขียนข้อมูลลงแคช 27
3.5 การกำหนดมูลค่าของข้อมูล 29
บทที่ 4 การทดลองและผลการทดลอง 32
4.1 ภาระงาน 32
4.2 วิธีการทดลอง 32
4.3 การประเมินประสิทธิภาพ 33
4.4 แบบจำลองอัตราค่าบริการ 35
4.5 ผลการทดลอง 35
บทที่ 5 สรุปผลการวิจัย 43
5.1 บทสรุป 43
5.2 ข้อเสนอแนะ 43
รายการอ้างอิง 45
ประวัติผู้เขียนวิทยานิพนธ์ 48

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สารบัญตาราง

หน้า

ตารางที่ 2-1 ค่าใช้บริการพื้นที่เก็บข้อมูลบนคลาวด์แบบส่วนบุคคลจาก Dropbox 7
ตารางที่ 2-2 ค่าใช้บริการพื้นที่เก็บข้อมูลบนคลาวด์แบบองค์กรจาก IBM SmartCloud 8
ตารางที่ 2-3 ค่าใช้บริการถ่ายโอนข้อมูลบนคลาวด์แบบองค์กรจาก IBM SmartCloud 8
ตารางที่ 2-4 การแบ่งกลุ่มอัลกอริธึมการวางทับข้อมูล 11
ตารางที่ 4-1 คุณลักษณะของภาระงาน 32
ตารางที่ 4-2 รายละเอียดในการทดลอง 33
ตารางที่ 4-3 ตัวชี้วัดประสิทธิภาพ 34
ตารางที่ 4-4 อัตราค่าบริการของ IBM SmartCloud Enterprise 35

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สารบัญภาพ

หน้า

ภาพที่ 2-1 รูปแบบการทำงานของคลาวด์คอมพิวติง 6
ภาพที่ 2-2 รูปแบบของแคช 9
ภาพที่ 2-3 เปรียบเทียบการจัดเก็บข้อมูลแบบทั้งไฟล์และการจัดเก็บแบบแบ่งบล็อก 10
ภาพที่ 2-4 ตัวอย่างการทำงานของอัลกอริธึมการวางทับข้อมูลแอลอาร์ยู 12
ภาพที่ 2-5 ตัวอย่างการทำงานของอัลกอริธึมการวางทับข้อมูลไซส์ 13
ภาพที่ 2-6 ตัวอย่างการทำงานของอัลกอริธึมการวางทับข้อมูลแอลเอฟยู 14
ภาพที่ 2-7 ตัวอย่างการทำงานของอัลกอริธึมการวางทับข้อมูลจีดีเอสเอฟ 15
ภาพที่ 2-8 ตัวอย่างการทำงานของอัลกอริธึมการวางทับข้อมูลแรนด์ 16
ภาพที่ 2-9 การทำงานของการเขียนข้อมูลแบบเขียนทั้งหมด 17
ภาพที่ 2-10 การทำงานของการเขียนข้อมูลแบบเขียนทีหลัง 18
ภาพที่ 3-1 โครงสร้างของคลาวด์แคช 21
ภาพที่ 3-2 ขั้นการทำงานทั้งหมดของแคช 22
ภาพที่ 3-3 ตัวอย่างการแบ่งข้อมูลออกเป็นบล็อกย่อยๆ 23
ภาพที่ 3-4 ตัวอย่างข้อมูลที่มีขอบเขตของตำแหน่งพอดีบล็อก 23
ภาพที่ 3-5 ตัวอย่างข้อมูลที่มีขอบเขตของตำแหน่งอยู่ในบางส่วนของบล็อก 24
ภาพที่ 3-6 ตัวอย่างข้อมูลที่มีขอบเขตของตำแหน่งอยู่ระหว่างบล็อก 24
ภาพที่ 3-7 คำสั่งเทียมแสดงการทำงานในส่วนของการจัดเก็บข้อมูลลงแคช 26
ภาพที่ 3-8 คำสั่งเทียมแสดงการทำงานในส่วนของการเขียนข้อมูลลงแคช 28
ภาพที่ 3-9 ตัวอย่างปัญหาที่เกิดขึ้นกับอัลกอริธึมการวางทับข้อมูลแอลเอฟยู 30
ภาพที่ 3-10 การแก้ไขปัญหาที่เกิดกับอัลกอริธึมการวางทับข้อมูลแอลเอฟยูด้วยไดนามิก เอจจิง 31
ภาพที่ 4-1 ปริมาณการถ่ายโอนข้อมูล 35
ภาพที่ 4-2 ปริมาณแบนด์วิดธ์ที่ลดลง 36
ภาพที่ 4-3 Hit Rate 37
ภาพที่ 4-4 Byte Hit Rate 37

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ภาพที่ 4-5 ค่าใช้จ่าย 38
ภาพที่ 4-6 ค่าความหน่วงโดยเฉลี่ย 39
ภาพที่ 4-7 ผลกระทบที่เกิดจากอัตราส่วนค่าใช้จ่ายของการถ่ายโอนข้อมูล 40
ภาพที่ 4-8 ค่าบริการเมื่อใช้งานผู้ให้บริการคลาวด์หลายแห่ง 41

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

คลาวด์คอมพิวติง (Cloud Computing) ถือเป็นแนวคิดที่รอการต่อยอดให้เกิดขึ้นจริงในแง่ของการบริการสาธารณะทางด้านการประมวลผล (Computing as a Utility) มาอย่างยาวนาน ซึ่งคลาวด์คอมพิวติงก่อให้เกิดการเปลี่ยนแปลงต่างๆในโลกของคอมพิวเตอร์ไปอย่างสิ้นเชิงทั้งทางด้านฮาร์ดแวร์และซอฟต์แวร์ [1] โดยข้อดีหลักๆของคลาวคอมพิวติงนั้นประกอบไปด้วย

- การใช้เงินที่น้อยกว่าสำหรับธุรกิจขนาดเล็กเพื่อให้ได้ประสิทธิในการประมวลผลหรือทรัพยากรต่างๆที่มากเทียบเท่ากับธุรกิจขนาดใหญ่
- ไม่ต้องบริหารจัดการระบบและทรัพยากรต่างๆเอง ทำให้สามารถลดค่าใช้จ่ายในเรื่องของ
- ค่าบำรุงรักษาและค่าจ้างพนักงานทางด้านไอทีลงไป
- ไม่จำเป็นต้องใช้เงินลงทุนในส่วนของทรัพยากรทางด้านการประมวลผลหรือระบบศูนย์ข้อมูล (Data Center) เพราะสามารถเริ่มต้นงานได้ทันทีที่เข้าใช้งานคลาวด์
- ขยายหรือลดขนาดของทรัพยากรต่างๆได้อย่างอิสระ

ด้วยข้อดีต่างๆหล่านี้ [1, 2] จึงทำให้สามารถใช้ทรัพยากรต่างๆในระบบได้อย่างคุ้มค่าและเต็มประสิทธิภาพมากขึ้นทั้งในแง่ของผู้ให้บริการและผู้ใช้บริการ ซึ่งถือว่าเป็นสิ่งที่สำคัญอย่างยิ่งสำหรับโลกในยุคปัจจุบันที่มีปัญหาทางด้านทรัพยากรและสิ่งแวดล้อมต่างๆมากมาย รวมไปถึงปัญหาทางด้านเศรษฐกิจอีกด้วย ดังนั้นระบบคลาวคอมพิวติงที่มีความยืดหยุ่นทั้งทางด้านฮาร์ดแวร์และแพลตฟอร์มมากกว่าระบบกริดคอมพิวติง (Grid Computing) [3] รวมไปถึงแนวคิดในเรื่องของค่าใช้จ่ายที่เป็นแบบสาธารณูปโภค (Utility) จึงน่าจะเป็นสิ่งที่สามารถตอบโจทย์ได้ตรงกับความต้องการของทุกคนมากที่สุดทั้งในตอนนี้และในอนาคต

สำหรับพื้นที่เก็บข้อมูลบนคลาวด์ (Cloud Storage) นั้นก็ถือเป็นหนึ่งในบริการของคลาวด์คอมพิวติ้งที่ได้รับความนิยมในปัจจุบัน ทั้งในรูปแบบการใช้งานส่วนตัว [4-6] การใช้งานทางด้านธุรกิจ [7-9] และการใช้งานทางด้านการศึกษา [10, 11] เพราะมีค่าใช้จ่ายที่ค่อนข้างต่ำกว่า เมื่อเปรียบเทียบกับการสร้างหรือขยายขนาดศูนย์ข้อมูลของตนเอง อีกทั้งยังสามารถเข้าถึงข้อมูลได้อย่างอิสระทุกที่ทุกเวลา แต่ในความเป็นจริงแล้วเมื่อพิจารณาค่าใช้จ่ายของทั้ง 2 ระบบจะพบว่า ค่าใช้จ่ายของพื้นที่เก็บข้อมูลบนคลาวด์ ไม่ได้มีเพียงแค่ค่าเช่าพื้นที่เท่านั้น แต่ยังมีค่าใช้จ่ายในเรื่องของการถ่ายโอนข้อมูล (Data Transfer) อีกด้วย ซึ่งจำเป็นจะต้องคิดรวมเข้าไปจึงจะถือได้ว่าเป็นค่าใช้จ่ายที่แท้จริงของพื้นที่เก็บข้อมูลบนคลาวด์ ซึ่งนั่นอาจจะทำให้ระบบที่มีการถ่ายโอนข้อมูลเป็นจำนวนมาก มีค่าใช้จ่ายที่สูงกว่าการใช้งานศูนย์ข้อมูลของตนเอง

การจัดเก็บข้อมูลลงแคช (Data Caching) ถือเป็นหนึ่งในวิธีที่สามารถนำมาใช้แก้ปัญหานี้อย่างได้ผล โดยนอกจากจะสามารถแก้ปัญหาเรื่องของค่าใช้จ่ายได้แล้ว ยังสามารถลดค่าความหน่วง (Latency) ของระบบได้อีกด้วย ซึ่งถือเป็นข้อเสียเปรียบอีกอย่างหนึ่งของพื้นที่เก็บข้อมูลบนคลาวด์ที่ทำงานผ่านระบบเครือข่ายอินเตอร์เน็ตที่มีความเร็วต่ำกว่าศูนย์ข้อมูลภายในองค์กรที่ทำงานผ่านระบบเครือข่ายท้องถิ่น (LAN) อย่างไรก็ตาม การจะทำให้แคชของเราสามารถทำงานได้ตรงตามวัตถุประสงค์ที่วางไว้มากที่สุด จำเป็นจะต้องมีการออกแบบแคชนั้นๆให้มีความเหมาะสมกับระบบที่จะนำไปใช้งาน ซึ่งหนึ่งในส่วนประกอบที่สำคัญของแคชที่ควรได้รับการพิจารณาอย่างรอบคอบก็คืออัลกอริธึมการวางทับข้อมูลในแคช (Cache Replacement Algorithm) ที่ทำหน้าที่ในการตัดสินใจเลือกข้อมูลบางส่วนออกจากแคชเมื่อพื้นที่ของแคชเต็ม เพื่อสร้างพื้นที่ว่างให้เพียงพอกับข้อมูลใหม่ ซึ่งเมื่อพิจารณาจากงานวิจัยที่เกี่ยวข้องกับการจัดเก็บข้อมูลลงแคชของพื้นที่เก็บข้อมูลบนคลาวด์แล้วจะพบว่าอัลกอริธึมการวางทับข้อมูลในแคชที่ออกแบบมาสำหรับระบบนี้ยังมีไม่ค่อยมาก และอัลกอริธึมการวางทับข้อมูลอย่างแอลอาร์ยูที่ใช้กันอย่างแพร่หลายในระบบต่างๆ ก็ทำงานได้ไม่ดีนักภายใต้สภาวะแวดล้อมเช่นนี้

งานวิจัยชิ้นนี้จึงได้นำเสนอแนวทางในการทำจัดเก็บข้อมูลลงแคชเพื่อลดค่าใช้จ่ายจากการใช้งานพื้นที่เก็บข้อมูลบนคลาวด์ โดยการใช้พื้นที่เก็บข้อมูลขนาดเล็กมาเป็นแคชของข้อมูลเพื่อลดปริมาณการถ่ายโอนข้อมูล รวมไปถึงการออกแบบอัลกอริธึมการวางทับข้อมูลที่เน้นไปในเรื่องของการลดค่าใช้จ่ายที่เกิดจากการถ่ายโอนข้อมูลนั้นๆให้ได้มากที่สุด ซึ่งถ้าหากเราสามารถลดค่าใช้จ่ายที่เกิดจากการใช้งานระบบคลาวด์ให้มีมูลค่าที่ต่ำกว่าระบบอื่นๆได้ ก็น่าจะทำให้การใช้งานคลาวด์คอมพิวติ้งเป็นไปอย่างแพร่หลายมากยิ่งขึ้นโดยเฉพาะในประเทศไทย

1.2 วัตถุประสงค์ของการวิจัย

เพื่อศึกษา ออกแบบและนำเสนออัลกอริธึมการวางทับข้อมูลที่เหมาะสมกับพื้นที่เก็บข้อมูลบนคลาวด์ โดยมีจุดประสงค์หลักเพื่อลดค่าใช้จ่ายที่เกิดขึ้นจากการถ่ายโอนข้อมูลระหว่างผู้ใช้และผู้ให้บริการพื้นที่เก็บข้อมูลบนคลาวด์ และมีประสิทธิภาพที่ใกล้เคียงหรือดีกว่าอัลกอริธึมการวางทับข้อมูลแบบอื่นๆ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

1.3 ขอบเขตของการวิจัย

1. ข้อมูลการใช้งานพื้นที่เก็บข้อมูลบนเครือข่ายที่นำมาทดสอบ คือข้อมูลการใช้งาน NFS Server ของคณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เป็นระยะเวลาอย่างน้อย 7 วัน
2. วิธีการกำหนดปลายทางที่เป็นพื้นที่เก็บข้อมูลในกรณีใช้งานพื้นที่เก็บข้อมูลบนคลาวด์หลายแห่ง คือ แบบสลับตัวกันไป (Round Robin)
3. จำลองพื้นที่เก็บข้อมูลในแคชตั้งแต่ 10%, 20%, 30%, 40% และ 50% ของขนาดเนื้อที่ในแคชที่น้อยที่สุดที่สามารถเก็บข้อมูลทั้งหมดได้โดยไม่มีการแทนที่เกิดขึ้น

4. อัลกอริธึมการวางทับข้อมูลที่นำมาเปรียบเทียบประสิทธิภาพ ได้แก่ แอลอาร์ยู (Least Recently Used (LRU)) แอลเอฟยู (Least Frequency Used (LFU)) และแอลเอฟยูดีเอ (Least Frequency Used with Dynamic Aging (LFUDA))
5. เกณฑ์การวัดประสิทธิภาพของอัลกอริธึม ได้แก่
    5.1 ปริมาณการถ่ายโอนข้อมูล (Data Transfer)
    5.2 ค่าฮิตเรต (Hit Rate)
    5.3 ค่าไบต์ฮิตเรต (Byte Hit Rate)
    5.4 การประหยัดแบนด์วิธ (Bandwidth Saving)
    5.5 ค่าใช้จ่าย
    5.6 ค่าความหน่วง (Latency)
    5.7 ผลกระทบที่เกิดจากอัตราส่วนค่าใช้จ่ายของการถ่ายโอนข้อมูล (Effect of Transfer Cost Ratio)
    5.8 ค่าใช้จ่ายเมื่อมีการใช้งานพื้นที่เก็บข้อมูลบนคลาวด์หลายแห่ง
6. งานวิจัยนี้เป็นการออกแบบอัลกอริธึมการวางทับข้อมูลสำหรับแคชแบบแบ่งบล็อก (Block-based Cache) ซึ่งทำหน้าที่เป็นพร็อกซี (Proxy) ระหว่างผู้ให้บริการพื้นที่เก็บข้อมูลบนแคชกับเครื่องคอมพิวเตอร์ลูกข่าย
7. กำหนดโครงสร้างของระบบเครือข่ายและสภาพแวดล้อมในรูปแบบที่ประกอบไปด้วย ลูกข่าย แม่ข่ายแคช และพื้นที่เก็บข้อมูลบนคลาวด์ โดยลูกข่ายเชื่อมต่อกับแม่ข่ายพร็อกซีผ่านระบบเครือข่ายท้องถิ่น และแม่ข่ายพร็อกซีเชื่อมต่อกับพื้นที่เก็บข้อมูลบนคลาวด์ผ่านระบบเครือข่ายอินเตอร์เน็ท
8. งานวิจัยนี้ทำการทดลองด้วยโปรแกรมจำลองการทำงานของระบบเครือข่าย
9. กำหนดให้การเขียนข้อมูลกลับไปยังพื้นที่เก็บข้อมูลบนคลาวด์เป็นแบบ เขียนกลับ (Write Back)
10. กำหนดให้แม่ข่ายพร็อกซีตอบสนองต่อการร้องขอของเครื่องคอมพิวเตอร์ลูกข่ายครั้งละ 1 เครื่อง ณ ช่วงเวลาหนึ่งๆ

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. ทำให้สามารถออกแบบและพัฒนาอัลกอริธึมการวางทับข้อมูลที่เหมาะสำหรับพื้นที่เก็บข้อมูลบนคลาวด์ได้ ทั้งในแง่ของค่าใช้จ่ายและประสิทธิภาพ
2. เป็นแนวทางในการลดค่าใช้จ่ายจากการใช้งานพื้นที่เก็บข้อมูลบนคลาวด์
3. ทำให้พื้นที่เก็บข้อมูลบนคลาวด์ตัวเลือกที่น่าสนใจมากขึ้น

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

1.5 วิธีดำเนินการวิจัย

1. ศึกษาทฤษฎี หลักการพื้นฐานที่เกี่ยวข้องกับแคชและอัลกอริธึมการวางทับข้อมูล
2. ศึกษาและประเมินประสิทธิภาพของอัลกอริธึมการวางทับข้อมูลแบบต่างๆถึงแนวคิด หลักการ ข้อดี และข้อบกพร่องของแต่ละชนิด
3. ออกแบบและนำเสนออัลกอริธึมการวางทับข้อมูลที่เหมาะสำหรับนำไปใช้งานกับพื้นที่เก็บข้อมูลบนคลาวด์
4. ทดสอบวิธีการที่นำเสนอ
5. วิเคราะห์ผลการทดลอง
6. สรุปผลและเรียบเรียงวิทยานิพนธ์

1.6 ผลงานตีพิมพ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้ตีพิมพ์และนำเสนอในการประชุมวิชาการดังนี้

1.6.1 บทความชื่อ “Lower Storage TCO with Local Cloud Caching” [12]

1.6.1.1 ชื่อผู้แต่ง Sumet Siriruangampai และ Natawut Nupairoj

1.6.1.2 ตีพิมพ์และนำเสนอในงานประชุมวิชาการชื่อ The 2011 Thai Grid and Cloud Conference (TGCC ‘11) ซึ่งจัดขึ้นในวันที่ 20 มิถุนายน 2554 จ.กรุงเทพมหานคร ประเทศไทย

1.6.2 บทความชื่อ “Cost-Based Replacement Policy for Cloud Storage” [13]

1.6.2.1 ชื่อผู้แต่ง Sumet Siriruangampai และ Natawut Nupairoj

1.6.2.2 ตีพิมพ์และนำเสนอในงานประชุมวิชาการชื่อ The 10th National Conference on Computing and Information Technology (NCCIT2014) ซึ่งจัดขึ้นในวันที่ 8-9 พฤษภาคม 2557 จ.ภูเก็ต ประเทศไทย

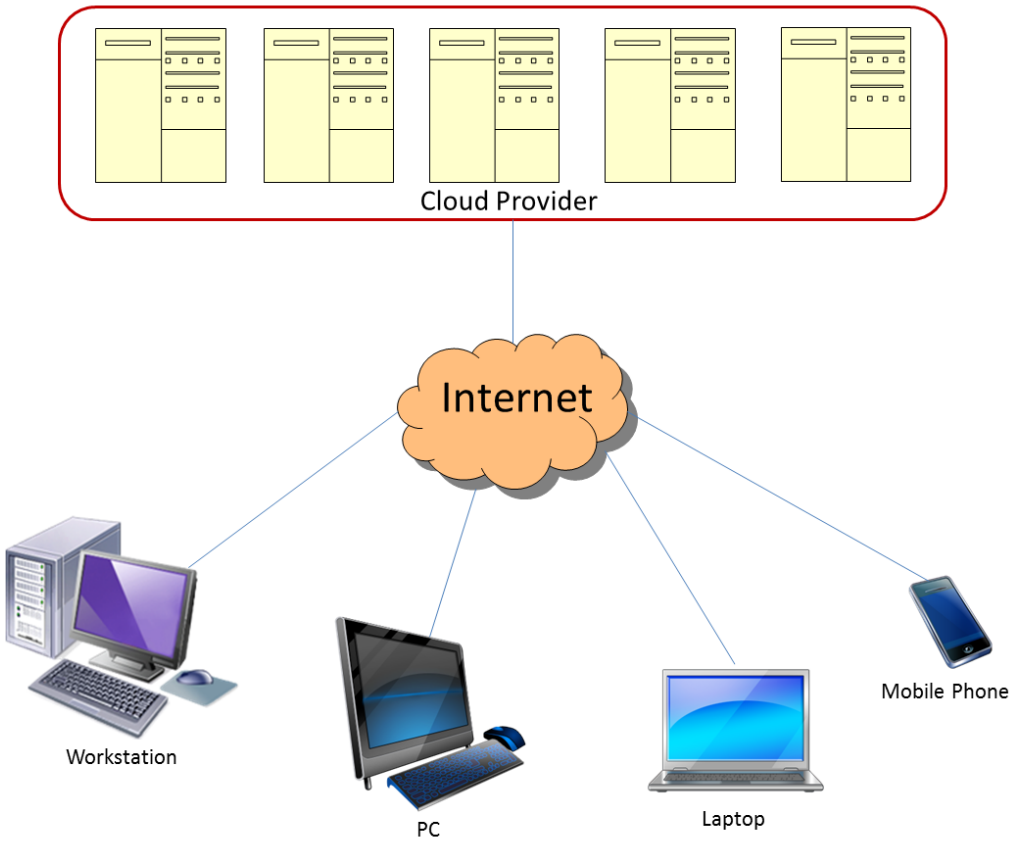
CHULALONGKORN UNIVERSITY

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในส่วนนี้เป็นแนวคิดและทฤษฎีที่เกี่ยวข้องกับคลาวด์คอมพิวติง และแคช ซึ่งจะกล่าวถึงลักษณะทั่วไปของคลาวด์คอมพิวติง การแบ่งประเภทของคลาวด์คอมพิวติง พื้นที่เก็บข้อมูลบนคลาวด์ ลักษณะทั่วไปของแคช หลักการทำงานของแคช และอัลกอริธึมการวางทับข้อมูลในแคช ตามลำดับ

2.1 คลาวด์คอมพิวติง

คลาวด์คอมพิวติงเป็นแนวคิดที่ได้รับการพัฒนาต่อจากกริดคอมพิวติงให้มีความยืดหยุ่นและง่ายต่อการเข้าถึงมากขึ้น [3] เพียงแค่ผู้ใช้มีบัตรเครดิต ก็สามารถเข้าถึงบริการต่างๆได้ทันที แตกต่างจากกริดคอมพิวติง ที่ผู้ใช้จำเป็นจะต้องมีทรัพยากรของตัวเอง ในการนำไปรวมกับทรัพยากรของคนอื่นๆเพื่อสร้างกลุ่มผู้ใช้งานกริดคอมพิวติงขึ้นมา ทำให้ต้องใช้เงินลงทุนในช่วงแรกที่สูงกว่า นอกจากนี้ในการใช้งานแต่ละครั้ง ผู้ใช้จำเป็นจะต้องร้องขอไปยังผู้จัดการทรัพยากรให้จัดสรรช่วงเวลาใช้งานให้ และต้องรอจนกว่าจะถึงกำหนด ทำให้การใช้งานกริดคอมพิวติงเกิดขึ้นแค่ในวงแคบๆ เช่น สถาบันการศึกษา และหน่วยงานต่างๆของภาครัฐ เป็นต้น แต่ยังไม่เป็นที่แพร่หลายในวงการธุรกิจหรือผู้ใช้ทั่วไป ในปัจจุบันการใช้งานคลาวด์คอมพิวติงได้เริ่มเป็นที่แพร่หลายมากขึ้นทั้งในระดับบุคคลและองค์กรต่างๆ อีกทั้งยังมีบริการที่หลากหลาย ไม่ได้จำกัดอยู่แค่การใช้งานเพื่อเน้นการประมวลผลประสิทธิภาพสูงเหมือนกริดคอมพิวติงเพียงอย่างเดียว นอกจากนี้ยังไม่ถูกจำกัดอยู่แค่การใช้งานบนคอมพิวเตอร์อีกด้วย เช่น บริการจาก Dropbox ที่มีลักษณะเป็นพื้นที่เก็บข้อมูลส่วนบุคคล (Personal Storage) ที่ผู้ใช้สามารถฝากไฟล์ต่างๆไว้กับพื้นที่เก็บข้อมูลบนคลาวด์ผ่านทางอุปกรณ์ต่างได้เช่น โทรศัพท์เคลื่อนที่ คอมพิวเตอร์พกพา เป็นต้น นอกจากนี้โปรแกรม Dropbox เองยังมีคุณสมบัติของการซิงค์ข้อมูล ทำให้ข้อมูลที่เราอัพโหลดขึ้นไปบนคลาวด์ ไปปรากฏอยู่ในอุปกรณ์ทุกชนิดที่เราลงทะเบียนไว้กับบัญชีผู้ใช้ของเราอีกด้วย ซึ่งสร้างความสะดวกสบายให้กับผู้ใช้เป็นอ่างมาก สำหรับรูปแบบการทำงานของคลาวด์คอมพิวติงนั้นมีลักษณะดังภาพที่ 2-1 คือผู้ใช้จำเป็นจะต้องเชื่อมต่ออุปกรณ์ของตัวเองเข้ากับอินเตอร์เน็ตเพื่อใช้งานบริการทางด้านคลาวด์คอมพิวติงจากผู้ให้บริการคลาวด์ (Cloud Provider)



ภาพที่ 2-1 รูปแบบการทำงานของคลาวด์คอมพิวติง

2.1.1 ประเภทของคลาวด์คอมพิวติง

ปัจจุบันประเภทของคลาวด์คอมพิวติงได้มีการแยกย่อยออกไปจากในช่วงเริ่มแรกเป็นจำนวนมาก เช่น Data-as-a-Service (DaaS), Business Process-as-a-Service (BPaaS) เป็นต้น แต่หากยึดตามมาตรฐานที่กำหนดโดยสถาบันมาตรฐานและเทคโนโลยีแห่งชาติ (National Institute of Standards and Technology) ซึ่งเป็นหน่วยงานในสังกัดกระทรวงพาณิชย์แห่งสหรัฐอเมริกา คลาวด์คอมพิวติ้งจะถูกแบ่งออกเป็น 3 ประเภทหลักดังนี้ [14]

- Infrastructure-as-a-Service (IaaS) คือการให้บริการทางด้านฮาร์ดแวร์ (Hardware) ทั้งหน่วยประมวลผล หน่วยความจำ พื้นที่เก็บข้อมูล เครือข่าย และทรัพยากรต่างๆ ในลักษณะของเวอร์ชวลแมชชีน (Virtual Machine) เพื่อรองรับการทำซอฟต์แวร์ (Software) หรือแอพพลิเคชั่น (Application)
- Platform-as-a-Service (PaaS) คือการให้บริการทางด้านแพลตฟอร์มเพื่อใช้ในการรองรับการทำงานของซอฟต์แวร์หรือแอพพลิเคชั่น ซึ่งจะประกอบไปด้วย

ระบบปฏิบัติการและฐานข้อมูล รวมถึงมิคเดิลแวร์ (Middleware) ต่างๆอีกด้วย โดยผู้ใช้สามารถดูแลและควบคุมได้อย่างอิสระในลักษณะของเวอร์ชวลแมชชีนเช่นเดียวกับ IaaS

- Software-as-a-Service (SaaS) คือการให้บริการทางด้านซอฟต์แวร์ที่ผู้ใช้สามารถเรียกใช้งานผ่านอินเตอร์เน็ตได้ทันที โดยไม่จำเป็นต้องติดตั้งโปรแกรมนั้นๆ ซึ่งจะมีการคิดค่าบริการในรูปแบบของไลเซนส์ ระยะเวลา หรือตามปริมาณการใช้งานที่เกิดขึ้น

2.1.2 พื้นที่เก็บข้อมูลบนคลาวด์

พื้นที่เก็บข้อมูลบนคลาวด์เป็นบริการที่จัดอยู่ในประเภท IaaS ที่เราสามารถนำมาประยุกต์รวมเข้ากับระบบที่มีอยู่แล้วได้ทันที โดยไม่ได้ต้องเสียเวลาติดตั้งหรือจัดเตรียมอุปกรณ์ต่างๆก่อน อีกทั้งยังรองรับมาตรฐานต่างๆได้อย่างครบครัน ไม่ต่างจากการขยายหรือก่อตั้งศูนย์ข้อมูลของตัวเอง ทำให้หลายๆ องค์กรต่างหันมาสนใจพื้นที่เก็บข้อมูลบนคลาวด์กันมากขึ้น เพราะความสะดวกและมีค่าใช้จ่ายในส่วนของเงินลงทุนที่น้อยกว่า เราสามารถแบ่งลักษณะของพื้นที่เก็บข้อมูลบนคลาวด์แบบคร่าวๆได้ 2 ประเภท คือแบบการใช้งานส่วนบุคคล และแบบการใช้งานในระดับองค์กร ซึ่งในส่วนของการใช้งานส่วนบุคคลนั้น ส่วนใหญ่จะมีการเรียกเก็บค่าใช้บริการตามขนาดพื้นที่ที่เช่าไว้ แต่การใช้งานในระดับองค์กร นอกจากจะมีการเรียกเก็บค่าใช้บริการจากจำนวนพื้นที่ที่ถูกใช้งานไปแล้ว ยังมีค่าใช้จ่ายในเรื่องของการถ่ายโอนข้อมูลเพิ่มเข้ามาอีกด้วย ซึ่งถือเป็นอีกหนึ่งปัจจัยสำคัญที่ส่งผลให้ค่าใช้จ่ายที่เกิดขึ้นทั้งหมดมีปริมาณมากหรือน้อย ทั้งนี้ตัวอย่างค่าบริการในรูปแบบต่างๆสรุปได้ตามตารางที่ 2-1 [5] 2-2 [15] และ 2-3 [15]

ตารางที่ 2-1 ค่าใช้บริการพื้นที่เก็บข้อมูลบนคลาวด์แบบส่วนบุคคลจาก Dropbox

ขนาดพื้นที่	ค่าบริการแบบรายเดือน	ค่าบริการแบบรายปี
2 GB	ฟรี	ฟรี
100 GB	9.99 เหรียญ	99 เหรียญ
200 GB	19.99 เหรียญ	199 เหรียญ
500 GB	49.99 เหรียญ	499 เหรียญ

ตารางที่ 2-2 ค่าใช้บริการพื้นที่เก็บข้อมูลบนคลาวด์แบบองค์กรจาก IBM SmartCloud

จำนวนพื้นที่ที่ใช้เฉลี่ยต่อเดือน	อัตราค่าบริการต่อ GB
1 TB แรก	0.165 เหรียญ
99 TB ถัดไป	0.15 เหรียญ
400 TB ถัดไป	0.13 เหรียญ
500 TB ถัดไป	0.12 เหรียญ
4000 TB ถัดไป	0.1 เหรียญ
5000 TB ถัดไป	0.08 เหรียญ

ตารางที่ 2-3 ค่าใช้บริการถ่ายโอนข้อมูลบนคลาวด์แบบองค์กรจาก IBM SmartCloud

ปริมาณการถ่ายโอนข้อมูล	อัตราค่าบริการต่อ GB
10 TB แรก	0.15 เหรียญ
40 TB ถัดไป (ตั้งแต่ 10 TB จนถึง 50 TB)	0.11 เหรียญ
100 TB ถัดไป (ตั้งแต่ 50 TB จนถึง 150 TB)	0.09 เหรียญ
ส่วนเกินตั้งแต่ 150 TB ขึ้นไป	0.08 เหรียญ

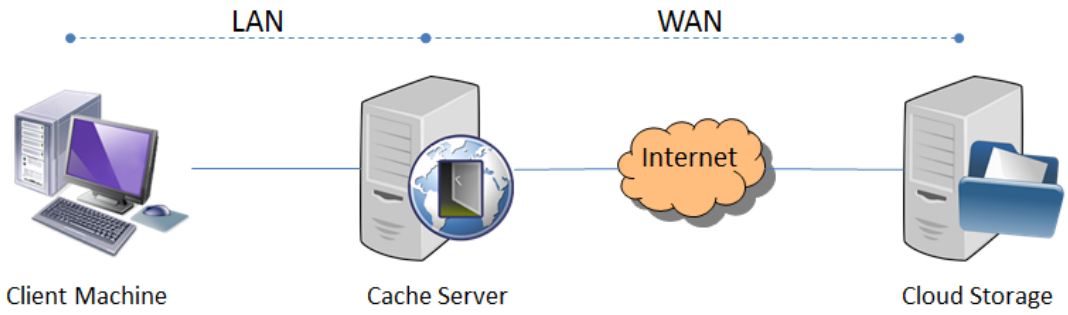
จากตารางที่ 2-2 และ 2-3 นั้นจะเห็นได้ว่า หากพื้นที่เก็บข้อมูลบนคลาวด์ถูกนำไปใช้งานกับองค์กรที่มีการถ่ายโอนข้อมูลเป็นปริมาณมากๆ อาจจะทำให้ค่าใช้จ่ายที่เกิดขึ้นจากการถ่ายโอนข้อมูลมีมูลค่าสูงกว่าค่าใช้จ่ายที่เกิดจากการเช่าพื้นที่อย่างมีนัยสำคัญ จนทำให้การเลือกใช้งานพื้นที่เก็บข้อมูลบนคลาวด์ สิ้นเปลืองกว่าการก่อตั้งหรือขยายขนาดศูนย์ข้อมูลของตนเอง

2.2 แคช

แคชถือเป็นเทคนิคที่ได้รับความนิยมอย่างแพร่หลาย และถูกนำไปใช้แก้ไขปัญหารวมถึงปรับปรุงประสิทธิภาพให้แก่ระบบต่างๆ ตั้งแต่ซีพียู ไปจนถึงดาตากริด [16] ซึ่งหน้าที่หลักๆของแคชได้แก่การทำสำเนาข้อมูลที่มีแนวโน้มจะถูกเรียกใช้ซ้ำในอนาคตเอาไว้ในพื้นที่เก็บข้อมูลของแคชที่มีความเร็วสูงกว่าพื้นที่ที่เก็บข้อมูลต้นฉบับ ซึ่งจุดประสงค์ในการนำแคชมาใช้งานนั้นก็จะแตกต่างกันไปในแต่ละระบบ แต่โดยหลักๆแล้วจะประกอบไปด้วย [17]

- เพื่อลดเวลาในการเข้าถึงข้อมูลของผู้ใช้
- เพื่อลดภาระให้กับเครื่องแม่ข่ายที่เก็บข้อมูลต้นฉบับเอาไว้
- เพื่อลดปริมาณแบนด์วิดธ์ที่เกิดขึ้น

จากข้อดีดังกล่าวจะพบว่าหากเรานำแคชเข้ามาประยุกต์ใช้งานกับพื้นที่เก็บข้อมูลบนคลาวด์ จะทำให้เกิดประโยชน์ทั้งในแง่ของประสิทธิภาพและค่าใช้ง่าย โดยผู้ใช้สามารถเรียกใช้ไฟล์ที่เก็บอยู่บนคลาวด์ได้ในระยะเวลาเฉลี่ยที่รวดเร็วขึ้น อีกทั้งยังสามารถลดค่าใช้จ่ายในเรื่องของการถ่ายโอนข้อมูลลง เนื่องจากไม่จำเป็นต้องโหลดข้อมูลที่เก็บอยู่แคชซ้ำจากคลาวด์ สำหรับรูปแบบของแคชที่จะนำมาใช้กับพื้นที่เก็บข้อมูลบนคลาวด์นั้นจะมีลักษณะที่คล้ายกับเว็บแคช โดยแคชที่จะนำมาใช้กับพื้นที่เก็บข้อมูลบนคลาวด์นั้นจะทำหน้าที่เป็นพร็อกซีคั่นระหว่างเครื่องลูกข่ายกับพื้นที่เก็บข้อมูลบนคลาวด์ดังภาพที่ 2-2



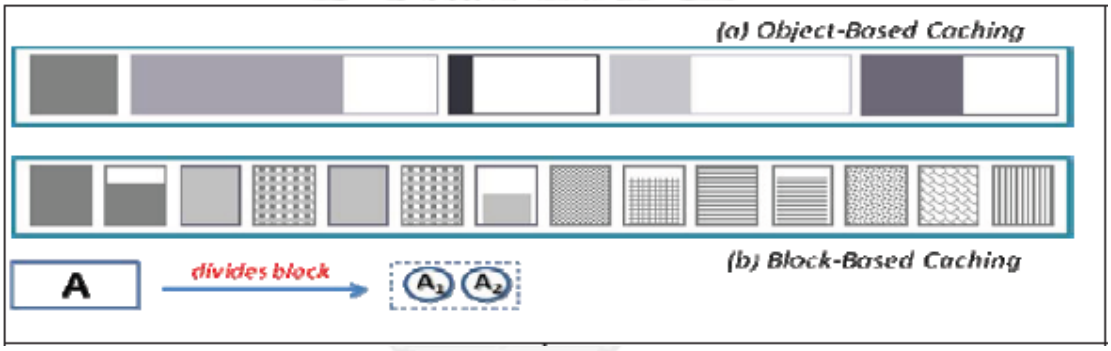
ภาพที่ 2-2 รูปแบบของแคช

2.2.1 หลักการทำงานเบื้องต้นของแคช

หลักการทำงานเบื้องต้นของแคชคือ หากผู้ใช้งานมีการเรียกหาข้อมูล ระบบจะทำการค้นหาข้อมูลนั้นๆในแคชก่อนเป็นอันดับแรก หากพบก็จะทำการส่งข้อมูลกลับไปยังผู้ใช้ทันที แต่หากไม่พบระบบก็จะทำการถ่ายโอนข้อมูลจากพื้นที่เก็บข้อมูลบนคลาวด์แล้วส่งไปให้ผู้ใช้ จากนั้นจึงทำการสำเนาข้อมูลเก็บไว้ในแคช ซึ่งหากสังเกตจากภาพที่ 2-2 แล้วจะพบว่าผู้ใช้เชื่อมต่อกับแคชผ่านระบบเครือข่ายท้องถิ่น ทำให้การถ่ายโอนข้อมูลในส่วนนี้จะมีความเร็วที่สูงกว่าการเชื่อมต่อระหว่างแคชกับพื้นที่เก็บข้อมูลบนคลาวด์ที่กระทำผ่านเครือข่ายบริเวณกว้าง (WAN) ดังนั้นหากเราสามารถทำให้ข้อมูลส่วนใหญ่ที่ผู้ใช้ต้องการ ถูกเก็บอยู่ในแคช ก็จะทำให้สามารถลดปริมาณการถ่ายโอนข้อมูลจากคลาวด์ลงได้ นอกจากนี้ยังทำให้ผู้ใช้ได้รับข้อมูลที่ต้องการในระยะเวลาอันสั้นอีกด้วย

2.2.2 การจัดเก็บข้อมูลลงแคชแบบแบ่งบล็อก (Block-Based Data Caching)

การจัดเก็บข้อมูลลงแคชแบบแบ่งบล็อก [16] มีหลักการทำงานที่สำคัญคือ ข้อมูลต่างๆจะถูกแบ่งออกเป็นหน่วยย่อยๆขนาดคงที่ที่เรียกว่าบล็อก แตกต่างจากเว็บแคชโดยทั่วไปที่มักจะเก็บข้อมูลแบบทั้งไฟล์ (File-Based) ซึ่งไม่เหมาะกับไฟล์ที่มีขนาดใหญ่เพราะจะทำให้เกิดความล่าช้าและมีค่าใช้จ่ายที่เกิดขึ้นจากการถ่ายโอนข้อมูลเป็นจำนวนมาก ทั้งนี้ในหลายๆกรณีผู้ใช้มีความจำเป็นเพียงแค่การอ่านหรือเขียนข้อมูลลงในบางส่วนของไฟล์เท่านั้น ทำให้นอกจากจะเสียค่าใช้จ่ายในส่วนที่ไม่จำเป็นต้องใช้งานแล้ว ยังส่งผลให้เสียพื้นที่ในแคชไปกับข้อมูลที่ไม่มีประโยชน์อีกด้วย ดังนั้นการจัดเก็บข้อมูลลงแคชแบบแบ่งบล็อกจะช่วยให้สามารถใช้งานแคชได้อย่างเต็มประสิทธิภาพและประหยัดค่าใช้จ่ายได้มากขึ้น



ภาพที่ 2-3 เปรียบเทียบการจัดเก็บข้อมูลแบบทั้งไฟล์และการจัดเก็บแบบแบ่งบล็อก [16]

2.2.3 อัลกอริธึมการวางทับข้อมูลในแคช

อัลกอริธึมการวางทับข้อมูลถือเป็นอีกหนึ่งกลไกสำคัญภายในแคช ที่สามารถเป็นตัว กำหนดได้ว่าแคชนั้นๆจะมีประสิทธิภาพมากน้อยแค่ไหน โดยหน้าที่สำคัญของอัลกอริธึมการวางทับข้อมูลจะอยู่ในช่วงที่แคชมีการเก็บข้อมูลจนเต็มความจุ ทำให้ไม่สามารถเพิ่มข้อมูลใหม่ๆลงไปได้ ระบบจึงจำเป็นจะต้องนำข้อมูลเก่าออกไปเพื่อสร้างพื้นที่สำหรับข้อมูลใหม่ ซึ่งกระบวนการในการคัดเลือกข้อมูลที่จะถูกวางทับนี้เป็นหน้าที่ของอัลกอริธึมการวางทับข้อมูล ปัจจุบันมีการนำเสนออัลกอริธึมการวางทับข้อมูลออกมาเป็นจำนวนมาก ซึ่งเราสามารถทำการจัดกลุ่มของอัลกอริธึมต่างๆที่มีคุณสมบัติคล้ายกันได้ตามตารางที่ 2-4 โดยจะแบ่งออกเป็น 5 กลุ่มดังนี้

ตารางที่ 2-4 การแบ่งกลุ่มอัลกอริธึมการวางทับข้อมูล [17, 18]

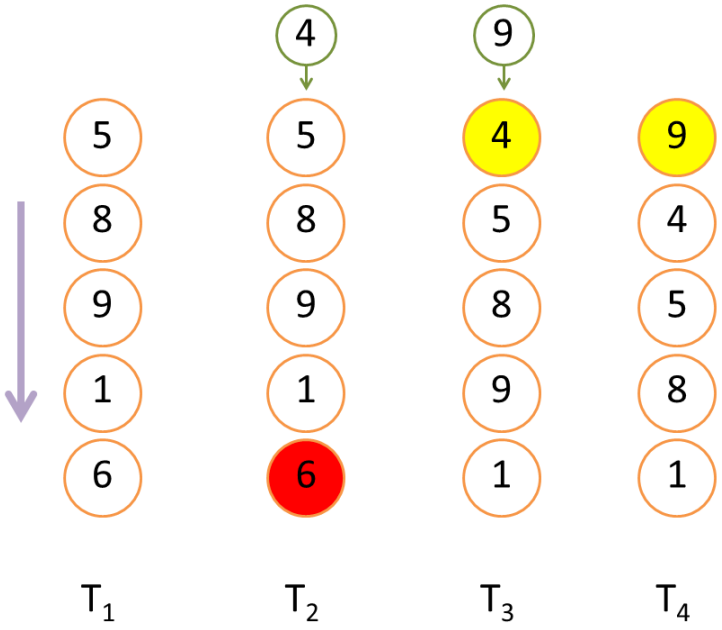
ชื่อกลุ่ม	ตัวอย่างอัลกอริธึมการวางทับข้อมูล
Recency-based	LRU, LRU-threshold, LRU*, LRU-hot, LRU-LSC, SB-LRU,SLRU, HLRU, Pitkow/Recker, EXP1, value-aging, generational replacement
Size-based	SIZE, LRU min, partitioned caching, PSS, CSS, LRU-SP
Frequency-based	LFU, LFU-Aging, LFUDA, Window-LFU, swLFU, AgedswLFU, a-Aging, HYPER-G
Function-based	GD-Size, GDSF, GD*, PGDS, server-assisted cache replacement, TSP, Bolot/Hoschka, MIX, M-Metric, HYBRID, LNCR-W3, LRV, LUV, LR, N-gram
Randomized	RAND, HARMONIC, LRU-C, LRU-S, randomized policies using utility functions

อัลกอริธึมในการวางทับข้อมูลแต่ละกลุ่มจะมีวิธีการทำงานที่แตกต่างกัน รวมไปถึงมีความเหมาะสมกับระบบต่างๆที่ไม่เหมือนกัน ดังนั้นการจะนำแคชไปใช้กับระบบใดจึงจำเป็นต้องคำนึงถึงอัลกอริธึมการวางทับข้อมูลที่เหมาะสมกับระบบนั้นๆอีกด้วย ซึ่งอัลกอริธึมกางวางทับข้อมูลของกลุ่มต่างๆ มีวิธีการทำงานดังต่อไปนี้

จุฬาลงกรณ์มหาวิทยาลัย

2.2.3.1 การวางทับข้อมูลโดยการพิจารณาความทันสมัย (Recency-based)

การวางทับข้อมูลนี้จะพิจารณาจากเวลาล่าสุดที่มีการเรียกใช้งานข้อมูลนั้นๆ ซึ่งสมมติฐานของวิธีการนี้เชื่อว่า ข้อมูลที่ถูกเรียกใช้งานล่าสุดมีความน่าจะเป็นในการถูกเรียกใช้งานซ้ำอีกครั้งสูงกว่าข้อมูลอื่นๆ ดังนั้นเมื่อเนื้อที่ภายในแคชเต็ม ระบบจะทำการนำข้อมูลที่ไม่ได้ถูกเรียกใช้งานเป็นระยะเวลานานที่สุดออกจากแคช แล้วจึงนำข้อมูลใหม่เข้ามาเก็บไว้ในแคช สำหรับอัลกอริธึมการวางทับข้อมูลที่ได้รับความนิยมอย่างมากในกลุ่มนี้ได้แก่ แอลอาร์ยู ซึ่งมีข้อดีคือ คือสามารถนำไปปรับใช้กับระบบต่างๆได้ง่ายและทำงานได้รวดเร็ว แต่มีข้อเสียคืออาจจะพบปัญหาเมื่อเจอกับการจราจรในเครือข่ายแบบฟลัดแอทแท็ค (Flood Attack) หรือแบบแสกนนิง (Scanning) [19]



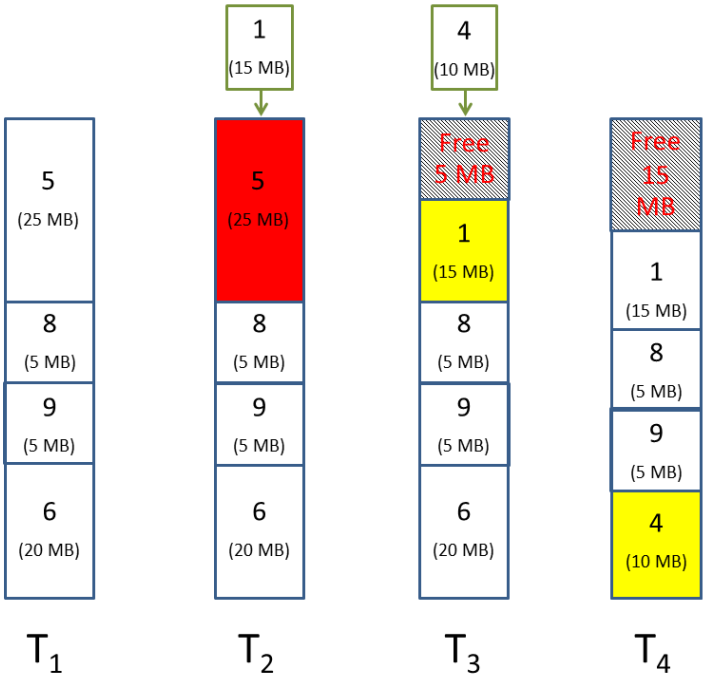
ภาพที่ 2-4 ตัวอย่างการทำงานของอัลกอริธึมการวางทับข้อมูลแอลอาร์ยู

จากภาพที่ 2-4 จะเห็นว่าระบบจะจัดเรียงข้อมูลตามลำดับที่มีการเรียกใช้ล่าสุด โดยข้อมูลที่ถูกเรียกใช้งานล่าสุดจะอยู่ที่หัวแถว ส่วนข้อมูลที่ไม่ถูกเรียกใช้นานที่สุดจะอยู่ท้ายแถว เมื่อแคชเต็มและมีการเรียกใช้งานข้อมูลที่ไม่ได้อยู่ในแคช ระบบจะนำข้อมูลที่อยู่ท้ายแถวออกไป ซึ่งในภาพก็คือข้อมูลหมายเลข 6 จากนั้นจะนำข้อมูลใหม่มาไว้บริเวณหัวแถว หรือก็คือข้อมูลหมายเลข 4 ในภาพ แต่หากมีการเรียกใช้งานข้อมูลที่อยู่ในแคชอยู่แล้ว ระบบจะทำการย้ายข้อมูลนั้นๆมาไว้บริเวณหัวแถว (ข้อมูลหมายเลข 9)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2.2.3.2 การวางทับข้อมูลโดยการพิจารณาขนาด (Size-based)

สำหรับการวางทับข้อมูลลักษณะนี้จะนำขนาดของข้อมูลมาใช้ในการพิจารณา ซึ่งสมมติฐานของวิธีการนี้คือ เมื่อข้อมูลที่อยู่ในแคชมีขนาดไม่เท่ากัน การนำข้อมูลที่มีขนาดใหญ่ที่สุดออกไปในกรณีที่แคชเต็ม จะทำให้สามารถนำข้อมูลใหม่ๆเข้ามาเก็บไว้ในแคชได้มากกว่า ซึ่งจะเพิ่มโอกาสของการพบข้อมูลในแคช (Hit Rate) ให้สูงขึ้น โดยอัลกอริธึมที่ได้รับความนิยมในกลุ่มนี้จะได้แก่ไซส์ (Size) แต่มีข้อเสียคือ หากข้อมูลขนาดใหญ่ที่ถูกนำออกไป เป็นข้อมูลที่ถูกเรียกใช้งานบ่อยๆ จะทำให้ความถี่ในการถ่ายโอนข้อมูลมาจากแหล่งต้นทางเพิ่มสูงขึ้น นอกจากนี้ยังส่งผลกระทบต่อระยะเวลาที่ผู้ใช้งานต้องรอข้อมูลที่ร้องขออีกด้วย

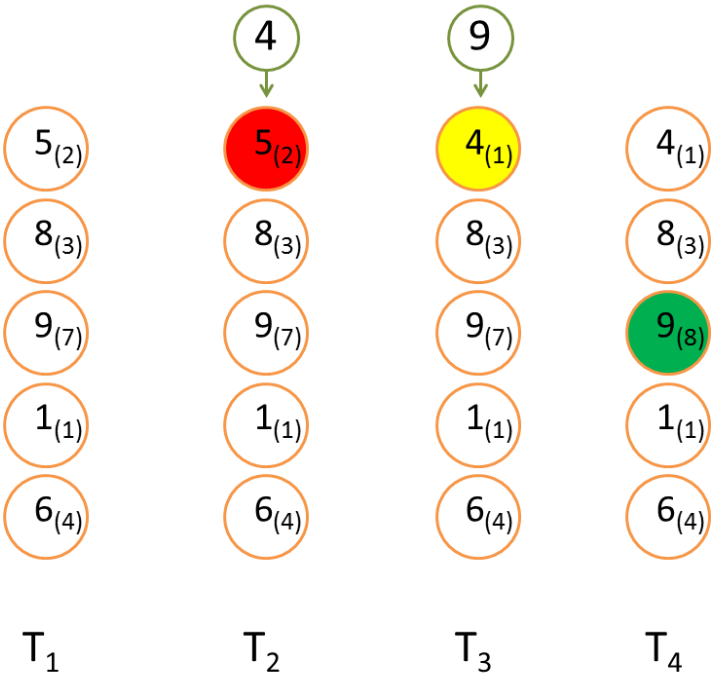


ภาพที่ 2-5 ตัวอย่างการทำงานของอัลกอริธึมการวางทับข้อมูลไซส์

จากภาพที่ 2-5 จะเห็นว่าเมื่อมีการเพิ่มข้อมูลหมายเลข 1 ที่มีขนาด 15 MB ซึ่งเกินความจุของแคชที่เหลืออยู่ ระบบจะนำข้อมูลหมายเลข 5 ที่มีขนาดใหญ่ที่สุดออกไป แล้วนำข้อมูลหมายเลข 1 เข้ามาเก็บไว้ในแคช ต่อมาเมื่อต้องการจะเพิ่มข้อมูลหมายเลข 4 ที่มีขนาด 10 MB ในขณะที่แคชเหลือความจุเพียง 5 MB ระบบจึงต้องนำข้อมูลหมายเลข 6 ซึ่งมีขนาดใหญ่ที่สุด ณ ขณะนั้นออกไป แล้วจึงค่อยนำข้อมูลหมายเลข 4 เข้ามา

2.2.3.3 การวางทับข้อมูลโดยการพิจารณาความถี่ (Frequency-based)

การวางทับข้อมูลโดยการพิจารณาความถี่จะอาศัยการจดบันทึกจำนวนครั้งที่ข้อมูลแต่ละตัวถูกเรียกใช้ และนำข้อมูลที่มีความถี่ในการถูกเรียกใช้งานน้อยที่สุดออกไปในกรณีที่แคชเหลือพื้นที่ไม่เพียงพอกับขนาดของข้อมูลใหม่ ซึ่งวิธีการนี้มีสมมติฐานที่ว่า ข้อมูลใดๆที่ถูกเรียกใช้งานบ่อย มีแนวโน้มที่จะถูกเรียกใช้งานซ้ำในอนาคต ซึ่งอัลกอริธึมที่ได้รับความนิยมมากที่สุดของการวางทับข้อมูลในกลุ่มนี้ได้แก่ แอลเอฟยู ซึ่งมีข้อดีคือสามารถรับมือกับรูปแบบการจราจรในเครือข่ายแบบฟลัดแอทแท็ค หรือแบบแสกนนิงได้ แต่มีข้อเสียคือหากข้อมูลใดๆถูกเรียกใช้งานมากๆในช่วงระยะเวลาหนึ่งแล้วไม่ถูกเรียกใช้งานอีกเลย ข้อมูลนั้นอาจจะค้างอยู่ในแคชจนทำให้ข้อมูลใหม่ๆที่เพิ่งบันทึกลงในแคชถูกนำออกไปเมื่อแคชเต็ม จนระบบไม่สามารถบันทึกข้อมูลใหม่ๆลงไปได้อีก



ภาพที่ 2-6 ตัวอย่างการทำงานของอัลกอริธึมการวางทับข้อมูลแอลเอฟยู

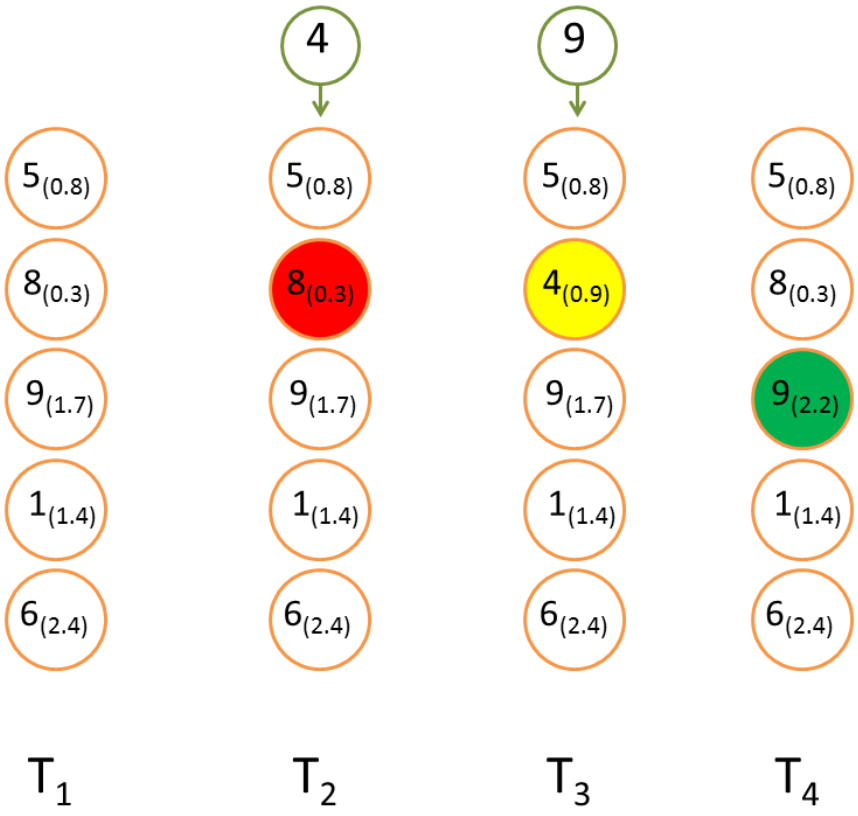
จากภาพที่ 2-6 จะเห็นว่าเมื่อมีการเพิ่มข้อมูลหมายเลข 4 ในขณะที่แคชเต็ม ระบบจะนำข้อมูลหมายเลข 5 ที่มีความถี่ในการถูกเรียกน้อยที่สุดออกไป จากนั้นจึงบันทึกข้อมูลหมายเลข 4 พร้อมทั้งความถี่เริ่มต้นที่ 1 ลงในแคช ถัดมาเมื่อมีการเรียกข้อมูลหมายเลข 9 ซึ่งมีอยู่ในแคชอยู่แล้ว ระบบจะทำการส่งข้อมูลจากแคชไปยังผู้ใช้ทันที พร้อมทั้งเพิ่มความถี่ของข้อมูลหมายเลข 9 จาก 7 ครั้ง เป็น 8 ครั้ง

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

2.2.3.4 การวางทับข้อมูลโดยการกำหนดฟังก์ชัน (Function-based)

การวางทับข้อมูลในลักษณะนี้จะอาศัยฟังก์ชั่นที่มีการคิดขึ้นมาเพื่อกำหนดมูลค่าให้กับข้อมูลแต่ละตัวโดยอาศัยตัวแปรต่างๆ เช่น ความถี่ ขนาด ความทันสมัย และระยะเวลาในการเข้าถึงข้อมูล ซึ่งเราสามารถให้น้ำหนักกับตัวแปรแต่ละตัวไม่เท่ากันได้ โดยจะดูจากความเหมาะสมของระบบต่างๆ เช่น ถ้าเป็นระบบที่ต้องการความเร็วในการเข้าถึงข้อมูล เราก็อาจจะให้น้ำหนักในส่วนนี้สูงกว่าตัวแปรอื่นๆ ซึ่งมูลค่าที่ได้จากการคำนวณจะต้องมีการเก็บบันทึกไว้ควบคู่กับข้อมูล เหมือนกับอัลกอริธึมแอลเอฟยูที่มีการเก็บค่าความถี่ของข้อมูลแต่ละตัวไว้คู่กัน และเมื่อแคชเต็ม ส่วนใหญ่ระบบจะนำข้อมูลที่มูลค่าน้อยที่สุดออกไป เพื่อสร้างพื้นที่สำหรับข้อมูลใหม่ที่จะนำเข้ามา โดยอัลกอริธึมที่ได้รับความนิยมมากที่สุดของการวางทับข้อมูลในกลุ่มนี้ได้แก่ จีดีเอสเอฟ (GDSF) สำหรับข้อดีของอัลกอริธึมลักษณะนี้คือเราสามารถกำหนดฟังก์ชันให้เหมาะสมกับสภาวะในการทำงาน ทำให้แคชสามารถทำงานได้อย่าง

เต็มประสิทธิภาพ ส่วนข้อเสียจะเป็นเรื่องความซับซ้อนในการคำนวณที่บางครั้งจำเป็นจะต้องอาศัยพลังงานในการคำนวณสูง รวมถึงหน่วยความจำขนาดใหญ่ ทำให้ระบบทำงานช้าลง หรือในกรณีที่ฟังก์ชันมีความซับซ้อนมากๆ ก็อาจจะทำให้ไม่สามารถนำไปประยุกต์ใช้ได้



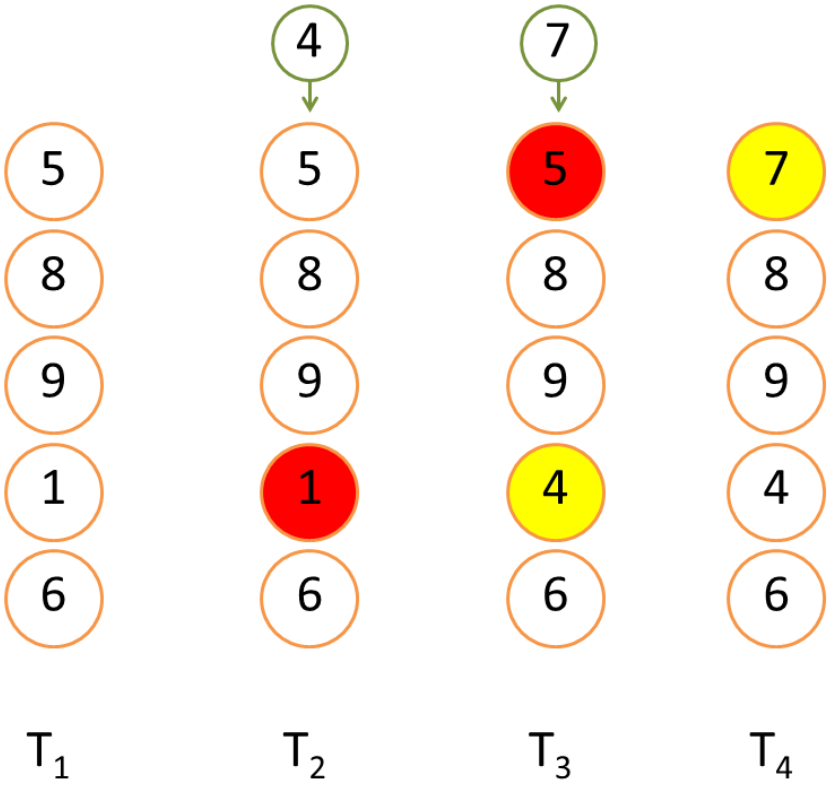
ภาพที่ 2-7 ตัวอย่างการทำงานของอัลกอริธึมการวางทับข้อมูลจีดีเอสเอฟ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

จากภาพที่ 2-7 จะเห็นว่าเมื่อมีการเพิ่มข้อมูลหมายเลข 4 ในขณะที่แคชเต็ม ระบบจะนำข้อมูลหมายเลข 8 ที่มีมูลค่าน้อยที่สุดออกไป จากนั้นจึงบันทึกข้อมูลหมายเลข 4 พร้อมทั้งมูลค่าที่คำนวณได้จากฟังก์ชันลงในแคช ถัดมาเมื่อมีการเรียกข้อมูลหมายเลข 9 ซึ่งมีอยู่ในแคชอยู่แล้ว ระบบจะทำการส่งข้อมูลจากแคชไปยังผู้ใช้ พร้อมทั้งคำนวณมูลค่าใหม่ของข้อมูลหมายเลข 9 แล้วทำการบันทึกมูลค่าใหม่ลงไปในแคช

2.2.3.5 การวางทับข้อมูลโดยการสุ่ม (Randomized)

สำหรับการวางทับข้อมูลวิธีนี้จะอาศัยการสุ่มเลือก ไม่ต้องอาศัยโครงสร้างใดๆในการตัดสินใจ จึงเหมาะสำหรับระบบที่มีหน่วยความจำหรือทรัพยากรทางด้านการคำนวณน้อย ซึ่งนอกจากข้อดีในเรื่องของการใช้ทรัพยากรน้อยแล้วยังมีจุดเด่นในเรื่องของความเร็วอีกด้วย แต่ในขณะเดียวกันก็มีข้อเสียคือ ระบบอาจจะไม่มีประสิทธิภาพ เนื่องจากการสุ่มเลือกข้อมูลออกจากแคชนั้น อาจจะสุ่มไปโดนข้อมูลที่มีการเรียกใช้งานมากๆ ทำให้ต้องร้องขอข้อมูลนั้นใหม่อีกครั้ง สำหรับอัลกอริธึมที่ได้รับความนิยมมากที่สุดของการวางทับข้อมูลในกลุ่มนี้ได้แก่ แรนด์ (RAND)



ภาพที่ 2-8 ตัวอย่างการทำงานของอัลกอริธึมการวางทับข้อมูลแรนด์

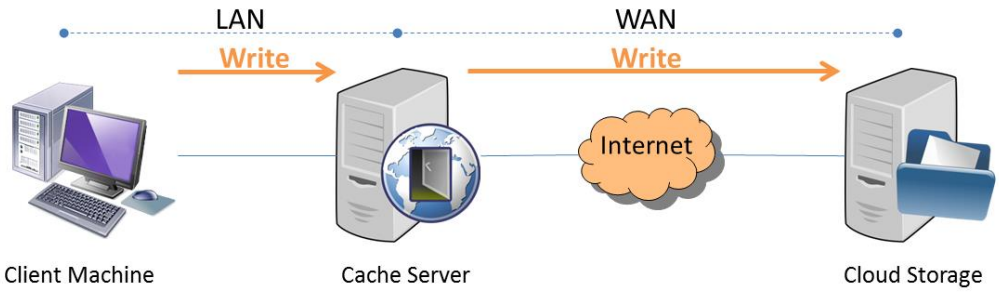
จากภาพที่ 2-8 จะเห็นว่าเมื่อมีการเพิ่มข้อมูลหมายเลข 4 ในขณะที่แคชเต็ม ระบบสุ่มเลือกข้อมูลหมายเลข 1 ออกไป จากนั้นจึงบันทึกข้อมูลหมายเลข 4 ลงในแคช ถัดมาเมื่อมีการเพิ่มข้อมูลหมายเลข 7 ระบบจึงทำการสุ่มเลือกข้อมูลหมายเลข 5 ออกไป แล้วจึงบันทึกข้อมูลหมายเลข 7 ลงในแคช

2.2.4 การเขียนข้อมูลลงสู่แคช (Write Caching)

ในกรณีที่ข้อมูลซึ่งถูกเก็บอยู่ในแคชมีการเปลี่ยนแปลงไป ระบบจะต้องทำการเปลี่ยนแปลงข้อมูลต้นฉบับให้มีความตรงกัน ซึ่งจะมีวิธีในการเขียนข้อมูลอยู่ 2 อย่างด้วยกัน ได้แก่ การเขียนทั้งหมด (Write Through) และ การเขียนทีหลัง (Write Back) [20]

2.2.4.1 การเขียนทั้งหมด

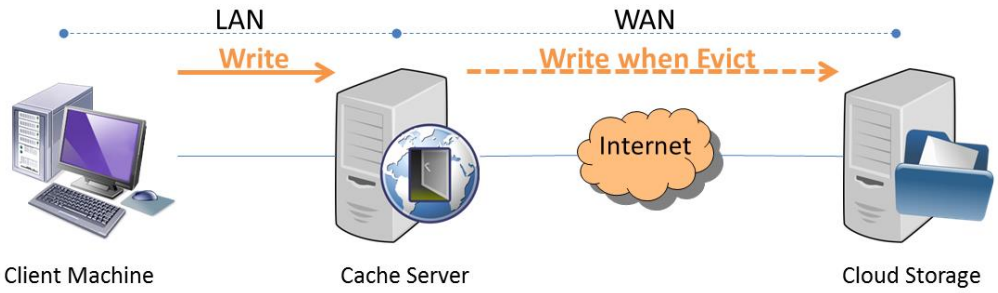
สำหรับการเขียนข้อมูลแบบเขียนทั้งหมดนั้น จะมีขั้นตอนง่ายๆคือ เมื่อข้อมูลในส่วนของผู้ใช้มีการเปลี่ยนแปลง ระบบจะทำการเขียนข้อมูลที่เก็บอยู่ในแคช และข้อมูลต้นฉบับที่อยู่บนพื้นที่เก็บข้อมูลให้มีความตรงกันของข้อมูลในทันที ดังภาพที่ 2-9



ภาพที่ 2-9 การทำงานของการเขียนข้อมูลแบบเขียนทั้งหมด

2.2.4.2 การเขียนทีหลัง

สำหรับการเขียนข้อมูลแบบเขียนทีหลังนั้น จะแต่ต่างจากการเขียนทั้งหมดตรงที่ระบบจะยังไม่ทำการเขียนข้อมูลต้นฉบับที่อยู่บนพื้นที่เก็บข้อมูลทันที แต่จะทำการบันทึกว่าข้อมูลนั้นๆที่อยู่ในแคชมีการเปลี่ยนแปลง และเมื่อถึงเวลาที่ข้อมูลนั้นๆจะต้องถูกนำออกจากแคช ระบบจึงค่อยทำการเขียนไปยังข้อมูลต้นฉบับที่อยู่ในพื้นที่เก็บข้อมูล ดังภาพที่ 2-10



ภาพที่ 2-10 การทำงานของการเขียนข้อมูลแบบเขียนทีหลัง

ซึ่งหากเรานำวิธีการเขียนข้อมูลแบบการเขียนทีหลังมาประยุกต์ใช้กับแคชสำหรับพื้นที่เก็บข้อมูลบนคลาวด์ ก็จะเป็นอีกหนึ่งวิธีในการช่วยลดปริมาณการถ่ายโอนข้อมูลของระบบลง

2.3 งานวิจัยที่เกี่ยวข้อง

2.3.1 ผลกระทบจากการถ่ายโอนข้อมูลในพื้นที่เก็บข้อมูลบนคลาวด์

จากบทความวิจัย [21] ที่สร้างแบบจำลองทางสมการขึ้นมาเพื่อใช้ในการคำนวณว่าองค์กรนั้นๆ เหมาะสมที่จะใช้บริการคลาวด์คอมพิวติงหรือไม่ จะพบว่า 1 ในตัวแปรหลักของสมการก็คือปริมาณข้อมูลที่มีการถ่ายโอน ซึ่งหากเป็นองค์กรที่มีลักษณะการถ่ายโอนข้อมูลบ่อย หรือมีปริมาณมาก จะทำให้ความเหมาะสมของคลาวด์คอมพิวติงกับองค์กรแปรผกผันกับปริมาณข้อมูลที่มีการถ่ายโอน

จากงานวิจัยของ Mayur และคณะ [22] ได้มีการวิเคราะห์ถึงความเป็นไปได้ในการรับเอา Amazon S3 มาใช้งานกับกริดในโครงการวิทยาศาสตร์ โดยหนึ่งในปัจจัยที่ใช้พิจารณาจะได้แก่เรื่องค่าใช้จ่ายที่เกิดจากการนำโครงการ DZero ซึ่งเป็นโครงการฟิสิกส์พลังงานสูงมาใช้เป็นกรณีตัวอย่าง โดยเมื่อพิจารณาจากค่าใช้จ่ายที่เกิดแล้วจะพบว่า ค่าใช้จ่ายที่เกิดขึ้นจากการถ่ายโอนข้อมูลนั้นมีมูลค่าที่ค่อนข้างสูง โดยคิดเป็นร้อยละ 68 ของค่าใช้จ่ายที่เกิดขึ้นจากค่าเช่าพื้นที่

2.3.2 เว็บแคช

จากงานวิจัยของ Podlipnig และคณะ [17] ได้ทำการรวบรวมข้อมูลของอัลกอริธึมการวางทับข้อมูลแบบต่างๆที่ถูกใช้งานใช้อยู่ในเว็บแคช โดยมีการจัดกลุ่มให้กับอัลกอริธึมต่างๆที่มีพื้นฐานคล้ายกันรวมไปถึงอธิบายหลักการทำงานของอัลกอริธึมต่างๆอย่างละเอียด

งานวิจัยของ Wong, K.-Y [18] ได้ทำการรวบรวมอัลกอริธึมการวางทับข้อมูลต่างๆที่มีการนำเสนอออกมา เพื่อจัดกลุ่มและชี้ให้เห็นว่าอัลกอริธึมกลุ่มไหนเหมาะสำหรับแคชชนิดใด ทำให้ผู้ใช้สามารถเลือกอัลกอริธึมการวางทับข้อมูลที่เหมาะสมที่สุดกับระบบของตัวเองมาใช้ได้

2.3.3 พฤติกรรมของพื้นที่เก็บข้อมูลบนเครือข่าย (Network Storage)

งานวิจัยของ Nagasako และ Yamaguchi [23] ได้นำเสนออัลกอริธึมการวางทับข้อมูลแบบใหม่สำหรับพื้นที่เก็บข้อมูลบนเครือข่ายแบบแบ่งบล็อก โดยมีการชี้ให้เห็นว่าพฤติกรรมของพื้นที่เก็บข้อมูลบนเครือข่ายแตกต่างจากลักษณะการทำงานของระบบทั่วๆไป กล่าวคือไม่มีคุณลักษณะของการอ้างอิงข้อมูลเดียวกันบ่อยครั้งในช่วงเวลาใกล้เคียงกัน (Temporal Locality) ทำให้อัลกอริธึมการวางทับข้อมูลแอลอาร์ยูที่ถูกใช้งานอย่างแพร่หลายในทุกๆระบบ ไม่สามารถทำงานได้อย่างมีประสิทธิภาพกับระบบที่มีลักษณะเช่นนี้

งานวิจัยของ Willick และคณะ [24] ได้ทำการทดสอบประสิทธิภาพของอัลกอริธึมการวางทับข้อมูลแบบต่างๆเมื่อถูกนำไปใช้งานเครื่องแม่ข่ายจัดเก็บไฟล์บนเครือข่าย (Network Fileserver) ซึ่งผลการทดลองชี้ให้เห็นว่าอัลกอริธึมการวางทับข้อมูลแอลอาร์ยูที่สามารถทำงานได้ดีในระบบส่วนใหญ่ ไม่สามารถจัดการกับการทำงานในลักษณะนี้ได้อย่างมีประสิทธิภาพ ในขณะที่อัลกอริธึมการวางทับข้อมูลในกลุ่มที่พิจารณาจากความถี่ของการเรียกใช้ข้อมูลเป็นหลักอย่างแอลเอฟยู สามารถทำงานได้มีประสิทธิภาพกว่ามาก

2.3.4 คลาวด์แคช

Banditwattanawong [25] ได้นำเสนอวิธีการวางทับข้อมูลสำหรับคลาวด์โดยมีจุดประสงค์เพื่อปรับปรุงประสิทธิภาพและลดค่าใช้จ่ายที่เกิดจากการถ่ายโอนข้อมูล ซึ่งผลการทดสอบได้แสดงให้เห็นว่าวิธีการวางทับข้อมูลของเขามีประสิทธิภาพที่ดีกว่าวิธีอื่นๆ อย่างไรก็ตามภาระงานที่ถูกนำมาใช้ในงานวิจัย [25] ไม่ใช่ภาระงานที่สะท้อนถึงสภาพแวดล้อมที่แท้จริงของการทำงานภายใต้พื้นที่เก็บข้อมูลบนคลาวด์ เนื่องจากภาระงานทั้งหมดเป็นภาระงานที่ได้มาจากเว็บแคช และตัวแปรบางส่วนในสมการที่ใช้คำนวณมูลค่าข้อมูลยังได้รับอิทธิพลมาจากลักษณะภาระงานประเภทเว็บแคชอีกด้วย เช่น ค่าความหน่วง (Latency) นอกจากนี้ในการหาค่า Time-To-Live เพื่อนำมาใช้กับสมการยังจำเป็นจะต้องประมวลผลภาระงานก่อนที่จะนำมาทดลอง ซึ่งนั่นจะทำให้เกิดความยุ่งยากในกรณีที่ต้องการนำไปปรับใช้งานจริง สมการคำนวณมูลค่าข้อมูลในงานวิจัยของ Banditwattanawong [25] จะเป็นไปตามสมการที่ 2-1

$$s_i \times c_i \times l_i \times f_i \times TTL_i \quad \text{(2-1)}$$

โดยที่ s_i เป็นขนาดของข้อมูล i

c_i คือค่าใช้จ่ายในการดาวน์โหลดข้อมูล i

l_i คือความหน่วงของการดาวน์โหลดข้อมูล i

f_i คือความถี่ในการเรียกใช้งานข้อมูล i

TTL_i คือค่า Time-To-Live ของข้อมูล i

ดังนั้นหากเราตัดตัวแปร l_i และ TTL_i ซึ่งเป็นตัวแปรที่ไม่ปรากฏอยู่ในภาระงานจาก NFS Server ของเราออกจากสมการ จะส่งผลให้วิธีการวางทับข้อมูลตามสมการที่ 1 ทำงานเหมือนกับแอลเอฟยู ซึ่งจะประสบปัญหาแคชทำงานผิดพลาดจากข้อมูลที่เคยได้รับความนิยมอย่างมากในอดีต [17]

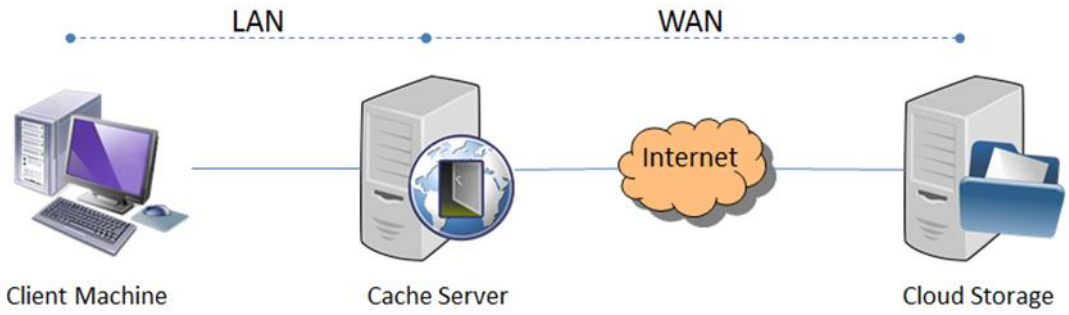
จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

บทที่ 3
การออกแบบอัลกอริธึมการวางทับข้อมูลโดยการเปรียบเทียบมูลค่า

ในบทนี้จะกล่าวถึงระบบแคชสำหรับพื้นที่เก็บข้อมูลบนคลาวด์ ที่มีการแบ่งการทำงานออกเป็น 4 ส่วน ได้แก่ การแบ่งข้อมูลออกเป็นบล็อก การจัดเก็บข้อมูลลงแคช การเขียนข้อมูลลงแคช และการกำหนดมูลค่าให้ข้อมูลแต่ละตัว

3.1 ภาพรวมของแคช

3.1.1 โครงสร้างของแคช

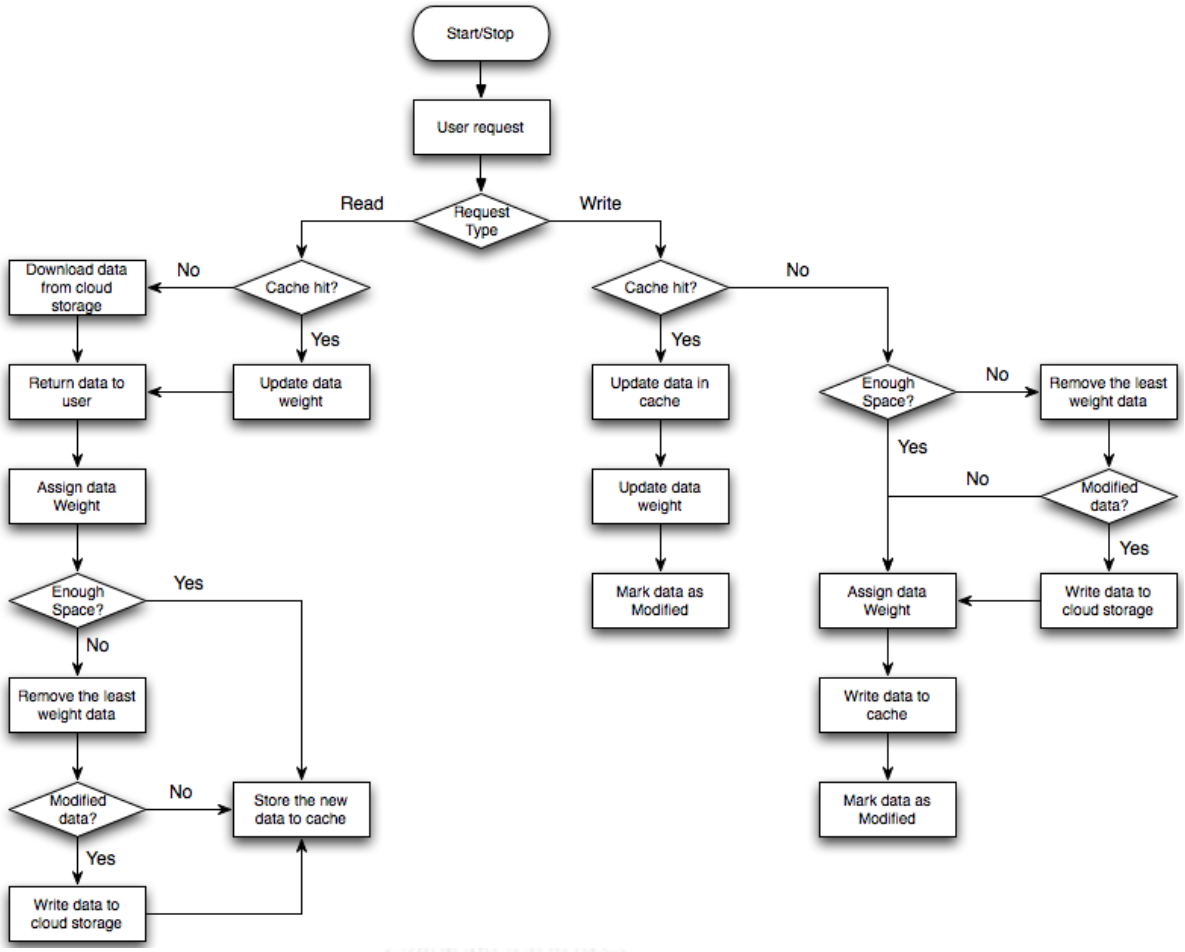


ภาพที่ 3-1 โครงสร้างของคลาวด์แคช

สำหรับโครงสร้างของแคชในงานวิจัยนี้จะมีรูปแบบดังภาพที่ 3-1 ซึ่งมีขั้นตอนในการทำงานหลักๆคือ เมื่อผู้ใช้เรียกหาข้อมูล คำร้องขอจะถูกส่งมายังเครื่องแม่ข่ายแคช ซึ่งเครื่องแม่ข่ายแคชจะคอยจัดการกับคำร้องขอจากผู้ใช้ทุกคน โดยทำงานเสมือนว่าข้อมูลทั้งหมดถูกเก็บอยู่ในเครื่องแม่ข่ายแคช ทำให้ไม่เกิดปัญหาในเรื่องของข้อมูลที่ไม่ตรงกันระหว่างผู้ใช้ต่างๆ (Cache coherency)

3.1.2 การทำงานของแคช

การทำงานของแคชมีภาพรวมเป็นดังภาพที่ 3-2 ซึ่งจะประกอบไปด้วย 3 ขั้นตอนใหญ่ๆ ได้แก่ การจัดเก็บข้อมูลที่ดาวน์โหลดจากคลาวด์ลงสู่แคช การเขียนข้อมูลจากผู้ใช้ลงสู่แคช และการกำหนดมูลค่าให้ข้อมูลแต่ละตัว ซึ่งจะกล่าวถึงในหัวข้อถัดไป



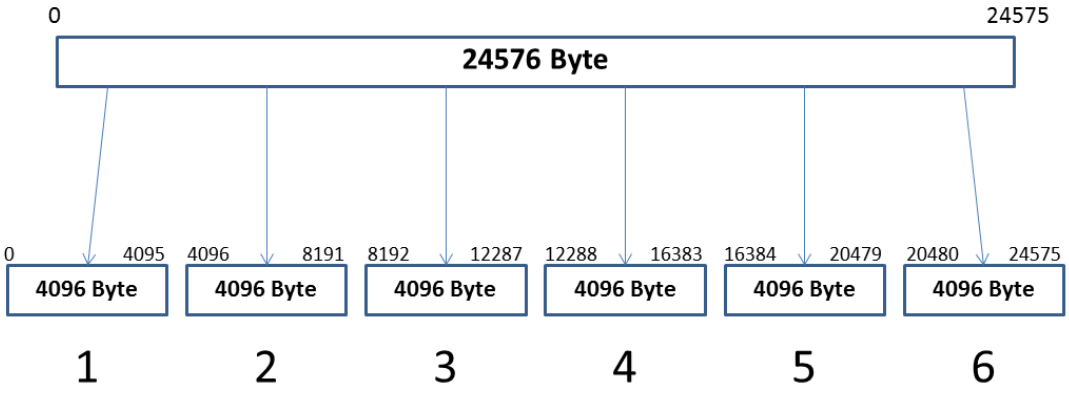
ภาพที่ 3-2 ขั้นการทำงานทั้งหมดของแคช

สำหรับแคชที่จะนำมาใช้กับพื้นที่เก็บข้อมูลบนคลาวด์ จะมีหลักการทำงานที่คล้ายคลึงกับเว็บแคช แต่จะมีการเพิ่มการทำงานในส่วนของการเขียนข้อมูลจากผู้ใช้ลงสู่แคชเข้าไปเพื่อจัดการปริมาณการถ่ายโอนข้อมูลในลักษณะของการอัพโหลดข้อมูลขึ้นไปสู่พื้นที่เก็บข้อมูลบนคลาวด์ รวมไปถึงการนำอัลกอริธึมการวางทับข้อมูลโดยการเปรียบเทียบมูลค่าเข้ามาใช้เป็นเครื่องมือในการตัดสินใจของแคชนี้ด้วย

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

3.2 การแบ่งข้อมูลออกเป็นบล็อก

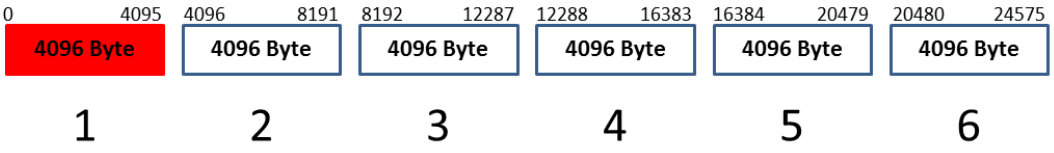
การแบ่งข้อมูลออกเป็นบล็อก (จากหัวข้อ 2.2.2) จะเป็นขั้นตอนในการจัดการข้อมูลที่เกิดจากการร้องขอหรือเขียนข้อมูลจากผู้ใช้ โดยการแบ่งข้อมูลนั้นๆ ออกเป็นส่วนย่อยๆตามขนาดที่กำหนด จากนั้นระบุตำแหน่งเริ่มต้นและตำแหน่งสิ้นสุดของแต่ละบล็อก แล้วค่อยจัดเก็บลงในแคชดังภาพที่ 3-3



ภาพที่ 3-3 ตัวอย่างการแบ่งข้อมูลออกเป็นบล็อกย่อยๆ

โดยการแบ่งข้อมูลออกเป็นบล็อกจะมีรูปแบบที่เกิดขึ้นได้ 3 กรณีดังนี้

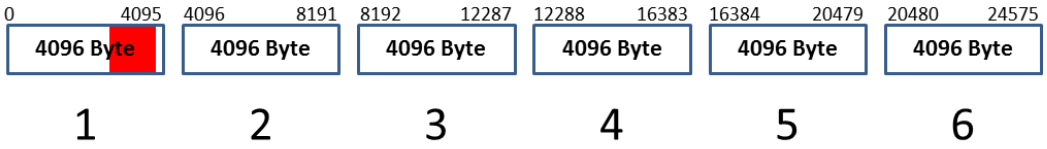
1. ข้อมูลที่ผู้ใช้ร้องขอหรือเขียนมีขอบเขตของตำแหน่งพอดีกับบล็อกที่กำหนดดังภาพที่ 3-4 เมื่อผู้ใช้ร้องขอหรือเขียนข้อมูลโดยเริ่มต้นจากตำแหน่งที่ 0 ถึง 4095 ข้อมูลที่จะถูกใช้งานก็คือบล็อกที่ 1 ในภาพ



ภาพที่ 3-4 ตัวอย่างข้อมูลที่มีขอบเขตของตำแหน่งพอดีบล็อก

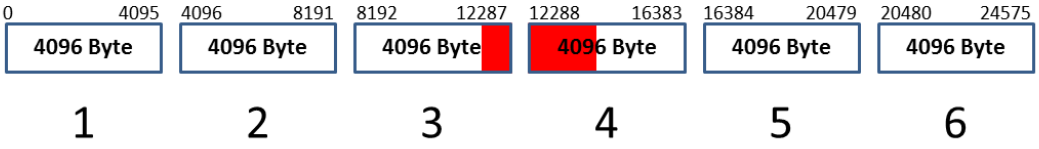
CHULALONGKORN UNIVERSITY

2. ข้อมูลที่ผู้ใช้ร้องขอหรือเขียนมีขอบเขตของตำแหน่งอยู่ในบางส่วนของบล็อกที่กำหนด ดังภาพที่ 3-5 เมื่อผู้ใช้ร้องขอหรือเขียนข้อมูลโดยเริ่มต้นจากตำแหน่งที่ 3546 ถึง 4001 ในกรณีนี้ระบบจะทำการดึงบล็อกข้อมูลที่ผู้ใช้ต้องการออกมาทั้งบล็อก แม้ว่าจะมีการเรียกใช้บางส่วนก็ตามซึ่งหากสังเกตจากภาพ ตำแหน่งที่ถูกใช้งานจะอยู่ภายในบล็อกที่ 1



ภาพที่ 3-5 ตัวอย่างข้อมูลที่มีขอบเขตของตำแหน่งอยู่ในบางส่วนของบล็อก

3. ข้อมูลที่ผู้ใช้ร้องขอหรือเขียนมีขอบเขตของตำแหน่งอยู่ระหว่างบล็อกที่กำหนดมากกว่า 1 บล็อกดังภาพที่ 3-6 เมื่อผู้ใช้ร้องขอหรือเขียนข้อมูลโดยเริ่มต้นจากตำแหน่งที่ 12000 ถึง 15000 ในกรณีนี้ระบบจะทำการดึงบล็อกข้อมูลที่ผู้ใช้ต้องการออกมาทั้ง 2 บล็อก แม้ว่าจะมีการเรียกใช้บางส่วนก็ตามซึ่งหากสังเกตจากภาพ ตำแหน่งที่ถูกใช้งานจะอยู่ภายในบล็อกที่ 3 และ 4



ภาพที่ 3-6 ตัวอย่างข้อมูลที่มีขอบเขตของตำแหน่งอยู่ระหว่างบล็อก

3.3 การจัดเก็บข้อมูลลงแคช

การทำงานส่วนแรกของระบบแคชในงานวิจัยนี้ จะเป็นการจัดเก็บข้อมูลลงแคชซึ่งจะมีขั้นตอนในการทำงานดังนี้

1. เมื่อผู้ใช้ร้องขอข้อมูล ระบบจะทำการค้นหาข้อมูลภายในแคชก่อนเป็นอันดับแรก หากเจอก็จะทำการส่งข้อมูลนั้นๆกลับไปให้ผู้ใช้ แล้วทำการปรับปรุง (Update) มูลค่าของข้อมูลนั้นๆที่อยู่ในแคช

2. หากข้อมูลที่ผู้ใช้ร้องขอไม่อยู่ในแคช ระบบจะทำการดาวน์โหลดข้อมูลมาจากคลาวด์แล้วทำการส่งข้อมูลกลับไปให้ผู้ใช้ จากนั้นจึงทำการกำหนดมูลค่าให้แก่ข้อมูลนั้นๆ ซึ่งจะได้มีการอธิบายถึงวิธีการกำหนดมูลค่าของข้อมูลเพิ่มเติมในหัวข้อที่ 3.4

3. ระบบจะทำการตรวจสอบพื้นที่ในแคชเพื่อดูว่ายังมีพื้นที่ว่างเหลืออยู่หรือไม่ หากมีก็จะทำการเก็บข้อมูลใหม่ลงไปในแคชทันที

4. หากมีที่ว่างไม่เพียงพอระบบจะค้นหาข้อมูลที่มีมูลค่าน้อยที่สุด จากนั้นตรวจสอบว่าข้อมูลนั้นๆมีการเปลี่ยนแปลงไปจากต้นฉบับหรือไม่ ถ้าไม่มีการเปลี่ยนแปลง ระบบจะนำข้อมูลนั้นๆออกไป แล้วบันทึกข้อมูลใหม่ลงแคช

5. หากข้อมูลที่มีมูลค่าน้อยที่สุดมีการเปลี่ยนแปลงไปจากต้นฉบับ ระบบจะทำการอัพโหลดข้อมูลนั้นๆไปที่คลาวด์ก่อนนำออกไปจากแคช จากนั้นจึงบันทึกข้อมูลใหม่ลงแคช

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
algorithm CACHING
input DATA REQUEST
begin
    if REQUESTED DATA found in CACHE
        update DATA WEIGHT by using DATA WEIGHT EQUATION
        Return REQUESTED DATA to USER
    else
        download REQUESTED DATA from CLOUD STORAGE
        assign DATA WEIGHT by using DATA WEIGHT EQUATION
        Return REQUESTED DATA to USER
        if CACHE has free space
            save REQUESTED DATA into CACHE
        else
            if LEAST WEIGHT DATA modified
                write LEAST WEIGHT DATA to CLOUD STORAGE
                evict LEAST WEIGHT DATA from CACHE
                save REQUESTED DATA into CACHE
            else
                evict LEAST WEIGHT DATA from CACHE
                save REQUESTED DATA into CACHE
end
```

ภาพที่ 3-7 คำสั่งเทียมแสดงการทำงานในส่วนของการจัดเก็บข้อมูลลงแคช

3.4 การเขียนข้อมูลลงแคช

การทำงานในส่วนถัดมาจะได้แก่ขั้นตอนการเขียนข้อมูลของผู้ใช้ลงสู่แคช (จากหัวข้อ 2.2.4) ซึ่งในงานวิจัยนี้เราจะเลือกใช้วิธีการเขียนทีหลังมาประยุกต์ใช้กับอัลกอริธึมการวางทับข้อมูลโดยการเปรียบเทียบมูลค่า โดยมีจุดประสงค์หลักเพื่อลดปริมาณการถ่ายโอนข้อมูลระหว่างแคชกันคลาวด์ในส่วนที่เป็นการอัพโหลด สำหรับขั้นตอนการทำงานจะเป็นดังนี้

1. เมื่อผู้ใช้เขียนข้อมูล ระบบจะทำการค้นหาข้อมูลนั้นๆภายในแคช หากเจอก็จะทำการปรับปรุงข้อมูลให้ตรงกับข้อมูลที่ผู้ใช้เขียนมา แล้วทำการปรับปรุง (Update) มูลค่าของข้อมูลนั้นๆที่อยู่ในแคช จากนั้นจึงบันทึกไว้ว่าข้อมูลมีการเปลี่ยนแปลงไปจากต้นฉบับ

2. หากไม่พบข้อมูลนั้นๆในแคช ระบบจะทำการตรวจสอบพื้นที่ในแคชเพื่อดูว่ายังมีพื้นที่ว่างเหลืออยู่หรือไม่ หากมีก็จะทำการเขียนข้อมูลลงไปในแคช แล้วทำการกำหนดมูลค่าให้กับข้อมูลนั้นๆ แล้วจึงบันทึกว่าข้อมูลนั้นๆมีการเปลี่ยนแปลงไปจากต้นฉบับ

3. หากมีที่ว่างไม่เพียงพอระบบจะค้นหาข้อมูลที่มีมูลค่าน้อยที่สุด จากนั้นตรวจสอบว่าข้อมูลนั้นๆมีการเปลี่ยนแปลงไปจากต้นฉบับหรือไม่ ถ้าไม่มีการเปลี่ยนแปลง ระบบจะนำข้อมูลนั้นๆออกไป แล้วเขียนข้อมูลลงไปในแคช จากนั้นจะทำการกำหนดมูลค่าให้กับข้อมูลนั้นๆ แล้วจึงบันทึกว่าข้อมูลนั้นๆมีการเปลี่ยนแปลงไปจากต้นฉบับ

4. หากข้อมูลที่มีมูลค่าน้อยที่สุดมีการเปลี่ยนแปลงไปจากต้นฉบับ ระบบจะทำการอัพโหลดข้อมูลนั้นๆไปที่คลาวด์ก่อนนำออกไปจากแคช จากนั้นจึงเขียนข้อมูลลงไปในแคช แล้วทำการกำหนดมูลค่าให้กับข้อมูลนั้นๆ ตามด้วยการบันทึกว่าข้อมูลนั้นๆมีการเปลี่ยนแปลงไปจากต้นฉบับ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
algorithm WRITE BACK POLICY
input DATA WRITING
begin
    if WRITTEN DATA found in CACHE
        update DATA WEIGHT by using DATA WEIGHT EQUATION
        update DATA in CACHE
    else
        assign DATA WEIGHT by using DATA WEIGHT EQUATION
        if CACHE has free space
            write DATA into CACHE
        else
            if LEAST WEIGHT DATA modified
                write LEAST WEIGHT DATA to CLOUD STORAGE
                evict LEAST WEIGHT DATA from CACHE
                write DATA into CACHE
            else
                evict LEAST WEIGHT DATA from CACHE
                write DATA into CACHE
    mark DATA as modified
end
```

CHULALONGKORN UNIVERSITY

ภาพที่ 3-8 คำสั่งเทียมแสดงการทำงานในส่วนของการเขียนข้อมูลลงแคช

3.5 การกำหนดมูลค่าของข้อมูล

สำหรับการกำหนดมูลค่าของข้อมูลถือเป็นอีกหนึ่งขั้นตอนสำคัญของการออกแบบอัลกอริธึมการวางทับข้อมูล เพราะเป็นส่วนสำคัญที่ใช้กำหนดว่าข้อมูลไหนมีความสำคัญมากกว่ากัน โดยเมื่อพิจารณาจากงานวิจัยที่เกี่ยวข้องในหัวข้อ 2.3.3 และ 2.3.4 แล้วสามารถสรุปได้ว่าอัลกอริธึมในการวางทับข้อมูลที่เหมาะสมกับพื้นที่เก็บข้อมูลบนคลาวด์นั้น จะเป็นอัลกอริธึมในกลุ่มของ “การวางทับข้อมูลโดยการพิจารณาความถี่” อย่างไรก็ตามการใช้งานพื้นฐานการทำงานของการวางทับข้อมูลในกลุ่มนี้จะมีการใช้ประโยชน์จากความถี่ในการอ้างอิงเท่านั้น ซึ่งยังขาดคุณลักษณะที่สำคัญของพื้นที่เก็บข้อมูลบนคลาวด์นั่นก็คือเรื่องของค่าใช้จ่ายในการถ่ายโอนข้อมูล ดังนั้นแนวทางในการออกแบบอัลกอริธึมการวางทับข้อมูลภายในงานวิจัยนี้ จะเป็นการนำหลักการทำงาน “การวางทับข้อมูลโดยการพิจารณาความถี่” มาปรับปรุงโดยการเพิ่มตัวแปรในเรื่องของค่าใช้จ่ายที่เกิดจากการดาว์นโหลดหรืออัพโหลดข้อมูลระหว่างแคชกับพื้นที่เก็บข้อมูลบนคลาวด์เข้าไปดังสมการที่ 3-1

$$w_i = (f_i \times c_i \times s_i) + w_{evict} \tag{3-1}$$

โดยที่ f_i เป็นความถี่ของข้อมูล i ที่ถูกเรียกใช้งาน โดยจะมีการเพิ่มจำนวนขึ้นครั้งละ 1 หน่วย ต่อการเรียกใช้งาน 1 ครั้ง

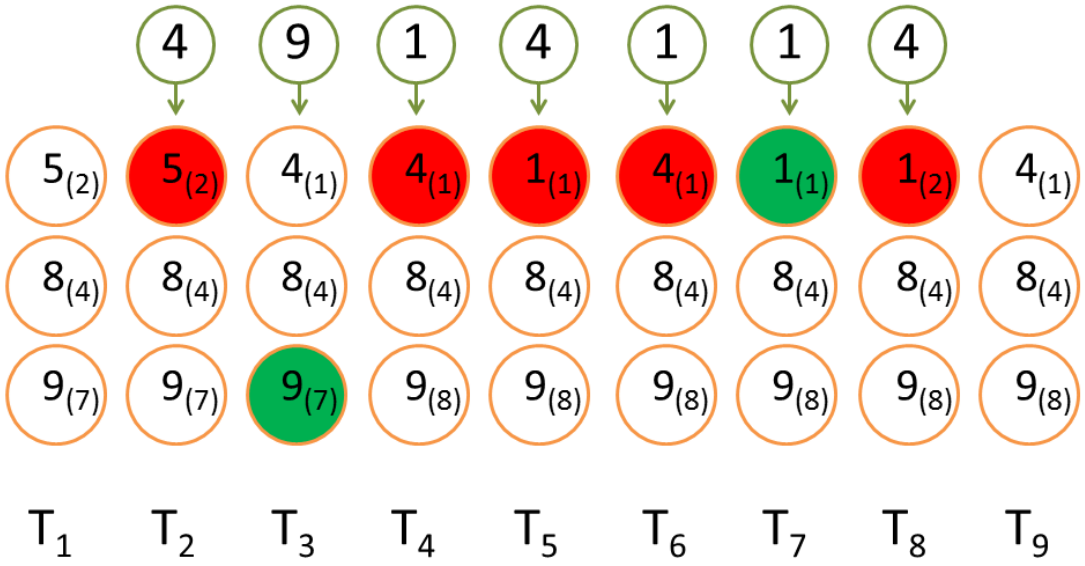
c_i เป็นค่าใช้จ่ายที่เกิดขึ้นจากการถ่ายโอนข้อมูล i ระหว่างแคชกับพื้นที่เก็บข้อมูลบนคลาวด์

s_i เป็นขนาดของข้อมูล i

w_{evict} เป็นมูลค่าของข้อมูลล่าสุดที่ถูกนำออกไปจากแคช โดย w_{evict} จะมีค่าเริ่มต้นเป็น 0

จากสมการที่ 3-1 จะเห็นว่าหนึ่งในตัวแปรหลักที่ส่งผลกระทบต่อมูลค่าของข้อมูลก็คือความถี่ ซึ่งการใช้งานความถี่เข้ามาเป็นตัวแปรจะมีข้อเสียเช่นเดียวกับอัลกอริธึมการวางทับข้อมูลแบบแอลเอฟยู กล่าวคือข้อมูลที่เคยได้รับความนิยมอย่างสูงในช่วงระยะเวลาหนึ่ง อาจจะไม่ถูกเรียกใช้งานอีกแล้วในอนาคต ฉะนั้นจึงมีโอกาสที่ข้อมูลเหล่านี้จะค้างอยู่ในแคช ทำให้สูญเสียพื้นที่ไปอย่างเปล่าประโยชน์ และหากมีข้อมูลประเภทนี้อยู่ในแคชเยอะๆก็จะส่งผลให้ข้อมูลใหม่ๆ ไม่มีโอกาสถูกจัดเก็บลงแคช ส่งผลให้กลายเป็นระบบที่เสมือนว่าไม่มีแคชอยู่

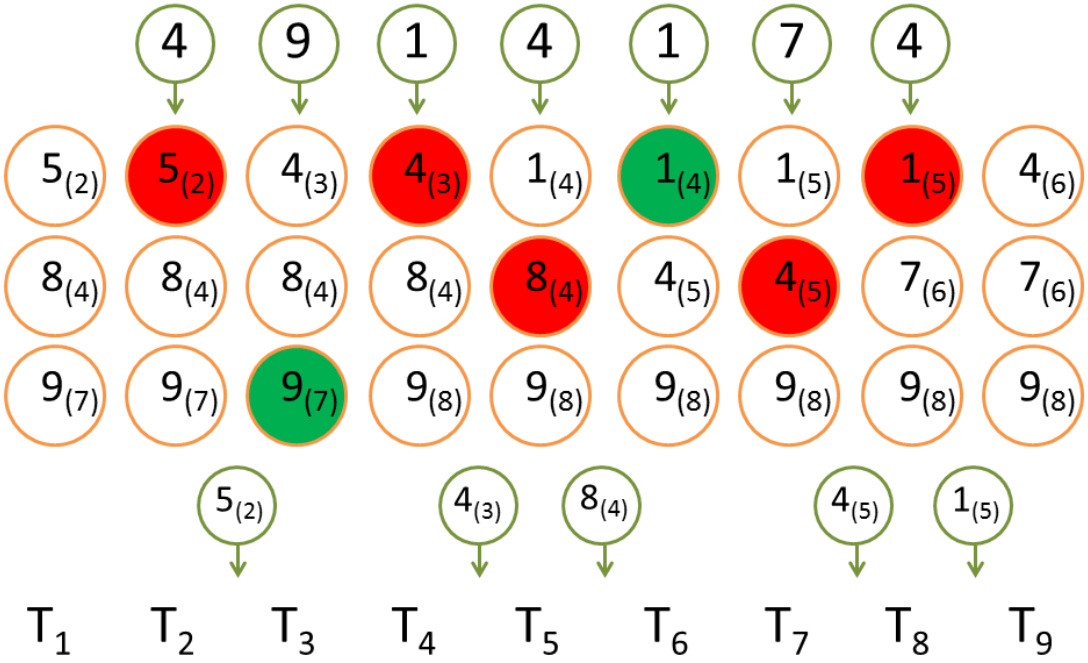
จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



ภาพที่ 3-9 ตัวอย่างปัญหาที่เกิดขึ้นกับอัลกอริธึมการวางทับข้อมูลแอลเอฟยู

จากภาพที่ 3-3 จะเห็นได้ว่าเมื่อแคชเต็ม การทำงานของอัลกอริธึมแบบแอลเอฟยูจะเลือกเอาข้อมูลที่มีความถี่ของการถูกใช้งานน้อยที่สุดออกไป ซึ่งในเวลา T_2 ก็คือข้อมูลหมายเลข 5 และหลังจากเวลา T_4 เป็นต้นไป จะเห็นว่าการเรียกใช้งานข้อมูลจากผู้ใช้จะมีรูปแบบที่สลับไปมาระหว่างข้อมูลหมายเลข 1 กับข้อมูลหมายเลข 4 ซึ่งจะทำให้ข้อมูลทั้งสองนี้สลับกันเข้าออกแคช เพราะข้อมูลหมายเลข 8 และ 9 มีความถี่ที่มากกว่า ถึงแม้ว่าจะไม่ถูกเรียกใช้งานอีกแล้วก็ตาม จึงส่งผลให้ระบบทำงานเสมือนว่าไม่มีแคชอยู่ ซึ่งหนึ่งในวิธีแก้ไขปัญหาที่ได้ผลมากที่สุดและมีความซับซ้อนน้อยที่สุดก็คือ ไดนามิก เอจจิง (Dynamic Aging) โดยหลักการสำคัญของวิธีนี้ก็การบวกความถี่ของข้อมูลสุดท้ายที่ถูกนำออกไปจากแคชเข้ากับความที่ของข้อมูลใหม่ ซึ่งจะส่งผลให้ความถี่ของข้อมูลที่เพิ่งเข้ามาใหม่มีค่าสูงขึ้นทุกครั้ง จนในที่สุดข้อมูลที่เคยมีความถี่ๆก็จะตกลงมาเป็นข้อมูลที่มีความถี่เป็นอันดับท้ายๆของแคช ดังภาพที่ 3-5

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



ภาพที่ 3-10 การแก้ไขปัญหาที่เกิดกับอัลกอริธึมการวางทับข้อมูลแอลเอฟยูด้วยไดนามิก เอจจิง

ในงานวิจัยนี้ได้นำเอาวิธีการแก้ปัญหาด้วยไดนามิก เอจจิงเข้ามาปรับใช้ภายในสมการกำหนดมูลค่าของข้อมูล ซึ่งก็คือตัวแปร W_{evict} ภายในสมการ เนื่องจากมีความซับซ้อนต่ำจึงทำให้สามารถคำนวณมูลค่าให้แก่ข้อมูลต่างๆได้อย่างรวดเร็ว

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 4
การทดลองและผลการทดลอง

สำหรับในบทนี้จะกล่าวถึงรายละเอียดการทดลองของงานวิจัยซึ่งประกอบไปด้วย รายละเอียดของภาระงาน วิธีการทดลอง การประเมินประสิทธิภาพของการวางทับข้อมูลแบบต่างๆ และแบบจำลองอัตราค่าบริการที่ใช้ภายในงานวิจัยนี้ ส่วนสุดท้ายจะเป็นการสรุปและวิเคราะห์ผลการทดลองอย่างละเอียด โดยงานวิจัยชิ้นนี้จะใช้วิธีจำลองการทำงานของระบบแคชขึ้นมาโดยใช้ข้อมูลประวัติการใช้งานจริงที่อยู่ในภาระงานมาประมวลผล

4.1 ภาระงาน

ภาระงานที่ใช้ภายในงานวิจัยนี้เป็นข้อมูลที่เก็บมาจากเครื่องแม่ข่ายเอ็นเอฟเอส (NFS Server) ขนาด 4 เทราไบต์ที่อยู่ในคณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ตั้งแต่วันที่ 28 ส.ค. 2554 จนถึงวันที่ 3 กันยายน 2554 ซึ่งจะมีคุณลักษณะของภาระงานตามที่แสดงอยู่ในตางรางที่ 4-1 ทั้งนี้ HR_∞ และ BHR_∞ คือค่า Hit Rate และ Byte Hit Rate เมื่อแคชมีเนื้อที่ไม่จำกัดตามลำดับ และเนื่องจากงานวิจับนี้ได้กำหนดให้แบบจำลองแบ่งข้อมูลออกเป็นบล็อกที่มีขนาด 4,096 ไบต์ จึงทำให้ HR_∞ และ BHR_∞ มีค่าเท่ากัน

ตารางที่ 4-1 คุณลักษณะของภาระงาน

การร้องขอข้อมูลทั้งหมด	244,802,026 ครั้ง (933.85 GB)
การร้องขอข้อมูลที่ไม่ซ้ำกัน	125,801,287 ครั้ง (479.89 GB)
HR_∞	51.39%
BHR_∞	51.39%

4.2 วิธีการทดลอง

งานวิจัยนี้จะใช้วิธีการจำลองการทำงานของแคชขึ้นมาภายใต้สภาพแวดล้อมของพื้นที่เก็บข้อมูลบนคลาวด์เพื่อเปรียบเทียบประสิทธิภาพของวิธีการวางทับข้อมูลในแบบต่างๆ โดยใช้โปรแกรมจำลองการทำงานของแคชที่พัฒนาขึ้นจากภาษาจาวา ซึ่งจะมีการกำหนดเงื่อนไขต่างๆในการทดลองดังนี้

1. ขนาดของแคชที่ใช้ในการทดลองจะมีทั้งหมด 5 ขนาด ได้แก่ 10% 20% 30% 40% และ 50% ของจำนวนข้อมูลที่ถูกร้องขอโดยไม่ซ้ำกัน
2. ขนาดของบล็อกข้อมูลที่ใช้ในการทดลองจะมี 2 ขนาด ได้แก่ 4,096 และ 8,192 ไบต์
3. อัลกอริธึมการวางทับข้อมูลที่ใช้ในการทดลองนี้มีทั้งหมด 4 ชนิด ได้แก่ แอลอาร์ยู แอลเอฟยู แอลเอฟยูดีเอ และการเปรียบเทียบมูลค่า (Cost-based Replacement Policy) ซึ่งหลังจากนี้เราจะเรียกสั้นๆว่า ซีบีอาร์พี (CBRP)

นอกจากนี้ยังมีรายละเอียดอื่นๆเช่น ค่าแบนด์วิดธ์ในระบบ เวลาจากการหน่วงสัญญาณ (Propagation delay) ซึ่งสามารถสรุปรายละเอียดทั้งหมดได้ดังตารางที่ 4-2

ตารางที่ 4-2 รายละเอียดในการทดลอง

คุณสมบัติ	รายละเอียด
ขนาดของแคช 100%	479.89 GB
ขนาดของแคชที่ใช้ในการทดลอง	10%-50%
ขนาดของบล็ฮกข้อมูล	4,096 และ 8,192 Byte
วิธีการวางทับข้อมูล	LRU, LFU, LFUDA, CBRP
ค่าแบนด์วิดธ์และเวลาจากการหน่วงสัญญาณในเครือข่ายท้องถิ่น	100 Mbps และ 1 ms
ค่าแบนด์วิดธ์และเวลาจากการหน่วงสัญญาณในเครือข่ายบริเวณกว้าง	2 Mbps และ 30 ms

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

4.3 การประเมินประสิทธิภาพ

งานวิจัยนี้จะมีการประเมินประสิทธิภาพของอัลกอริธึมการวางทับข้อมูลซีบีอาร์พี โดยการเปรียบเทียบกับแอลอาร์ยูและแอลเอฟยู โดยจะมีตัวชี้วัดที่สำคัญคือ ปริมาณการถ่ายโอนข้อมูล และค่าใช้จ่าย นอกจากนี้ยังประกอบไปด้วยตัวชี้วัดอื่นๆตามตารางที่ 4-3

ตารางที่ 4-3 ตัวชี้วัดประสิทธิภาพ

ชนิด	รายละเอียด
ปริมาณการถ่ายโอนข้อมูล (Data Transfer)	เป็นการวัดปริมาณการดาวน์โหลดหรืออัพโหลดข้อมูลที่เกิดขึ้นระหว่างแคชกับพื้นที่เก็บข้อมูลบนคลาวด์
ปริมาณแบนด์วิดธ์ที่ลดลง (Bandwidth Saving)	เป็นการวัดปริมาณแบนด์วิดธ์ที่ลดลงเมื่อเปรียบเทียบระหว่างระบบที่มีแคชและไม่มีแคช
Hit Rate	เป็นอัตราส่วนระหว่างจำนวนครั้งที่พบข้อมูลในแคชต่อจำนวนครั้งที่มีการร้องขอข้อมูลทั้งหมด
Byte Hit Rate	เป็นอัตราส่วนระหว่างขนาดของข้อมูลที่พบในแคชต่อขนาดของข้อมูลที่มีการร้องขอทั้งหมด
ค่าใช้จ่าย (Cost)	เป็นการวัดค่าใช้จ่ายที่เกิดขึ้นจากการถ่ายโอนข้อมูลของแคช
ค่าความหน่วงโดยเฉลี่ย (Average Latency)	เป็นการวัดเวลาโดยเฉลี่ยในการเข้าถึงข้อมูลที่ถูกร้องขอโดยนับตั้งแต่เวลาที่ร้องขอจนกระทั่งได้รับข้อมูลครบถ้วน
ผลกระทบจากค่าบริการที่เปลี่ยนไป	เป็นการเปรียบเทียบประสิทธิภาพของอัลกอริธึมการวางทับข้อมูลแบบต่างๆ ภายใต้เงื่อนไขที่กำหนดให้มีการเปลี่ยนแปลงอัตราค่าบริการในส่วนของการอัพโหลดข้อมูล
ค่าบริการเมื่อใช้งานผู้ให้บริการคลาวด์หลายแห่ง	เป็นการเปรียบเทียบประสิทธิภาพของอัลกอริธึมการวางทับข้อมูลแบบต่างๆ ภายใต้เงื่อนไขที่กำหนดให้มีการเก็บข้อมูลต่างๆไว้ที่ผู้ให้บริการคลาวด์มากกว่า 1 แห่ง

การประเมินประสิทธิภาพต่างๆตามตารางที่ 4-3 จะมีการกำหนดรายละเอียดในการทดลองให้เป็นไปตามตารางที่ 4-2 ยกเว้นการวัดค่าผลกระทบจากค่าบริการที่เปลี่ยนไป และค่าบริการเมื่อใช้งานผู้ให้บริการคลาวด์หลายแห่ง ที่จะกำหนดให้แคชมีขนาดร้อยละ 20 ของจำนวนข้อมูลที่ถูกร้องขอโดยไม่ซ้ำกัน และกำหนดบล็อกข้อมูลมีขนาด 4,096 ไบต์

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

4.4 แบบจำลองอัตราค่าบริการ

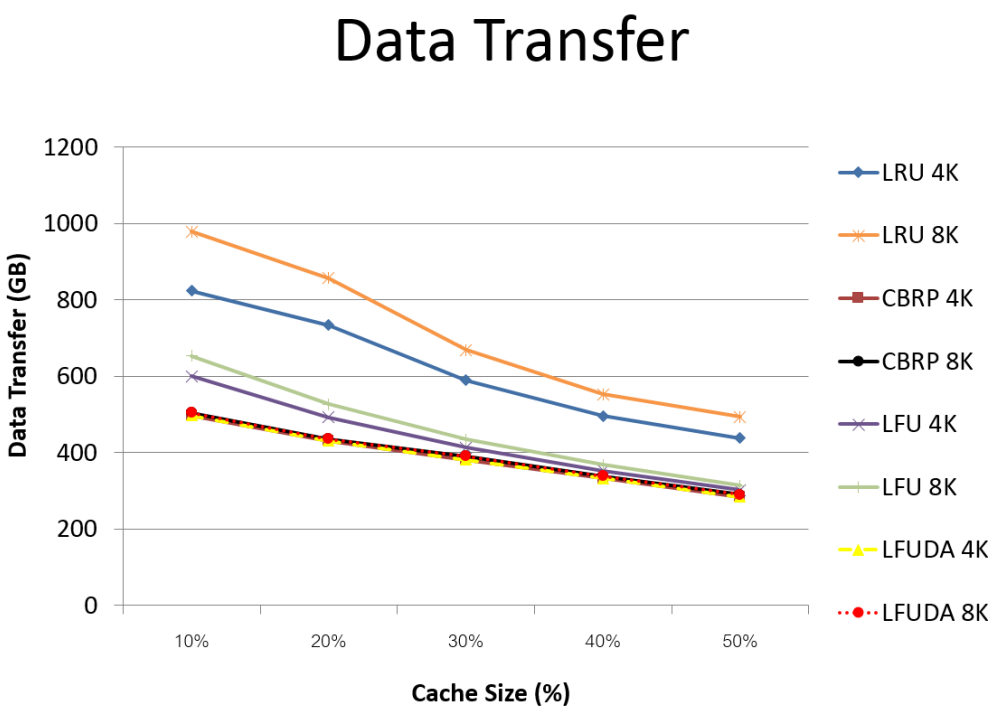
ในงานวิจัยนี้เราจะใช้อัตราค่าบริการถ่ายโอนข้อมูลจาก IBM SmartCloud Enterprise [15] ซึ่งมีข้อมูลโดยสรุปดังตารางที่ 4-4

ตารางที่ 4-4 อัตราค่าบริการของ IBM SmartCloud Enterprise

ชนิดของบริการ	อัตราค่าบริการต่อ GB
Upload traffic	0.15 เหรียญ
Download traffic	0.15 เหรียญ

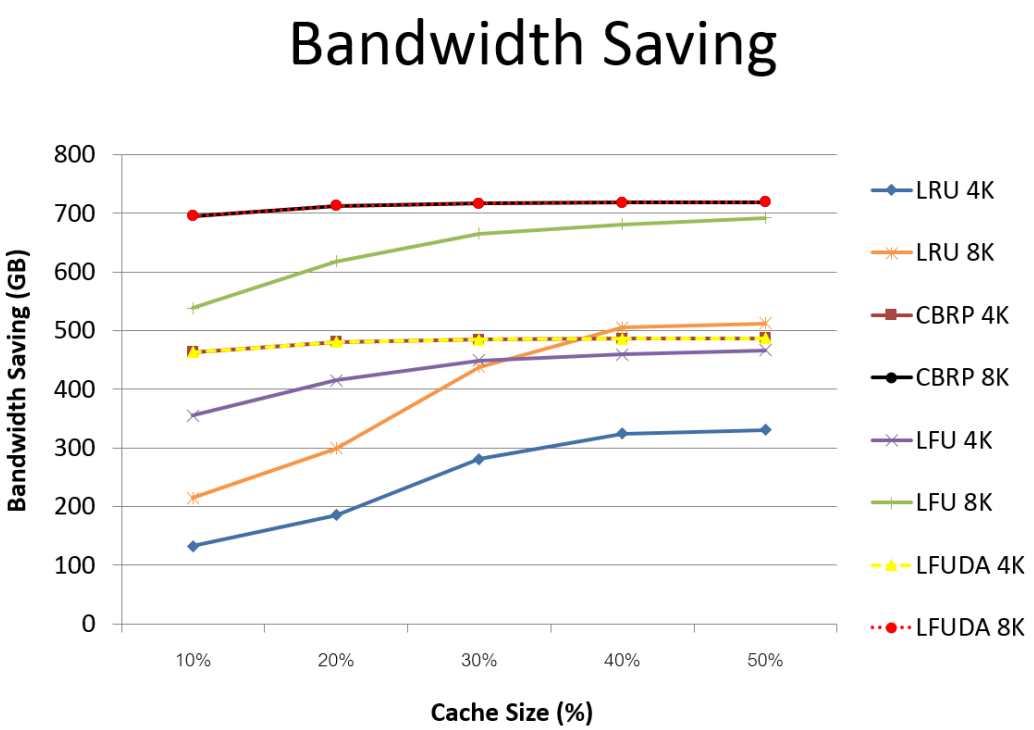
4.5 ผลการทดลอง

ในหัวข้อนี้จะนำเสนอผลการเปรียบเทียบประสิทธิภาพด้านต่างๆของอัลกอริธึมการวางทับข้อมูลระหว่างซีบีอาร์พี แอลอาร์ยู แอลเอฟยู และแอลเอฟยูดีเอ โดยในส่วนของแอลเอฟยูดีเอนั้นจะถูกนำมาใช้เป็นเกณฑ์ประสิทธิภาพมาตรฐานที่ระบบสามารถทำงานได้ดีที่สุด ภายใต้อัลกอริธึมที่ใช้ความถี่เป็นตัวเปรียบเทียบ ซึ่งเราจะกล่าวสรุปถึงแอลเอฟยูดีเออีกครั้งในตอนท้ายของหัวข้อนี้



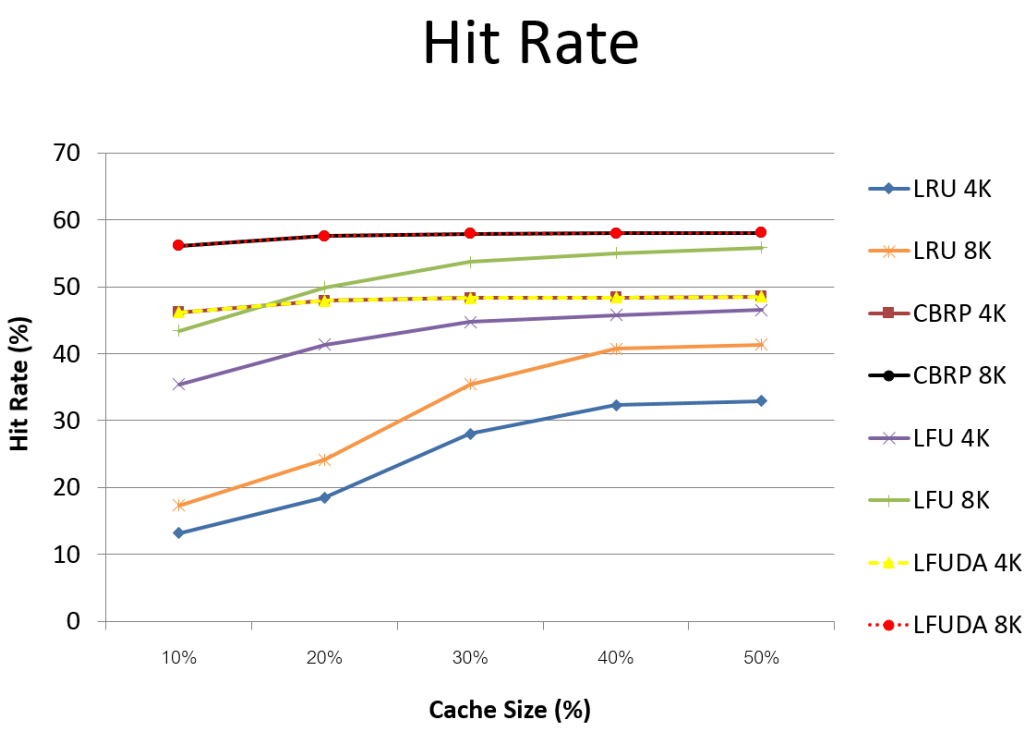
ภาพที่ 4-1 ปริมาณการถ่ายโอนข้อมูล

จากภาพที่ 4-1 จะเห็นว่าปริมาณการถ่ายโอนข้อมูลของซีบีอาร์พี จะมีค่าน้อยกว่าแอลอาร์ยู และแอลเอฟยู ไม่ว่าแคชและบล็อกข้อมูลจะมีขนาดเท่าไหร่ก็ตาม ทั้งการเปรียบเทียบในกรณีที่บล็อกข้อมูลมีขนาดเท่ากันระหว่างอัลกอริธึมทั้งสาม และการเปรียบเทียบแบบบล็อกข้อมูลมีขนาดไม่เท่ากัน นอกจากนี้ผลการทดลองยังบ่งบอกอีกว่าแอลอาร์ยูไม่เหมาะกับแคชที่มีเงื่อนไขในการทำงานเช่นนี้

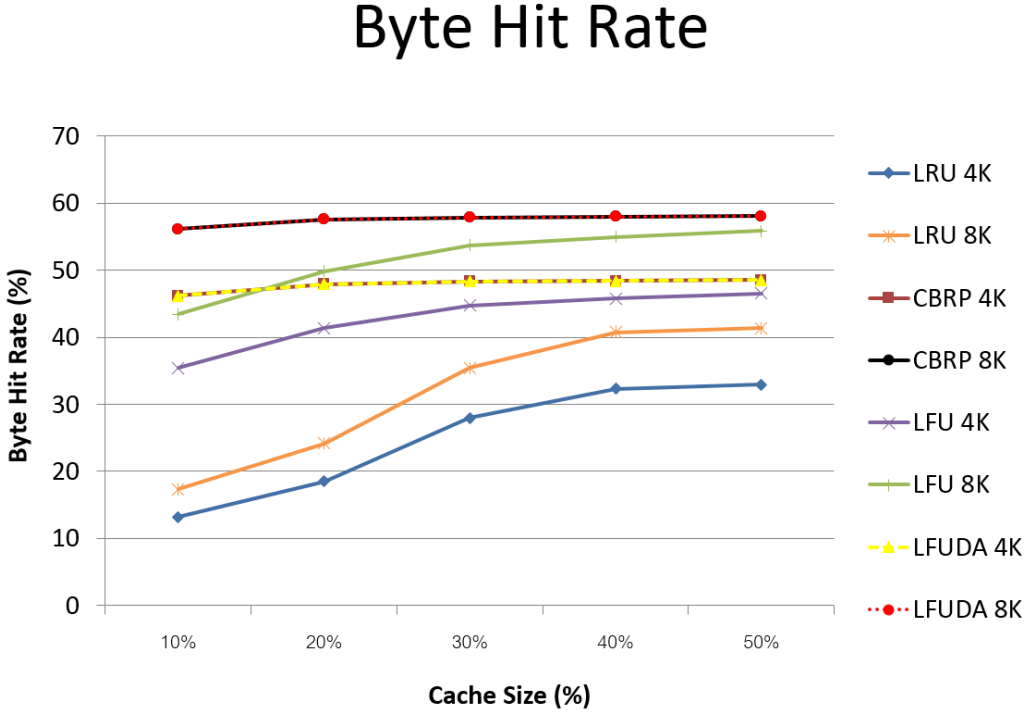


ภาพที่ 4-2 ปริมาณแบนด์วิดธ์ที่ลดลง

จากภาพที่ 4-2 เมื่อเปรียบเทียบประสิทธิภาพทางด้านปริมาณแบนด์วิดธ์ที่ลดลงระหว่างบล็อกข้อมูลที่มีขนาด 4 กิโลไบต์กับ 8 กิโลไบต์นั้นจะพบว่ายิ่งบล็อกข้อมูลมีขนาดใหญ่ขึ้นเท่าไหร่ ปริมาณแบนด์วิดธ์ที่ลดลงก็จะมากขึ้นเท่านั้น แต่ทั้งนี้บล็อกข้อมูลทั้ง 2 ขนาดจะมีแนวโน้มของประสิทธิภาพที่เป็นไปในทิศทางเดียวกันทั้ง 3 อัลกอริธึม เมื่อแคชมีขนาดที่เปลี่ยนไป ซึ่งจากการสรุปผลแล้วจะพบว่าซีบีอาร์พีมีประสิทธิภาพที่ดีกว่าอัลกอริธึมที่เหลือ นอกจากนี้ยังมีแนวโน้มของประสิทธิภาพที่เริ่มคงที่ตั้งแต่แคชมีขนาดที่ 20% ขึ้น ซึ่งเป็นข้อบ่งชี้ว่าซีบีอาร์พีสามารถทำงานได้อย่างมีประสิทธิภาพในแคชที่มีขนาดไม่ใหญ่มากนัก

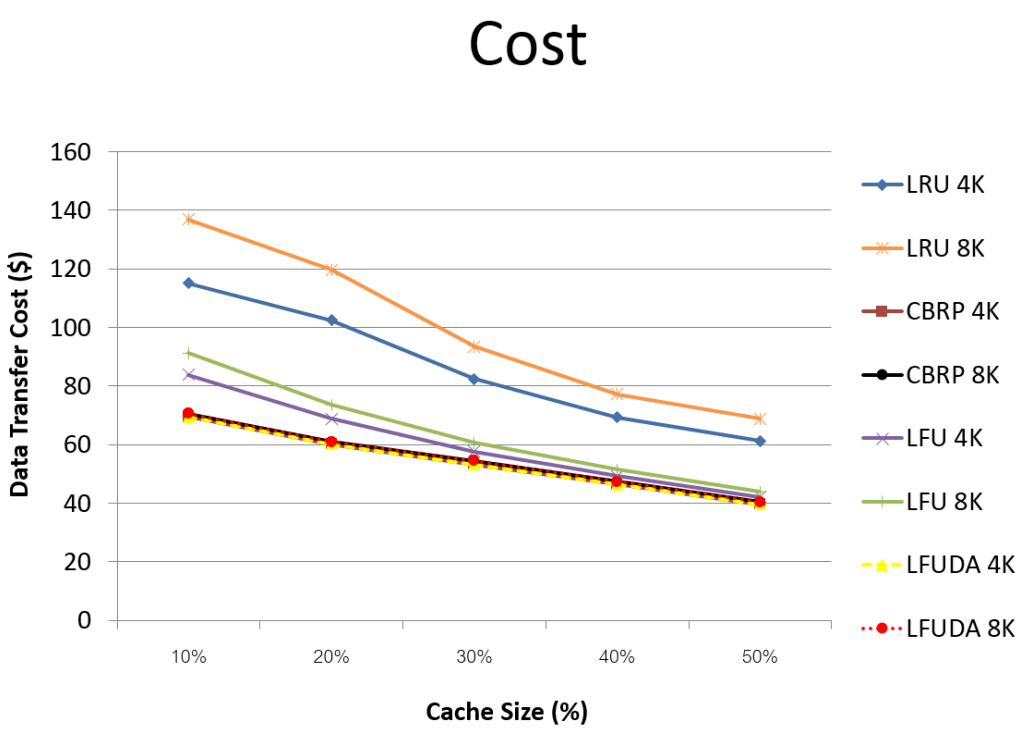


ภาพที่ 4-3 Hit Rate



ภาพที่ 4-4 Byte Hit Rate

จากภาพที่ 4-3 และ 4-4 จะสังเกตเห็นว่ากราฟผลการทดลองที่ออกมามีลักษณะเหมือนกัน ซึ่งเป็นผลมาจากการแบ่งข้อมูลออกเป็นบล็อกนั่นเอง ทำให้การพบข้อมูลในแคชแต่ละครั้งจะพบข้อมูลที่มีขนาดเท่ากันทุกครั้งนั่นเอง ซึ่งผลการทดลองที่ออกมาจะพบว่าซีบีอาร์พี ยังคงมีประสิทธิภาพที่เหนือกว่าทั้งแอลเอฟยู และแอลอาร์ยู ไม่ว่าแคชและบล็อกข้อมูลจะมีขนาดเท่าไหร่ก็ตาม และจะสังเกตเห็นได้ว่าประสิทธิภาพจะเริ่มคงที่เมื่อแคชมีขนาด 20% ขึ้นไป ดังเช่นภาพที่ 4-2 ในขณะที่แอลเอฟยู และแอลอาร์ยูกลับต่อรอจนกว่าแคชจะมีขนาด 40% ขึ้นไป จึงจะสามารถทำงานได้อย่างเต็มประสิทธิภาพ

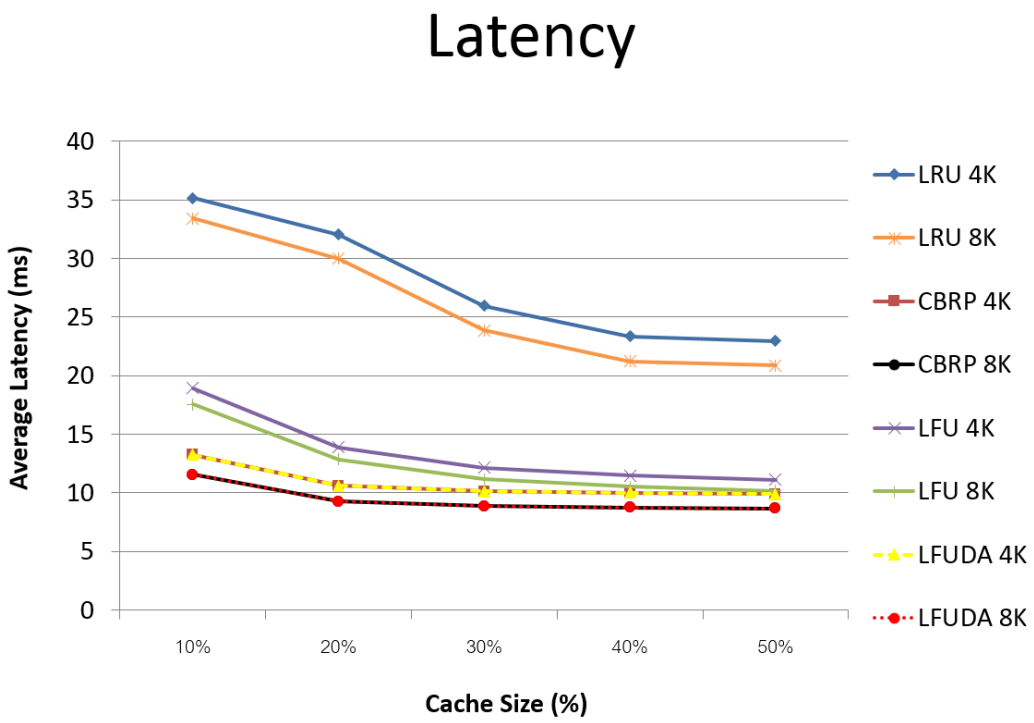


ภาพที่ 4-5 ค่าใช้จ่าย

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ในภาพที่ 4-5 จะเป็นการทดสอบประสิทธิภาพในเรื่องของค่าใช้จ่าย ซึ่งหากสังเกตดีๆจะพบว่าแนวโน้มของกราฟมีความคล้ายคลึงกับภาพที่ 4-1 เนื่องมาจากค่าใช้จ่ายที่เกิดขึ้ยภายในงานวิจัยนี้จะมุ่งเน้นไปที่เรื่องของค่าบริการที่เกิดขึ้นจากการถ่ายโอนข้อมูลเท่านั้น ซึ่งเป็นจุดประสงค์หลักของงานวิจัยชิ้นนี้ จึงส่งให้ปริมาณการถ่ายโอนข้อมูลมีผลโดยตรงกับค่าใช้จ่ายที่เกิดขึ้น ซึ่งจากการสรุปผลในส่วนนี้จะพบว่าอัลกอริธึมซีบีอาร์พีสามารถช่วยลดค่าใช้จ่ายที่เกิดขึ้นได้มากที่สุด ตามมาด้วยแอลเอฟยู และแอลอาร์ยู โดยเมื่อพิจารณาจากกราฟแล้วจะพบว่าซีบีอาร์พี และแอลเอฟยูมีประสิทธิภาพที่ใกล้เคียง ซึ่งเป็นผลมาจากการที่สมการกำหนดมูลค่าของซีบีอาร์พีมีค่าความถี่เข้ามาเป็น 1 ในตัวแปรหลัก อีกทั้งยังเป็นการทดลองที่ใช้งานพื้นที่เก็บข้อมูลบนคลาวด์จากผู้ให้บริการเพียงแห่งเดียว จึงทำให้ค่าของ C_i ในสมการที่ 3-1 ไม่ส่งผลกระทบต่อมูลค่าของข้อมูลมากนัก นอกจากนี้บล็อกข้อมูล

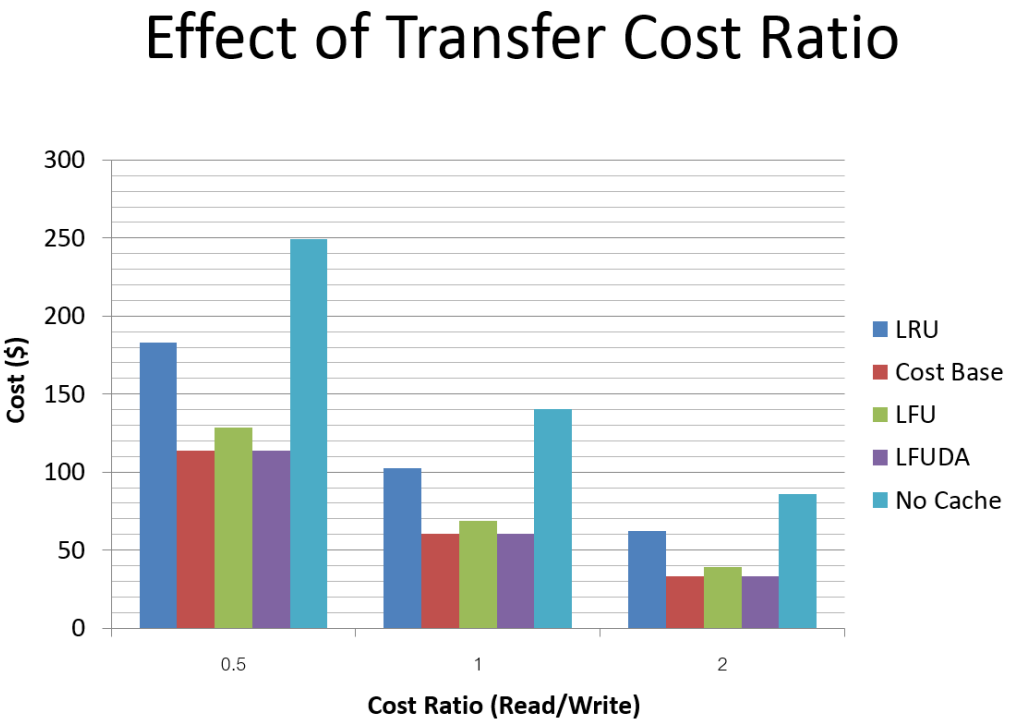
ยังมีขนาดเท่ากันอีกด้วยจึงทำให้ค่า S_i ซึ่งเป็นอีก 1 ตัวแปรหลักมีผลกระทบต่อสมการน้อยกว่าค่าความถี่นั่นเอง



ภาพที่ 4-6 ค่าความหน่วงโดยเฉลี่ย

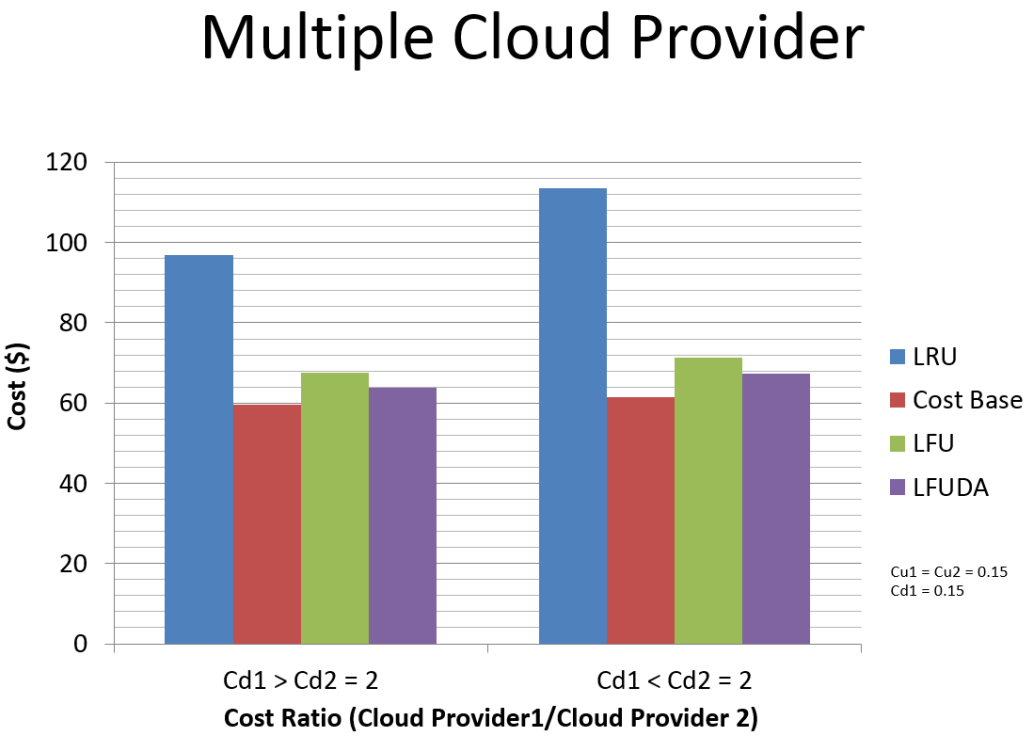
ทางด้านค่าความหน่วงโดยเฉลี่ยที่แสดงอยู่ในภาพ 4-6 นั้น เราจะพบว่าทั้งซีบีอาร์พี และแอลเอฟยู มีประสิทธิภาพที่ดีกว่าแอลอาร์ยูมาก ทั้งนี้หากพิจารณาควบคู่ไปกับภาพที่ 4-3 และ 4-4 จะพบว่าค่า Hit Rate ส่งผลโดยตรงกับประสิทธิภาพในส่วนนี้ ซึ่งเป็นไปตามสมการในการคำนวณหาค่าความหน่วงโดยเฉลี่ย

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY



ภาพที่ 4-7 ผลกระทบที่เกิดจากอัตราส่วนค่าใช้จ่ายของการถ่ายโอนข้อมูล

จากภาพที่ 4-7 จะเป็นการทดสอบประสิทธิภาพของอัลกอริธึมต่างๆ เมื่อต้องทำงานภายใต้อัตราค่าบริการที่ไม่กัน โดยจะกำหนดให้ค่าบริการในส่วนของการอัพโหลดมีค่าเป็น 0.075 0.15 และ 0.3 เหรียญต่อกิกกะไบต์ตามลำดับ และคงค่าบริการในส่วนของการดาวน์โหลดเอาไว้ที่ 0.15 เหรียญต่อกิกกะไบต์ทั้ง 3 กรณี โดยผลที่ออกมาแสดงให้เห็นว่าปริมาณการอัพโหลดข้อมูลส่งผลกระทบต่อค่าบริการสูงกว่าที่เราคาดการณ์ไว้ ซึ่งจะเห็นได้จากกราฟที่เมื่อค่าบริการในส่วนของการอัพโหลดลดลง แนวโน้มของค่าบริการจากการถ่ายโอนข้อมูลโดยรวมก็จะลดลงตามไปด้วย และนั่นยังทำให้เราได้เห็นถึงประโยชน์จากการที่ในปัจจุบันผู้ให้บริการคลาวด์หลายๆแห่งได้ยกเลิกการเก็บค่าบริการในส่วนนี้ไปแล้ว



ภาพที่ 4-8 ค่าบริการเมื่อใช้งานผู้ให้บริการคลาวด์หลายแห่ง

การวัดประสิทธิภาพในส่วนสุดท้ายจะเป็นการเปรียบเทียบค่าใช้จ่ายระหว่างแคชที่ใช้งานอัลกอลิธึมแอลอาร์ยู แอลเอฟยู และซีบีอาร์พี ภายใต้เงื่อนไขที่กำหนดให้มีการใช้บริการพื้นที่เก็บข้อมูลบนคลาวด์จากผู้ให้บริการมากกว่า 1 แห่ง ซึ่งจากภาพที่ 4-8 จะเห็นว่าเมื่อค่าบริการในส่วนของการดาวน์โหลดจากผู้ให้บริการหมายเลข 1 แพงกว่าผู้ให้บริการหมายเลข 2 สองเท่า ค่าบริการจากการถ่ายโอนข้อมูลของซีบีอาร์พีจะอยู่ที่ 59.5 $ ในขณะที่แอลอาร์ยูจะมีมูลค่าสูงถึง 96.91 $ และ LFU จะอยู่ที่ 67.49 $ และเมื่อค่าบริการในส่วนของการดาวน์โหลดจากผู้ให้บริการหมายเลข 1 ถูกกว่าผู้ให้บริการหมายเลข 2 สองเท่า ค่าบริการจากการถ่ายโอนข้อมูลของซีบีอาร์พี จะอยู่ที่ 61.5 $ ในขณะที่ LRU จะมีมูลค่า 113.51 $ และ LFU จะอยู่ที่ 71.3 $ ซึ่งจากทั้ง 2 กรณีจะพบว่า CBRP ทำให้เกิดค่าบริการที่น้อยที่สุด นอกจากนี้ยังมีส่วนต่างของค่าใช้บริการจากทั้ง 2 กรณีอยู่ที่ 2 $ เท่านั้นหรือคิดเป็น 3.36% ของค่าบริการเดิม ในขณะที่ LRU กลับมีส่วนต่างที่มากถึง 16.69 $ หรือคิดเป็น 17.13% ของค่าบริการเดิม และสำหรับ LFU นั้นส่วนต่างของราคาจะอยู่ที่ 3.81 $ หรือคิดเป็น 5.65% ของค่าบริการเดิม ซึ่งเราสามารถสรุปได้ว่าซีบีอาร์พี จัดการกับเรื่องค่าใช้จ่ายได้มีประสิทธิภาพมากที่สุด แม้ว่าจะมีการเปลี่ยนแปลงอัตราค่าบริการก็ตาม ทั้งนี้ในการทดลองเราได้กำหนดอัตราค่าบริการในการอัพโหลดของผู้ให้บริการทั้ง 2 แห่งเป็น 0.15 $/GB และได้กำหนดค่าบริการในส่วนของการดาวน์โหลดจากผู้ให้บริการหมายเลข 1 เป็น 0.15 $/GB ในส่วนการเลือก

พื้นที่เก็บข้อมูลจากผู้ให้บริการทั้งสองแก่ข้อมูลต่างๆ เราจะใช้วิธีนำ File ID มาหาค่าเศษเหลือ (Modulus operator) เพื่อดูว่าจะเก็บข้อมูลนั้นๆไว้ที่ใคร

สำหรับแอลเอฟยูดีเอที่ถูกนำมาใช้เป็นเกณฑ์ประสิทธิภาพมาตรฐานที่ระบบจะสามารถทำงานได้ดีที่สุดนั้น หากระบบทำงานภายใต้เงื่อนไขของการใช้ผู้ให้บริการคลาวด์เพียงแห่งเดียว จะสังเกตเห็นว่าซีบีอาร์พี มีประสิทธิภาพเทียบเท่ากับแอลเอฟยูดีเอ เนื่องจากเงื่อนไขที่เรากำหนดให้มีการแบ่งข้อมูลออกเป็นบล็อก จึงทำให้ตัวแปร S_i ที่เป็นขนาดของข้อมูลในสมการกำหนดมูลค่า มีค่าเท่ากันทุกบล็อกข้อมูล รวมไปถึงตัวแปร C_i ที่เป็นค่าใช้จ่ายในการถ่ายโอนข้อมูล ก็จะมีค่าเท่ากันด้วย เนื่องจากเป็นการเรียกใช้งานข้อมูลจากผู้ให้บริการแห่งเดียวกัน จึงเกิดค่าใช้จ่ายที่เท่ากัน ทำให้เราสามารถตัดตัวแปรทั้ง 2 ออกไปจากสมการได้ เนื่องจากในสภาพแวดล้อมเช่นนี้ตัวแปรทั้ง 2 จะมีลักษณะเป็นค่าคงที่ ซึ่งจะส่งผลให้ซีบีอาร์พีมีรูปแบบสมการเหมือนแอลเอฟยูดีเอ แต่หากสภาพแวดล้อมในการทำงานเปลี่ยนไปดังเช่นภาพที่ 4-8 ที่มีการใช้งานผู้ให้บริการคลาวด์หลายแห่งจะพบว่าซีบีอาร์พีมีประสิทธิภาพที่สูงกว่าในแง่ของการลดค่าใช้จ่าย เนื่องจากมีตัวแปร C_i เข้ามาเป็นหนึ่งในตัวแปรที่ใช้ประเมินประสิทธิภาพ ซึ่งแตกต่างจากแอลเอฟยูดีเอที่จะใช้เพียงค่าความถี่เท่านั้น

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 5
สรุปผลการวิจัย

5.1 บทสรุป

วิทยานิพนธ์ฉบับนี้ได้นำเสนอวิธีการลดค่าใช้จ่ายของพื้นที่เก็บข้อมูลบนคลาวด์ด้วยคลาวด์แคชชิ่งแบบท้องถิ่น โดยมุ่งเน้นไปที่การออกแบบอัลกอริธึมในการวางทับข้อมูลเพื่อลดปริมาณการถ่ายโอนข้อมูลอันเป็นอีกหนึ่งปัจจัยสำคัญที่อาจทำให้ค่าใช้จ่ายมีมูลค่าสูง งานวิจัยนี้ได้ทำการประเมินประสิทธิภาพของอัลกอริธึมในการวางทับข้อมูลด้วยวิธีการจำลองการทำงานของแคชโดยใช้ข้อมูลภาระงานที่เก็บมาจากเครื่องแม่ข่ายเอ็นเอฟเอส (NFS Server) ของคณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ซึ่งเป็นประวัติการเข้าใช้งานข้อมูลจริงจากผู้ใช้ต่างๆภายในคณะ จากการเปรียบเทียบประสิทธิภาพของอัลกอริธึมในการวางทับข้อมูลที่ออกแบบกับอัลกอริธึมอื่นๆที่นิยมนำไปใช้งานกันภายในระบบต่างๆพบว่า อัลกอริธึมการวางทับข้อมูลโดยการเปรียบเทียบมูลค่าที่ออกแบบขึ้นมาใหม่ สามารถทำงานได้มีประสิทธิภาพมากกว่าอัลกอริธึมที่นำมาเปรียบเทียบ ทั้งในแง่ของการลดค่าใช้จ่ายที่เป็นจุดประสงค์หลักของงานวิจัยนี้ และในแง่ประสิทธิภาพ ทำให้ระบบโดยรวมสามารถทำงานได้อย่างรวดเร็วขึ้น

ทั้งนี้ภาระงานที่ถูกนำมาใช้ภายในงานวิจัยนี้เป็นภาระงานที่เกิดขึ้นจากระบบที่มีแคชอยู่ในส่วนต่างๆ เช่น แคชในเครื่องแม่ข่าย และแคชในเครื่องลูกข่าย ทำให้ลักษณะการร้องขอหรือเขียนข้อมูลจากแคชของเครื่องแม่ข่ายเอ็นเอฟเอสไม่มีลักษณะของ Recency-Based ตามที่งานวิจัย [23] ได้นำเสนอ ซึ่งเป็นข้อพิสูจน์ที่ชัดเจนว่าภายใต้สภาวะแวดล้อมของพื้นที่เก็บข้อมูลบนเครือข่าย ไม่ว่าจะเป็นเครือข่ายแบบท้องถิ่น หรือเครือข่ายบริเวณกว้างอย่างคลาวด์ แอลอาร์ยูอาจจะไม่ใช่คำตอบที่ดีนักสำหรับระบบประเภทนี้ แต่กลับเป็นตัวแปรในเรื่องของความถี่ที่จะสามารถช่วยให้แคชทำงานได้อย่างมีประสิทธิภาพมากที่สุด

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

5.2 ข้อเสนอแนะ

งานวิจัยชิ้นนี้มีข้อจำกัดบางประการในขั้นตอนการทดลอง ซึ่งแม้ว่าจะมีการออกแบบสมการในการกำหนดมูลค่าของข้อมูลให้มีความซับซ้อนน้อยที่สุดแล้ว แต่เมื่อนำไปทดลองจริงกลับพบว่าข้อมูลที่จำเป็นต่อการประมวลผลยังคงมีปริมาณที่เยอะพอสมควร ส่งผลให้ต้องใช้เวลาและทรัพยากรในการจำลองการทำงานของแคชมากกว่าอัลกอริธึมอย่างแอลอาร์ยู จึงควรมีการพัฒนาต่อยอดอัลกอริธึมให้มีความซับซ้อนที่น้อยลงกว่านี้เพื่อลดภาระให้แก่ระบบ โดยแนวทางที่เป็นไปได้คือการตัดฟังก์ชั่นเขียนข้อมูลลงแคชออกไป เนื่องจากในปัจจุบันผู้ให้บริการคลาวด์หลายๆแห่งได้ยกเลิกการเก็บค่าบริการในการอัพโหลดข้อมูลขึ้นสู่คลาวด์แล้ว ทำให้สามารถลดภาระในการทำงานของระบบลง อีกทั้งยังเป็นการเพิ่มพื้นที่สำหรับการจัดเก็บข้อมูลในแคชอีกด้วย นอกจากนี้ในงานวิจัยชิ้นนี้ได้ทำการทดลองภายใต้เงื่อนไขของการบริหารจัดการแคชแบบลำดับชั้นเดียว ซึ่งมีข้อจำกัดในเรื่องของการใช้

งานพื้นที่เก็บข้อมูลบนคลาวด์จากหลายสถานที่พร้อมกัน เพราะจะทำให้เกิดปัญหาในเรื่องของข้อมูลที่ไม่ตรงกันระหว่างผู้ใช้ต่างๆ งานวิจัยนี้จึงสามารถพัฒนาเพิ่มเติมได้โดยการการเปลี่ยนแปลงแคชไปเป็นรูปแบบอื่นๆ ซึ่งนอกจากจะแก้ปัญหาที่เกิดขึ้นได้แล้ว ยังน่าจะสามารถเพิ่มประสิทธิภาพให้แก่ระบบได้มากขึ้นอีกด้วย

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

รายการอ้างอิง

1. Armbrust, M., et al., *A view of cloud computing.* Commun. ACM, 2010. 53(4): p. 50-58.
2. Marston, S., et al., *Cloud computing - The business perspective.* Decis. Support Syst., 2011. 51(1): p. 176-189.
3. Foster, I., et al. *Cloud Computing and Grid Computing 360-Degree Compared*. in *Grid Computing Environments Workshop,* 2008. *GCE '*08. 2008.
4. Apple Inc. *About iCloud*. [cited 2014, March 21]; Available from: http://www.apple.com/icloud.
5. Dropbox Inc. *About Dropbox*. [cited 2014, March 21]; Available from: https://www.dropbox.com/about
6. ADrive LLC. *About ADrive*. [cited 2014, March 21]; Available from: http://www.adrive.com/about
7. Amazon.com Inc. *AWS Case Study: Ericsson* [cited 2014, March 21]; Available from: http://aws.amazon.com/solutions/case-studies/hitachi-systems
8. Amazon.com Inc. *AWS Case Study: Hitachi*. [cited 2012, May 13]; Available from: http://aws.amazon.com/solutions/case-studies/hitachi-systems
9. Amazon.com Inc. *AWS Case Study: SmugMug's Cloud Migration*. [cited 2014, March 21]; Available from: http://aws.amazon.com/solutions/case-studies/smugmug.
10. Amazon.com Inc. *Customer Experiences: University of Texas at Austin*. [cited 2014, March 21]; Available from: http://aws.amazon.com/education/customer-experiences/#4.
11. Amazon.com Inc. *Customer Experiences: Stanford University* [cited 2014, March 21]; Available from: http://aws.amazon.com/education/customer-experiences/#9
12. Siriruangampai, S. and N. Nupairoj. *Lower Storage TCO with Local Cloud Caching*. in *the* 2011 *Thai Grid and Cloud Conference (TGCC '*11). 2011.
13. Siriruangampai, S. and N. Nupairoj. *Cost-Based Replacement Policy for Cloud Storage*. in *The* 10*th National Conference on Computing and Information Technology (NCCIT* 2014). 2014.
14. National Institute of Standards and Technology. *NIST Definition of Cloud Computing*. [cited 2014, March 21]; Available from: http://www.nist.gov/itl/cloud/index.cfm
15. International Business Machines Corporation. *IBM SmartCloud Enterprise Charges Schedule*. [cited 2013, December 23]; Available from: http://www-935.ibm.com/services/us/en/cloud-enterprise/contracts/charges_schedule.html.
16. เจษฎา เพ็งสุวรรณ, การประเมินประสิทธิภาพวิธีการแทนที่ข้อมูลในแคชแบบแบ่งบล็อกบนระบบดาตากริด. 2554, ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย.

17. Podlipnig, S., et al., *A survey of Web cache replacement strategies.* ACM Comput. Surv., 2003. 35(4): p. 374-398.
18. Wong, K.Y., *Web cache replacement policies: a pragmatic approach.* Network, IEEE, 2006. 20(1): p. 28-34.
19. Zadnik, M. and M. Canini. *Evaluation and design of cache replacement policies under flooding attacks*. in *Wireless Communications and Mobile Computing Conference (IWCMC),* 2011 7*th International*. 2011.
20. Jouppi, N.P., *Cache write policies and performance.* SIGARCH Comput. Archit. News, 1993. 21(2): p. 191-201.
21. Misra, S.C. and A. Mondal, *Identification of a company's suitability for the adoption of cloud computing and modelling its corresponding Return on Investment.* Mathematical and Computer Modelling, 2011. 53(3-4): p. 504-521.
22. Palankar, M.R., et al., *Amazon S3 for science grids: a viable solution?*, in *Proceedings of the* 2008 *international workshop on Data-aware distributed computing*. 2008, ACM: Boston, MA, USA. p. 55-64.
23. Nagasako, Y. and S. Yamaguchi. *A Server Cache Size Aware Cache Replacement Algorithm for Block Level Network Storage*. in *Autonomous Decentralized Systems (ISADS),* 2011 10*th International Symposium on*. 2011.
24. Willick, D.L., D.L. Eager, and R.B. Bunt. *Disk cache replacement policies for network fileservers*. in *Distributed Computing Systems,* 1993*., Proceedings the* 13*th International Conference on*. 1993.
25. Banditwattanawong, T., *From web cache to cloud cache*, in *Proceedings of the 7th international conference on Advances in Grid and Pervasive Computing*. 2012, Springer-Verlag: Hong Kong, China. p. 1-15.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ภาคผนวก

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ประวัติผู้เขียนวิทยานิพนธ์

สุเมธ สิริเรืองอำไพ เกิดวันที่ 4 กรกฎาคม พ.ศ. 2529 ที่จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาระดับปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ จากคณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ ในปีการศึกษา 2552 และเข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2553 งานวิจัยที่สนใจได้แก่ คลาวด์คอมพิวติง และดาตา แคชชิง

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University