การจัดสรรชิ้นส่วนข้อมูลบันทึกสารสนเทศในมองโกดีบี

นายบุญญฤทธิ์ อุณหะวัฒน์

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

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.

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2559
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

Sharding Allocation of IT Log Data in MongoDB

Mr. Boonyalit Oonhawat

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Computer Science
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2016

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การจัดสรรชิ้นส่วนข้อมูลบันทึกสารสนเทศในมองโกดีบี
โดย	นายบุญญฤทธิ์ อุณหะวัฒน์
สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ผู้ช่วยศาสตราจารย์ ดร. ณัฐวุฒิ หนูไพโรจน์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

.....คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร. สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร. วีระ เหมืองสิน)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(ผู้ช่วยศาสตราจารย์ ดร. ณัฐวุฒิ หนูไพโรจน์)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร. เกริก ภิรมย์โสภา)

.....กรรมการภายนอกมหาวิทยาลัย
(ดร. กาญจนา ศิลาวราเวทย์)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บุญญฤทธิ์ อุณหะวัฒน์ : การจัดสรรชิ้นส่วนข้อมูลบันทึกสารสนเทศในมองโกดีบี (Sharding Allocation of IT Log Data in MongoDB) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: ผศ. ดร. ณัฐวุฒิ หนูไพโรจน์, 47 หน้า.

ข้อมูลบันทึกสารสนเทศมีลักษณะที่ส่งผลกระทบต่อประสิทธิภาพการทำงานหากได้รับการพัฒนาที่ไม่เหมาะสมกับลักษณะข้อมูล จากการศึกษาพบว่าลักษณะแนวทางการกระจายตัวของชิ้นส่วนข้อมูลส่งผลถึงประสิทธิภาพการทำงานโดยรวมของระบบทั้งจากการจัดเก็บข้อมูลและการสืบค้น ซึ่งวิธีการกระจายข้อมูลที่มีมาแต่เดิมในระบบอันได้แก่การกระจายแบบพิสัยและแฮช(hash)นั้นมีข้อดีข้อเสียแตกต่างกันอย่างชัดเจน โดยเฉพาะเมื่อระบบอยู่ในสภาวะที่ต้องรองรับงานเกินขีดจำกัด เป้าหมายงานวิจัยนี้จะศึกษาถึงแนวทางการกระจายของข้อมูลและจัดสรรชิ้นส่วนข้อมูลให้อยู่ในลักษณะที่สามารถนำเอาทรัพยากรของระบบมาใช้ได้อย่างเต็มประสิทธิภาพและลดผลกระทบของภาระการทำงานของเครื่องในระบบ นี้ได้นำความสามารถของระบบการติดป้ายข้อมูลมาประยุกต์ใช้กับแผนการกระจายข้อมูลพื้นฐาน โดยในการทดลองเราได้จำลองระบบฐานข้อมูลแบบกระจายด้วยมองโกดีบี ขึ้นมาเพื่อใช้ในการวิจัยเปรียบเทียบประสิทธิภาพการทำงานโดยข้อมูลบันทึกสารสนเทศที่ถูกจำลองขึ้นมาใช้ในงานวิจัยนี้ได้ทำการอ้างอิงโครงสร้างข้อมูลจากระบบไอทีของมหาวิทยาลัยจุฬาลงกรณ์ และการทดลองจะวัดประสิทธิภาพจากการทำงานหลักๆสองประเภทได้แก่ การจัดเก็บข้อมูล และ การสืบค้นข้อมูล ภายใต้เงื่อนไขที่สภาพการทำงานของระบบอยู่ในสภาวะปกติและทำงานหนัก ซึ่งประสิทธิภาพการทำงานจะถูกประเมิณจากปริมาณงานที่ทำเสร็จและเวลาที่ใช้ในการประมวลผล

งานวิจัยนี้ได้นำเสนอกระบวนการกระจายข้อมูลแบบใหม่ที่เหมาะสมกับข้อมูลบันทึกสารสนเทศ โดยการประยุกต์การกระจายข้อมูลแบบติดป้ายข้อมูล ร่วมกับการคาดการณ์ความเร็วของข้อมูลขาเข้า ซึ่งทำให้ระบบสามารถจัดเก็บข้อมูลข้อมูลได้เร็วกว่าวิธีแบบพิสัยและแบบแฮช ในขณะที่ยังสามารถคงไว้ซึ่งประสิทธิภาพที่ดีในส่วนการสืบค้นข้อมูลไม่ว่าจะเป็นในภาวะที่ไม่มีการบันทึกข้อมูลจนถึงในภาวะที่มีการบันทึกข้อมูลจำนวนมาก

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต .....

สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์ ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

ปีการศึกษา 2559

5670264821 : MAJOR COMPUTER SCIENCE

KEYWORDS: DISTRIBUTION SYSTEM / NOSQL / BIG DATA

BOONYALIT OONHAWAT: Sharding Allocation of IT Log Data in MongoDB. ADVISOR: ASST. PROF. NATAWUT NUPAIROJ, Ph.D., 47 pp.

Log Data characteristics are fast growing and have large amount of data which end up affect system performance. Due to this reason, there are high demand of the distributed system that can answer the performance problems. According to the studied, data distribution strategy greatly affects overall system performance including both insert and query operation. Range and Hash data distribution strategy have different kind of advantage and disadvantage that will affect entire system especially during the time when system suffer from hotspot. The goal of this research is to study the data distribution strategy and data allocation and how to archive highest efficiency from using system resource and reduce the affect from performance problems. In this research, we take advantage of the tag aware function by incorporate function with range data distribution strategy. For the experiment, we simulate distribution database system using MongoDB for the purpose of compare the performance between each data distribution strategy. log data use in this experiment are based on the log data from Chulalongkorn University IT log system. There're two main experiment which are insert data and data query based experiment. The performance evaluation will be measure by the insert throughput and the completion time.

This research proposed a new data distribution strategy suitable for storing log data. Using tag distribution and incoming data velocity forecasting, our proposed strategy performed much faster than Range and Hash distribution for inserting operations. The strategy also performed very well for query operations during both light-load and heavy-load data insertion.

Department: Computer Engineering Student's Signature .....

Field of Study: Computer Science Advisor's Signature .....

Academic Year: 2016

กิตติกรรมประกาศ

ขอกราบขอบพระคุณผู้ช่วยศาสตราจารย์ ดร. ณัฐวุฒิ หนูไพโรจน์ อาจารย์ที่ปรึกษาวิทยานิพนธ์อย่างสูงที่กรุณาสละเวลาให้ความรู้คำปรึกษาและแนะแนวทางการวิจัยตลอดจนถึงแนวทางการแก้ไขปัญหา ตลอดระยะเวลาการจัดทำวิทยานิพนธ์ฉบับนี้ ทำให้งานสามารถสำเร็จลุล่วงไปได้ด้วยดี ขอขอบพระคุณเป็นอย่างยิ่ง

กราบขอบพระคุณผู้ช่วยศาสตราจารย์ ดร. วีระ เหมืองสิน ผู้ช่วยศาสตราจารย์ ดร. เกริก ภิรมย์โสภา และ ดร. กาญจนา ศิลาวราเวทย์ คณะกรรมการสอบวิทยานิพินธ์ที่ได้ให้ความเห็นและคำชี้แนะซึ่งเป็นประโยชน์อย่างมากต่อการปรับปรุงวิทยานิพนธ์ฉบับนี้

ขอบคุณเพื่อนๆ วิทยาศาสตร์คอมพิวเตอร์ วิศวกรรมคอมพิวเตอร์ และ วิศวกรรมซอฟต์แวร์ ที่ได้ศึกษาเล่าเรียนร่วมกันมา และคอยเป็นกำลังใจให้กันและกันทั้งด้านกิจกรรมต่างๆ ความร่วมมือ การศึกษาและการทำวิทยานิพนธ์ จนสามารถผ่านพ้นไปได้ด้วยดี

ท้ายที่สุดขอขอบคุณบิดามารดาพี่น้องและญาติสนิทที่คอยเป็นกำลังใจให้เสมอมาตลอดระยะเวลาที่เข้ารับการศึกษานี้

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง

บทคัดย่อภาษาอังกฤษ ..... จ

กิตติกรรมประกาศ ..... ฉ

สารบัญ ..... ช

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 วัตถุประสงค์งานวิจัย 3

1.3 ขอบเขตการวิจัย 3

1.4 ขั้นตอนการดำเนินงานวิจัย 4

1.5 ประโยชน์ที่ได้รับจากงานวิจัย 4

1.6 งานวิจัยที่ได้รับการตีพิมพ์ 4

บทที่ 2 งานวิจัยและทฤษฎีที่เกี่ยวข้อง 5

2.1 งานวิจัยที่เกี่ยวข้อง 5

2.2 ทฤษฎี CAP 8

2.3 ระบบฐานข้อมูล โนเอสคิวแอล (NoSQL) 9

2.4 ลักษณะภาระงาน (Workload) 11

2.5 การแบ่งส่วนข้อมูล (Data partitioning) 13

2.5.1 การแบ่งข้อมูลแบบแนวตั้ง (Vertical Partitioning) 13

2.5.2 การแบ่งข้อมูลแบบแนวนอน (Horizontal Partitioning) 13

2.6 ปัญหาฮอตสปอต (Hotspot Problem) 14

2.7 แนวทางการจัดสรรข้อมูล (Data Allocation) 15

2.7.1 พิจารณาค่าต้นทุนในการเคลื่อนย้าย (Migration Cost) 15

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

หน้า

2.7.2เพื่อความสมดุลของตัวจัดเก็บข้อมูล (Storage Balance) 15
2.7.3คำนึงถึงลักษณะรูปแบบการเข้าถึงข้อมูล (Data Access Pattern) 16
2.7.4เน้นกระจายภาระงานของข้อมูล (Workload Allocation) 16
2.8 ฐานข้อมูลโนเอสคิวแอล มองโกดีบี (MongoDB) 16
2.8.1 โครงสร้างของระบบมองโกดีบี 17
2.8.2 การทำชาร์ตดิ้งบนมองโกดีบี 18
2.8.3 การเลือกชาร์ตคีย์ที่เหมาะสมสำหรับระบบที่พัฒนา 18
2.9 กลไกการทำงานของชาร์ตดิ้งกับแผนการกระจายข้อมูล 19
2.9.1 ชาร์ตดิ้งกับการกระจายข้อมูลแบบพิสัย (Range Sharding) 19
2.9.2 ชาร์ตดิ้งกับการกระจายข้อมูลแบบแฮช (Hash Sharding) 20
2.10 ระบบปรับสมดุลอัตโนมัติบนมองโกดีบี 21
บทที่ 3 การออกแบบการจัดสรรชิ้นส่วนข้อมูลบันทึกเหตุการณ์ 23
3.1 แบบแผนการวิเคราะห์ข้อมูลบันทึกเหตุการณ์ 23
3.2 การทำชาร์ตดิ้งกับข้อมูลประเภทบันทึกเหตุการณ์ 24
3.3 ข้อดีและข้อเสียจากแผนการกระจายข้อมูลบนมองโกดีบี 24
3.4 ระบบชาดดิ้งที่ตระหนักถึงป้ายบอกข้อมูล (Tag Aware Shading) 26
3.5 การปรับชาร์ตให้เหมาะสมกับโอเปอเรชั่นของข้อมูลบันทึกเหตุการณ์ 27
บทที่ 4 การทดลองและสรุปผลงานวิจัย 30
4.1 การทดสอบประสิทธิภาพ 30
4.1.1 การเตรียมระบบที่ใช้ในการทดลอง 30
4.1.2 ระบบจำลองข้อมูลบันทึกเหตุการณ์ 30
4.1.3 เคสที่ใช้ในการทดลอง 31
4.2 การรวบรวมข้อมูลเพื่อประมวลผล 32

หน้า

4.2.2 การรวมรวบข้อมูลการใช้ประโยชน์ซีพียู 33
4.2.3 การเก็บค่าประสิทธิภาพการสอบถามข้อมูล 34
4.3 ประสิทธิภาพของการบันทึกข้อมูล (Write Performance) 36
4.4 ประสิทธิภาพการสอบถามข้อมูลของการกระจายแบบพิสัยและแฮช 37
4.5 ประสิทธิภาพของการกระจายแบบติดป้ายข้อมูล 40
บทที่ 5 ข้อสรุปงานวิจัย 42
รายการอ้างอิง 44
ประวัติผู้เขียนวิทยานิพนธ์ 47

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในยุคของข้อมูลขนาดใหญ่ (Big Data) ปริมาณของข้อมูลมีการเพิ่มปริมาณที่รวดเร็ว และมีเป็นจำนวนมาก ส่งผลให้มีการพัฒนาเทคโนโลยีในด้านตัวจัดเก็บข้อมูลต่างๆ เพื่อตอบสนองต่อความต้องการระบบที่สามารถขยายตัวที่ค่อนข้างสูง และสามารถนำข้อมูลที่มีขนาดใหญ่และมีความซับซ้อนสูงเหล่านั้น มาทำการวิเคราะห์เพื่อให้ได้ประโยชน์และองค์ความรู้อย่างมีประสิทธิภาพ (1) โดยเฉพาะฐานข้อมูลแบบไร้ความสัมพันธ์โนเอสคิวแอล (NoSQL) หรือย่อมาจาก Not only SQL (2) ซึ่งถูกพัฒนาขึ้นมาเพื่อใช้ตอบโจทย์ข้อมูลขนาดใหญ่ และเป็นทางเลือกสำหรับฐานข้อมูลนอกเหนือจากระบบฐานข้อมูลแบบเชิงสัมพันธ์ (relational database) ที่มีความสัมพันธ์ระหว่างกลุ่มข้อมูลที่ซับซ้อน ทำให้ยากต่อการขยายตัว และไม่เหมาะสมสำหรับการนำมาใช้เก็บข้อมูลที่มีขนาดใหญ่ ระบบโนเอสคิวแอลมีคุณลักษณะที่เหมาะสมต่อระบบแบบกระจายที่ต้องรองรับข้อมูลขนาดใหญ่ ซึ่งคุณลักษณะเหล่านั้นได้แก่ ความสามารถในการขยายตัวของระบบเมื่อมีความต้องการเพิ่มขึ้น (Scalability) ความพร้อมของระบบเพื่อให้ตอบรับต่อการบริการ (Availability) และ ความคงทนต่อข้อผิดพลาดที่เกิดขึ้นในระบบ (Partition Tolerance) โดยโนเอสคิวแอลมีลักษณะที่แตกต่างกับระบบฐานข้อมูลแบบเชิงสัมพันธ์ในเรื่องการลดความสำคัญของโครงสร้างและความสัมพันธ์ของข้อมูลเพื่อให้สะดวกต่อการกระจายข้อมูล ส่งผลให้ฐานข้อมูลประเภทดังกล่าวสามารถรับข้อมูลใหม่ได้มากและสามารถนำออกไปประมวลผลได้สะดวกยิ่งขึ้น

ในงานวิจัยนี้ได้เลือกระบบฐานข้อมูลแบบโนเอสคิวแอลที่ชื่อว่ามองโกดีบี (3) (MongoDB) มาใช้ในการทดลอง โดยที่ระบบดังกล่าว เป็นโนเอสคิวแอลประเภทเชิงเอกสาร (Document Based) ซึ่งได้รับความนิยมอย่างมากในกลุ่มผู้ใช้โนเอสคิวแอล (4) และมีโครงสร้างข้อมูลที่ค่อนข้างยืดหยุ่น ทำให้สามารถปรับใช้เข้ากับข้อมูลได้หลายประเภท รวมถึงการที่รองรับภาษาที่ใช้ในการเขียนโปรแกรมได้หลายภาษา ถึงแม้ระบบดังกล่าว จะมีลักษณะเด่นที่หลากหลาย และสามารถตอบสนองถึงความต้องการของข้อมูลขนาดใหญ่ได้ แต่ก็ยังมีข้อจำกัดในด้านประสิทธิภาพการทำงาน เมื่อระบบทำงานหนักเกินขีดจำกัดที่สามารถรองรับไหว จนเป็นเหตุให้ได้มีการพัฒนาระบบเสริมและงานวิจัยจำนวนมาก ที่มีเป้าหมายในการบริหารความสมดุลของปริมาณงานในระบบ เพื่อที่จะไม่ให้เกิดเหตุการณ์ที่จะส่งผลกระทบต่อประสิทธิภาพการทำงาน (5) (6) (7) (8) ปัญหาที่ส่งผลกระทบต่อการทำงานอย่างฮอตสปอต (hotspot) ได้ถูกยกมาเป็นเป้าหมายสำคัญในงานวิจัยจำนวนมาก โดยที่ปัญหาดังกล่าวเกิดขึ้นในระบบแบบกระจายเมื่อเครื่องสมาชิกในระบบเครื่องหนึ่ง ได้มีการใช้งานทรัพยากรเครื่องที่

มากเกินกว่าค่าเฉลี่ยของเครื่องสมาชิกอื่นๆในระบบ ส่งผลให้ประสิทธิภาพการทำงานของเครื่องที่เกิดปัญหาดังกล่าวลดลงอย่างมาก และยังส่งผลต่อภาพรวมการทำงานของระบบอีกด้วย ทั้งนี้ลักษณะการเกิดของฮอตสปอต ยังแตกต่างกันออกไปตามรูปแบบลักษณะการเข้าถึงข้อมูล โดยเฉพาะการจัดเก็บข้อมูลและการนำออกไปวิเคราะห์ ลักษณะของการเข้าถึงข้อมูลนั้นเกี่ยวโยงกับประเภทของข้อมูลที่ถูกจัดเก็บ ซึ่งสอดคล้องกับงานวิจัยของ Cruz และคณะ (5) ที่ได้กล่าวว่าภาระงาน (workload) แต่ละประเภท จะส่งผลต่อระบบแตกต่างกันออกไป ซึ่งมันจะเป็นประโยชน์อย่างมากต่อการพัฒนาระบบถ้าสามารถทำให้รองรับการทำงานกับภาระงานประเภทต่างๆเหล่านั้นได้ อาจสรุปได้ว่า การพัฒนาระบบนั้น ควรที่จะต้องคำนึงถึงรูปแบบของข้อมูลที่ถูกจัดเก็บด้วย เพื่อให้ระบบสามารถนำทรัพยากรไปใช้ได้อย่างมีประสิทธิภาพและสมดุลมากที่สุด

ข้อมูลบันทึกเหตุการณ์ (Log Data) เป็นข้อมูลที่แสดงถึงเหตุการณ์ที่เกิดขึ้นในระบบคอมพิวเตอร์ เช่น รายละเอียดและการเข้าใช้งานของผู้ใช้ระบบ เหตุการณ์ที่เกิดขึ้นในระบบเครือข่าย และ สถานะของระบบ ซึ่งข้อมูลดังกล่าวมีความจำเป็นอย่างยิ่งที่จะต้องทำการจัดเก็บไว้ เพื่อนำไปใช้ประโยชน์ในด้านต่างๆ เช่น การนำมาใช้วิเคราะห์ด้านความปลอดภัย และ การนำมาใช้ปรับปรุงพัฒนาระบบ เป็นต้น ซึ่งโดยปกติแล้ว ข้อมูลบันทึกเหตุการณ์ จะถูกส่งมาจากแหล่งข้อมูลตลอดเวลาเป็นแบบตามเวลาจริง และมักมีปริมาณที่มากทำให้ข้อมูลมีอัตราการเติบโตที่ค่อนข้างสูง จึงมีโอกาสที่กลุ่มข้อมูลจะมีขนาดใหญ่มาก จนเกินความสามารถที่ระบบจะรองรับได้ ซึ่งจะส่งผลโดยตรงต่อประสิทธิภาพของระบบจัดเก็บข้อมูล โดยเฉพาะอย่างยิ่ง ในระบบแบบกระจาย ซึ่งอาจจะนำไปสู่ปัญหาแบบฮอตสปอตได้ ถ้าการจัดเก็บข้อมูลมีรูปแบบที่ไม่เหมาะสม จากการศึกษางานวิจัยต่างๆ และจากการทดลองบนมองโกดีบีพบว่า สิ่งที่เป็นเป็นตัวแปรสำคัญที่ส่งผลต่อลักษณะการเคลื่อนไหวของข้อมูล และประสิทธิภาพการทำงานของระบบในกรณีของข้อมูลบันทึกเหตุการณ์ มาจากแผนการกระจายของข้อมูล (Data Distribution) โดยที่แผนการการกระจายข้อมูลในมองโกดีบีมีสองวิธีคือ การกระจายแบบพิสัย (Range Distribution) และการกระจายแบบแฮช (Hash Distribution) โดยที่ลักษณะการกระจายข้อมูลแบบพิสัย จะจัดเก็บข้อมูลตามระยะพิสัยของข้อมูลนั้น ส่วนการกระจายข้อมูลแบบแฮชจะทำการเข้ารหัสคีย์ข้อมูลด้วยวิธีแฮชก่อนที่จะกระจายข้อมูลออกไปตามเครื่องสมาชิกในระบบอย่างทัดเทียมกัน ซึ่งในเชิงทฤษฎีแล้วการกระจายแบบพิสัยจะทำงานได้ดีกว่าแบบแฮชในด้านการอ่านข้อมูลแบบต่อเนื่อง เนื่องจากการเข้าถึงข้อมูลแบบต่อเนื่องสามารถกระทำได้ในคราวเดียว ข้อมูลจึงสามารถถูกอ่านได้โดยทันทีจากระบบ แต่แผนการดังกล่าวจะประสบปัญหาเรื่องประสิทธิภาพที่ลดลงอย่างมากในด้านการบันทึกข้อมูล เนื่องจากข้อมูลจะถูกส่งไปบันทึกที่เครื่องเซอเวอร์เพียงเครื่องเดียวตลอดเวลา ซึ่งจะส่งผลให้เกิดฮอตสปอตในเครื่องเซอเวอร์ดังกล่าวได้ ในขณะที่การกระจายข้อมูลแบบแฮช จะมีความสามารถในการบันทึกข้อมูลที่ค่อนข้างสูง เพราะข้อมูลจะถูกกระจายลงไปในเครื่องสมาชิกหลายเครื่องพร้อมๆกัน แต่ต้องแลกกับความสามารถในด้านการ

อ่านข้อมูลแบบต่อเนื่องที่ลดลง เนื่องจากการเข้าถึงข้อมูล จะต้องนำคีย์ข้อมูลมาเข้ารหัสด้วยระบบแฮชก่อนที่จะสามารถระบุเครื่องเซอเวอร์ที่มีข้อมูลดังกล่าวได้ และเนื่องจากการเข้ารหัสด้วนระบบแฮช จะส่งผลให้ข้อมูลถูกจัดเก็บกระจายแบบสุ่ม จึงทำให้การค้นหาข้อมูลแบบต่อเนื่องแต่ละครั้ง ระบบจะต้องทำการกวาดตรวจจากเครื่องสมาชิกทุกเครื่อง ถึงแม้ว่าปัญหานี้ สามารถถูกแก้ไขได้โดยการทำดรรชนี แต่วิธีดังกล่าวจะส่งผลกระทบต่อปริมาณความจุของหน่วยความจำของระบบ และการบันทึกข้อมูลแต่ละครั้งยังต้องทำการปรับดรรชนีให้ล่าสุดตลอดเวลา ซึ่งจะส่งผลกระทบต่อประสิทธิภาพของการบันทึกข้อมูล โดยเฉพาะอย่างยิ่ง เมื่อระบบมีสภาพแวดล้อมที่ข้อมูลมีปริมาณการไหลเข้าที่ความเร็วสูง

จากสภาพแวดล้อมข้อมูลประเภทบันทึกเหตุการณ์ที่มีอัตราการเพิ่มขึ้นของข้อมูลที่ค่อนข้างสูงมาก ระบบจึงจำเป็นที่จะต้องมีประสิทธิภาพในด้านการบันทึกข้อมูลที่สูง โดยที่ยังจะต้องคงประสิทธิภาพในการค้นหาข้อมูลแบบต่อเนื่องไว้ด้วย ซึ่งวิธีการกระจายข้อมูลทั้งสองแบบยังไม่สามารถตอบโจทย์ดังกล่าวได้ เนื่องจากทั้งสองวิธีจะส่งผลให้ระบบมีประสิทธิภาพดีเพียงด้านการบันทึก หรือด้านการค้นหาเท่านั้น ในงานวิจัยนี้จึงมุ่งเน้นไปที่การศึกษาแนวทางที่จะนำมาช่วยเสริมประสิทธิภาพของแบบแผนการกระจายข้อมูลบนมองโกดีบี ให้ส่งผลกระทบต่อทรัพยากรระบบน้อยลง โดยที่ยังคงไว้ซึ่งประสิทธิภาพในด้านจัดเก็บและการเข้าถึงข้อมูลของแบบแผนทั้งสองแบบ

1.2 **วัตถุประสงค์งานวิจัย**

โครงงานวิจัยนี้มีวัตถุประสงค์หลักดังต่อไปนี้

1.2.1 เพื่อศึกษาแนวทางการจัดสรรข้อมูลบันทึกเหตุการณ์โดยใช้ฐานข้อมูลโนเอสคิวแอลมองโกดีบีเป็นระบบจัดเก็บข้อมูล

1.2.2 เพื่อศึกษาประสิทธิภาพของแผนการกระจายข้อมูลต่างๆต่อลักษณะของข้อมูลบันทึกเหตุการณ์

1.2.3 เพื่อศึกษาผลกระทบจากปัญหาฮอตสปอตที่ส่งผลต่อประสิทธิภาพการทำงานที่เกิดขึ้นในระบบ และศึกษาแนวทางในการลดทอนผลกระทบจากปัญหาเหล่านี้

1.3 **ขอบเขตการวิจัย**

ในงานวิจัยนี้จะมีขอบเขตสำหรับการวิจัยดังต่อไปนี้

1.3.1 ระบบฐานข้อมูลที่จะนำมาใช้ในการวิจัยนี้จะเป็นโนเอสคิวแอลประเภทเอกสารชื่อ มองโกดีบี เท่านั้น

1.3.2 ข้อมูลที่จะนำมาใช้ในงานวิจัยนี้จะเป็นข้อมูลบันทึกเหตุการณ์ ที่มีการอ้างอิงจากโครงสร้างข้อมูลจากระบบบันทึกเหตุการณ์สารสนเทศของจุฬาลงกรณ์มหาวิทยาลัย

1.3.3 การวิจัยนี้จะมุ่งเน้นไปที่การศึกษาวิธีจัดสรรข้อมูลและนำมาพิจารณาด้านประสิทธิภาพเพื่อนำไปประยุกต์ใช้เป็นหลัก

1.3.4 แนวทางการกระจายข้อมูลที่จะศึกษาในงานวิจัยนี้ได้แค่ การกระจายแบบพิสัย การกระจายแบบแฮช และ แนวทางที่เสนอในงานวิจัย

1.4 **ขั้นตอนการดำเนินงานวิจัย**

ในงานวิจัยนี้จะมีขั้นตอนในการดำเนินการดังต่อไปนี้

1.4.1 ศึกษาแนวทางการจัดสรรข้อมูลจากงานวิจัยที่เกี่ยวข้อง

1.4.2 ศึกษาคุณลักษณะและกลไกการทำงานของ โนเอสคิวแอล มองโกดีบี

1.4.3 เตรียมพร้อมระบบและข้อมูลทดสอบเพื่อที่จะนำมาใช้ในการทดลอง

1.4.4 ทำการทดลองประสิทธิภาพแนวทางการจัดสรรข้อมูลต่างๆ

1.4.5 สรุปผลการทดลองและจัดทำรูปเล่มวิทยานิพนธ์

1.5 **ประโยชน์ที่ได้รับจากงานวิจัย**

1.5.1 สามารถเข้าใจถึงหลักการและกลไกการทำงานของระบบมองโกดีบีต่อข้อมูลบันทึกเหตุการณ์ และ สามารถนำไปประยุกต์ใช้กับระบบอื่นๆได้อีกด้วย

1.5.2 การศึกษาลักษณะข้อดีข้อเสียของแนวทางการกระจายข้อมูลแบบต่างๆสามารถนำไปประยุกต์ใช้กับข้อมูลบันทึกเหตุการณ์ได้อย่างเหมาะสม

1.5.3 ปัญหาประสิทธิภาพการทำงานที่เกิดขึ้นจะส่งผลกระทบต่อภาพรวมของการทำงานในระบบน้อยลง

1.6 **งานวิจัยที่ได้รับการตีพิมพ์**

งานวิจัยดังกล่าวได้รับการบรรจุลงในระบบฐานข้อมูลของ IEEE เมื่อวันที่ 30 มีนาคม 2560 ภายใต้ชื่อ “Hotspot Management Strategy for Real-time Log Data in MongoDB” โดยได้รับการตอบรับตีพิมพ์ในงานสัมนาวิชาการชื่อ International Conference on Advanced Communication Technology (ICACT), 2017 19th ซึ่งจัดขึ้นที่ประเทศเกาหลีใต้ ระหว่างวันที่ 19 ถึง 22 กุมภาพันธ์ 2560

บทที่ 2
งานวิจัยและทฤษฎีที่เกี่ยวข้อง

2.1 งานวิจัยที่เกี่ยวข้อง

การได้มาซึ่งแนวทางในการทำให้การทำงานของระบบกระจายมี ประสิทธิภาพและสมดุลมากที่สุดเป็นหัวข้อที่มีแนวทางการวิจัยที่หลากหลายโดยเป้าหมายคือการได้มาซึ่ง ระบบที่สามารถตอบรับและปรับสภาพให้เข้ากับภาระงาน (workload) ที่เข้ามาในระบบหรือระบบที่ทำให้ประสิทธิภาพเต็มที่ด้วยการกระจายงานออกไปให้สมดุลที่สุด (load balancing) นั้นมีมีเป้าหมายเพื่อแก้ไขเหตุการณ์ที่มีการใช้งานทรัพยากรทั้งหมดในระบบได้อย่างไม่มีประสิทธิภาพหรือเครื่องสมาชิกในระบบที่ทำงานหนักจนเกิดเป็นปัญหาฮอตสปอต เนื่องจากแบกรับ ปริมาณงานที่มากกว่าเครื่องอื่นในระบบ แนวทางที่ได้รับความสนใจและศึกษาสามารถแบ่งได้ออกเป็นสองวิธีหลักๆคือ การปรับขยาย (Scaling) ซึ่งเป็นการปรับขยายความสามารถของระบบด้วยวิธีต่างๆ เช่นเพิ่มหรือลด ปริมาณของเครื่องสมาชิกซึ่งส่วนใหญ่จะเน้น และ การจัดสรรข้อมูล (Data Allocations) ซึ่งเป็นการจัดการข้อมูลเพื่อให้เกิดการกระจายกำลังการทำงานออกไปยังเครื่องสมาชิกที่มีอยู่ได้อย่างมีประสิทธิภาพ โดยทั้งสองประเภทงานวิจัยจะมีเป้าหมายหลักที่ใกล้เคียง กันคือ การเน้นไปที่ความสามารถของระบบในการตอบสนองต่อเหตุการณ์ที่เกิดขึ้นได้อัตโนมัติ และสามารถจัดการบริหารทรัพยากรได้ทันท่วงทีต่อปัญหาที่เกิดขึ้น

HasLab ได้พัฒนาระบบ MeT (5) ซึ่งเป็นระบบที่ทำให้การการตระหนักถึงภาระงานสำหรับระบบฐานข้อมูลโนเอสคิวแอล ระบบดังกล่าวถูกสร้างอยู่บนฐานข้อมูลโนเอสคิวแอลที่เรียกว่า Hbase โดยงานนี้ได้กล่าวอ้างว่าลักษณะการเข้าถึงข้อมูล (Data Access Pattern) นั่นมีส่วนสำคัญอย่างยิ่งในการที่จะพัฒนาประสิทธิภาพการทำงานให้ดียิ่งขึ้นไป โดยที่ระบบและงานวิจัยเก่าๆจะมุ่งเน้นไปที่ตัดวัดต่างๆที่เพื่อที่ทำให้ระบบสามารถที่จะลบและเพิ่มสมาชิกระบบตามต้องการได้เท่านั้นโดนงานนี้ได้เปรียบเทียบประสิทธิภาพกับงานวิจัยเก่าของระบบที่ชื่อ TIRAMOLA (6) ซึ่งเป็นระบบสำหรับโอเอสคิวแอลที่เก็บตัววัดต่างๆจากระบบแบบกระจายเพื่อที่จะนำมาใช้ตัดสินใจในการเคลื่อนย้ายข้อมูล สิ่งที่ MeT ได้นำเสนอเป็นวิธีการแก้ปัญหาด้วยการปรับแก้โครงแบบอย่างหลากหลาย (Heterogeneous Configuration) เพื่อให้ระบบแต่ละระบบสามารถที่จะตอบรับกับลักษณะการทำงานได้หลากหลายยิ่งขึ้น รูปแบบประเภทของลักษณะการเข้าถึงข้อมูลในงานนี้ได้ทำการอ้างอิงจาก YSCH (9) ทำให้ลักษณะงานที่ใช้ในการทดลองนี้ครอบคลุมกับระบบต่างๆในปัจจุบัน และเนื่องจากสิ่งที่งานดังกล่าวเสนอมีความหลากหลายในด้านข้อมูล งานนี้นอกเหนือจากนำเสนออัลกอริทึมในการคำนวณตัววัดจากระบบและคำนวณหาคำตอบเพื่อการเคลื่อนย้ายแล้ว ยังนำเสนออัลกอริทึมเพื่อ

จัดแยกกลุ่มข้อมูลตามลักษณะการเข้าถึงข้อมูลแต่ละประเภทซึ่งผลประเมินทำให้ทราบว่าความแตกต่างของลักษณะการเข้าถึงข้อมูลนั้นมีส่วนสำคัญและวิธีที่จะรับมือเพื่อให้ระบบสามารถทำงานได้เต็มประสิทธิภาพก็แตกต่างกันไปด้วย

Chenกับคณะ (7) ได้กล่าวว่า โนเอสคิวแอลเป็นระบบฐานข้อมูลที่คำนวณใกล้กับข้อมูล ดังนั้นแล้วการรูปแบบแผนการแบ่งส่วนและจัดสรรข้อมูลนั้นส่งผลกระทบโดยตรงต่อประสิทธิภาพการทำงานของระบบปรับสมดุลในฐานข้อมูลขนาดใหญ่โดยงานดังกล่าวได้นำเสนอแผนการจัดสรรชิ้นส่วนข้อมูลโดยตระหนักถึงภาระสำหรับโนเอสคิวแอล โดยการคำนวณค่าของภาระที่เกิดขึ้นในแต่ละสมาชิกของระบบ เพื่อเป็นการกระจายข้อมูลจากเครื่องที่ทำงานหนักลงไปสู่เครื่องที่ไร้ภาระโดยมีกลไกการคำนวณสองขั้นตอนคือ หนึ่ง คำนวณหาคุณลักษณะของน้ำหนักของภาระโดยลักษณะจะแบ่งเป็นสองรูปแบบคือ ข้อมูลที่มีประโยชน์กับข้อมูลที่เป็นส่วนที่เป็นภาระ โดยข้อมูลส่วนที่เป็นภาระจะถูกเคลื่อนย้ายไปยังเป้าหมายเมื่อการค่าแปรผันของภาระอยู่ในระดับที่ไม่สมดุล โดยหลักการเคลื่อนย้ายข้อมูลจะมาจากค่า ภาระของแต่ละเครื่อง เครื่องที่จะเป็นปลายทางจะมีค่าที่ต่ำกว่าค่าแปรผันภาระเพื่อจะนำมารองรับชิ้นส่วนข้อมูลมีภาระสูงที่เคลื่อนย้ายมาจากเครื่องที่เคยมีภาระในการทำงานที่สูงกว่าค่าแปรผัน และยังได้มีวิจัยต่อยอดจากงานดังกล่าวโดยการนำเสนอแนวคิดการกระจายชิ้นส่วนข้อมูลแนวใหม่ (8) โดยแนวทางใหม่ดังกล่าวคือการกระจายข้อมูลด้วยการใช้ Hypergraph ซึ่งในเชิงคณิตศาสตร์คือการทำให้กราฟมีความเป็นทั่วไป(Generalize)มากขึ้นโดยการนำขอบเขตของกราฟมาทับซ้อนกันในส่วนทีข้อมูลมีความเกี่ยวข้อง โดยที่เป้าหมายหลักของการกระจายในงานนี้คือการทำให้ข้อมูลที่ทับซ้อนหรือมีความสัมพันธ์กันจับกลุ่มอยู่ใกล้กันมากขึ้นเพื่อความสะดวกในการประมวลผล และการกระจายข้อมูลให้ออกไปในสภาพคล้ายคลึงกันให้มากที่สุดเพื่อใช้ประโยชน์จากการคำนวณแบบคู่ขนาน (Parallel Processing) และท้ายที่สุดคือการทำระบบมีความคงทนและพร้อมใช้งานมากขึ้นโดยงานวิจัยดังกล่าวได้ใช้กลไกการเคลื่อนย้ายชิ้นส่วนที่พัฒนามาก่อนหน้า (7) มาใช้ในการเคลื่อนย้ายชิ้นส่วนข้อมูลวิธีใหม่ และยังพัฒนาแนวคิดในด้านการเคลื่อนย้ายเพิ่มขึ้นอีกสี่แนวทางได้แก่ การเคลื่อนย้ายข้อมูลตามค่าภาระต้นทุน (Cost Based) ความสมดุลของตัวจัดเก็บข้อมูล (Storage Based) ต้นทุนก่อนความสมดุลทีหลัง (Cost First Balance Second) และ สมดุลก่อนต้นทุนทีหลัง (Balance First Cost Second) ซึ่งแต่ละวิธีจะมีการใช้งานและเป้าหมายที่ต่างกันไปตามสถานการณ์

Yahoo! (10) ได้พัฒนาระบบปรับสมดุลให้กับระบบจัดเก็บข้อมูลบนคราวของตัวเองที่เรียกว่า Sherpa โดยเป้าหมายหลักของการวิจับนี้คือการแก้ไขและบั่นทอนปัญหาฮอตสปอต โดยงานมีเป้าหมายหลักจากประสบการณ์ที่ได้ทำการพัฒนาระบบดังกล่าวไว้ว่าการจะทำระบบปรับสมดุลนั้นจำเป็นที่จะต้องเข้าใจถึงเป้าหมายของแต่ระบบอย่างเช่นในกรณีระบบดังกล่าวที่ต้องรับมือกับปัญหา

ฮอตสปอต การทำระบบให้ง่ายต่อการใช้งานและเพิ่มเติมหรือปรับแต่งรายละเอียด และสุดท้ายคือ ระบบจะต้องพยายามไม่ให้กินทรัพยากรเครื่องสูง ในงานนี้เนื่องจากว่าทำการแก้ไขปัญหาเกี่ยวกับฮอตสปอต จึงได้นำเสนอสูตรคำนวณเครื่องร้อน(Storage Heat)ออกมาเพื่อใช้คำนวณถึงปัญหาฮอตสปอตที่เกิดขึ้น โดยคำนวณจากเวลาแฝง(latency)และคำขอของงาน(request) และอีกสูตรจะเป็นการคำนวณความร้อนของชิ้นส่วนข้อมูลที่อยู่ในเครื่องร้อน โดยการเรียกชิ้นส่วนว่า แผ่นร้อน (Tablet Heat) ซึ่งสูตรดังกล่าวจะคำนวณจาก ชิ้นส่วนข้อมูลไหนที่ถูกเข้าถึงบ่อยครั้งมากที่สุดซึ่งจะมาจาก ขนาดชิ้นส่วนน้ำหนักของแต่ละภาระงานและค่าต้นทุนที่เกิดขึ้น ซึ่งเมื่อได้ค่าตามที่ต้องการแล้ว เครื่องร้อนจะทำการส่องแผ่นขอ้มูลที่ร้อนไปยังเครื่องที่เย็น

Konstantinouและคณะได้พัฒนาระบบชื่อ Dbalancer (11) ซึ่งในงานดังกล่าวได้นำเสนอการแบ่งแนวทางในการเคลื่อนย้ายข้อมูลออกไปสองแนวทางหลักๆคือ NIX (Neighboring Item Exchange) และ MIG (Migration) NIX เป็นการเคลื่อนย้ายข้อมูลไปสู่เครื่องที่ใกล้กับเครื่องต้นทางมากที่สุดจุดเสียของวิธีนี้คือเมื่อระบบมีสมาชิกจำนวนมาก ทำให้เกิดปัญหาในด้านการทำระบบสมดุลจึงเป็นไปได้ยากเนื่องจากข้อมูลจะยังคงอยู่ในพื้นที่ใกล้เคียงของเครื่องต้นทาง ส่วน MIG เครื่องต้นทางจะทำการติดต่อเครื่องที่อยู่ห่างออกไปเพื่อทำการขอเคลื่อน ข้อมูล วิธีนี้อาจจะช่วยทำให้ข้อมูลกระจายออกไปได้ไกลขึ้นแต่วิธีนี้จำเป็นต้องใช้การ ติดต่อสื่อสารกันระหว่างเครื่องต้นทางและปลายทางทำให้อาจจะเกิดปัญหาในด้านการรักษา ตัวกลางที่จะใช้เป็นตัวกลางในสื่อสารระหว่างเครื่องได้ สุดท้ายแล้วจากปัญหาข้างต้นงานวิจัยนี้จึงเสนอแบบที่สามคือ NIXMIG หรือการนำวิธีข้างต้นทั้งสองวิธีมาใช้ร่วมกันโดยระยะแรกของระบบจะทำการตรวจหา เครื่องที่สามารถรับข้อมูลเพิ่มได้ก่อนจะใช้วิธี NIX ทำการเคลื่อนย้ายข้อมูลทังหมดออกไปในลักษณะของคลื่นจนถึงปลายทาง

งานวิจัยของLinและคณะ (12) เป็นงานที่มีโจทย์และเป้าหมายคล้ายครึงกับงานวิจัยนี้ที่สุดโดยที่ เป้าหมายหลักๆคือการแก้ไขปัญหาเมื่อเกิดฮอตสปอตบนระบบของมองโกดีบีโดยงานนี้ได้กล่าวว่าระบบปรับสมดุลอัตโนมัติของมองโกดีบีนั้นไม่สามารถตอบโจทย์การแก้ไขปัญหาด้านสมดุลได้เพราะไม่สนเรื่องลักษณะการเข้าถึงข้อมูล โดยหลักการวิจัยนี้ได้เลือกที่จะใช้วิธีการสอดส่องและตรวจจับข้อมูลที่เป็นต้นเหตุของปัญหาฮอตสปอตก่อนที่จะเคลื่อนย้ายและกระจายข้อหมายเหล่านั้นออกไปในระบบ ซึ่งระบบดังกล่าวเป็นระบบที่ขึ้นกับภาระงานเป็นหลัก โดยการคำนวณที่เสนอในงานวิจัยนี้มาจากคำนวณค่าการใช้ประโยชน์ซีพียูของแต่ละเครื่องในระบบก่อนที่จะนำจัดเรียงตามความหนักของภาระน้ำหนักของแต่ละเครื่องที่คำนวณได้ เพื่อใช้เป็นเกณฑ์วัดในการเคลื่อนย้ายข้อมูลจากเครื่องที่ทำงานหนักไปสู่เครื่องที่ทำงานเบาที่สุด

2.2 **ทฤษฎี CAP**

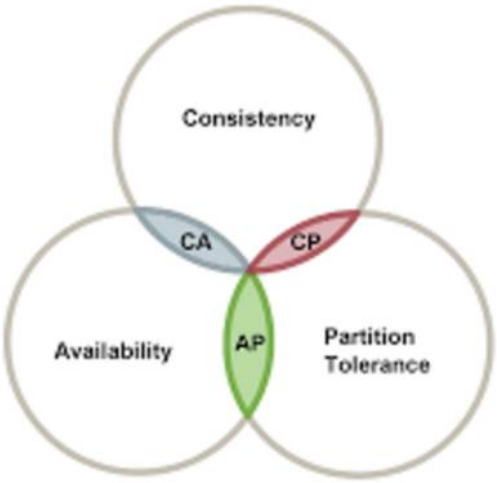
เป็นทฤษฎีที่กล่าวโดย เอริค บริวเวอร์ (13) ถึงลักษณะสำคัญในระบบกระจายซึ่งเป็นหลักการที่ใช้ในการพิจารณาประเภทระบบฐานข้อมูลต่างๆ ทฤษฎีดังกล่าวมาจากลักษณะสามอย่าง โดยเรียกแทนด้วยตัวอักษรภาษาอังกฤษคือ C ความสอดคล้องกัน (Consistency) A ความพร้อมใช้งาน (Availability) และ P ความคงทนต่อการแบ่งส่วน (Partition Tolerance) โดยลักษณะดังกล่าวจะไม่สามารถเป็นจุดเด่นได้พร้อมกันทั้งสามข้อเนื่องจากการที่เน้นไปที่ด้านหนึ่งจะทำให้ความสามารถในอีกด้านหนึ่งลดลงไปด้วย โดยที่ลักษณะต่างๆสามารถอธิบายได้ดังนี้

ความสอดคล้องกัน (Consistency) ทุกเครื่องในระบบจะเห็นข้อมูลเหมือนกัน ยกตัวอย่าง เช่นในระบบหนึ่งมีการกระจายข้อมูลออกไปสองเครื่องข้อมูลทั้งสองเครื่อง จำเป็นที่จะต้องมี เวอร์ชั่นตรงกันเกือบตลอดเวลาจึงจะสามารถบอกได้ว่าระบบนี้มีความสอดคล้องกันสูง ระบบจัดเก็บข้อมูลแบบสัมพันธ์ที่นิยมใช้กันทั่วไปจัดเป็นระบบที่มีความความสอดคล้องกันที่สูง

ความพร้อมใช้งาน (Availability) ระบบสามารถตอบรับการเข้าถึงได้ตลอดเวลา ซึ่งหมายความว่าระบบดังกล่าวจะสามารถติดต่อและตอบรับต่องานที่เข้ามาได้ไม่ว่าจะเกิดอะไรขึ้นกับระบบ โดยลักษณะดังกล่าวเป็นส่วนสำคัญสำหรับระบบประเภทแบบ สังคมเครือข่าย ซึ่งมีผู้เข้าใช้ตลอดเวลา

ความคงทนต่อการแบ่งส่วน (Partition Tolerance) ระบบสามารถทำงานต่อได้แม้จะเกิด ปัญหาขึ้น ลักษณะแบบนี้จะเห็นได้ชัดเจนที่สุดในระบบแบบกระจายที่มีการกระจายการทำงาน ออกไปแตกต่างกับระบบที่มีการรวมการทำงานไว้ที่เดียวซึ่งเมื่อเกิดปัญหาก็อาจจะทำให้ระบบ ต้องหยุดทำงานลงทั้งระบบ แต่ในกรณีนี้ ถ้าเครื่องใดเครื่องหนึ่งเกิดขัดข้อง ระบบก็จะสามารถ หาเครื่องที่เป็นตัวแทนมารับงานต่อได้

ลักษณะเด่นของระบบฐานข้อมูลต่างๆสามารถนำมาใช้จำแนกประเภทหลักๆได้ตามทฤษฎี CAP โดยจะมาจากการรวมกันของจุดเด่นสองข้อ ตัวอย่างเช่น ฐานข้อมูลแบบเชิงสัมพันธ์ถูกจัดอยู่ในประเภท CA ซึ่งเป็นลักษณะที่เน้นการให้บริการและข้อมูลที่สอดคล้องโดยที่ลดทอนความสามารถในส่วนความคงทนต่อการแบ่งส่วนลงไป เนื่องจากลักษณะโครงสร้างความสัมพันธ์ข้อมูลที่ซับซ้อนทำให้การแบ่งส่วนและกระจายข้อมูลเป็นไปได้ยาก



รูปที่ 1 ประเภทของฐานข้อมูลตามทฤษฎี CAP

2.3 **ระบบฐานข้อมูลโนเอสคิวแอล (NoSQL)**

เป็นระบบฐานข้อมูลที่ถูกพัฒนาขึ้นมาเพื่อตอบโจทย์ข้อมูลขนาดใหญ่ โดยพัฒนามาเพื่อให้สามารถทำงานในระบบแบบกระจายได้เป็นอย่างดี จุดที่ โนเอสคิวแอลแตกต่างจากระบบฐานข้อมูลทั่วไปนั้นคือการที่ข้อมูลไม่มีโครงสร้างความสัมพันธ์ตามแบบปกติ ข้อมูลจะถูกจัดอยู่ในรูปแบบของคีย์ ซึ่งจากลักษณะดังกล่าวทำให้ข้อมูลจะถูกจัดเก็บแบบเรียบง่ายโดยมีคีย์เป็นตัวกำหนดของข้อมูลแถวหนึ่ง โนเอสคิวแอลเองยังแบ่งออกได้เป็นสี่ประเภทที่มีการทำงานรูปแบบข้อมูลที่แตกต่างและตอบโจทย์ที่แตกต่างกันออกไปโดยประเภทต่างๆสามารถแบ่งออกได้ดังต่อไปนี้

โนเอสคิวแอลแบบขึ้นกับค่าคีย์ (Key Based) เป็นรูปแบบพื้นฐานของโนเอสคิวแอลที่จะมีแนวคิดเก็บข้อมูลทุกอย่างให้อยู่ในรูปแบบของคีย์โดยไม่คำนึงถึงลักษณะใดๆของข้อมูลเพื่อความสะดวกรวดเร็วในการจัดเก็บข้อมูล ฐานข้อมูลที่อยู่ในประเภทนี้ได้แก่ Riak และ Redis

โนเอสคิวแอลแบบขึ้นกับคอลัมน์ (Column Based) ในรูปแบบนี้จะมีการกำหนดชื่อของคัมลัมน์ขึ้นมาเพื่อใช้ในการกำหนดลักษณะโครงสร้างของข้อมูล โดยที่ข้อมูลจะยังคงถูกจัดเก็บในลักษณะคีย์เหมือนเดิม ฐานข้อมูลที่อยู่ในประเภทนี้ได้แก่ Cassandra และ Hbase

โนเอสคิวแอลแบบขึ้นกับเอกสาร (Document Based) ข้อมูลในโนเอสคิวแอลประเภทนี้จะถูกจัดเก็บในรูปแบบของเอกสารหรือเป็นวัตถุเช่น JSON หรือ XML โดยแต่ละเอกสารจะมีคีย์เป็นตัวกำหนด โนเอสคิวแอลประเภทนี้จะมีโครงสร้างที่ยืดหยุ่นมากขึ้น ฐานข้อมูลที่อยู่ในประเภทนี้ได้แก่ MongoDB และ CouchDB

โนเอสคิวแอลแบบขึ้นกับกราฟ (Graph Based) เป็นระบบที่นำกราฟมาใช้แก้ไขปัญหาที่ โนเอสคิวแอลไม่มีโครงสร้างความสัมพันธ์ที่ชัดเจน การเลือกใช้ระบบประเภทนี้จะเป็นกรณีที่ความสัมพันธ์ข้อมูลไม่สามารถถูกนำออกไปได้แม้จะอยู่ในสภาพแวดล้อมที่มีข้อมูลเป็นปริมาณมาก โดยที่กราฟดังกล่าวจะถูกสร้างมาเพื่อจำลองความสัมพันธ์ของข้อมูลต่างๆ โดยข้อมูลที่อยู่ในกลุ่มเดียวกันจะมีการเชื่อมโยงให้อยู่ในบริบทเดียวกัน ฐานข้อมูลที่จัดอยู่ในประเภทนี้ได้แก่ Neo4j

โนเอสคิวแอลประเภทต่างๆจะข้อดีข้อเสียและลักษณะที่แตกต่างกันออกไป และยังสามารถจำแนกประเภทได้ด้วยทฤษฎีแคปตามตารางที่ 1 จากตารางเปรียบเทียบจะสังเกตได้ว่า NoSQL ประเภทต่างๆมีลักษณะการทำงานการตอบ โจทย์ปัญหาและข้อดีข้อเสียที่แตกต่างกันออกไปตามแต่ละชนิด การจะเลือกนำมาใช้จึงจำเป็นที่จะต้องเลือกให้เข้ากับระบบที่จะใช้ในงานวิจัยนี้ให้มากที่สุด

ประเภท NoSQL	ทฤษฎี CAP	การจัดเก็บข้อมูล	การปรับขยาย	ความพร้อมระบบ	โครงสร้างข้อมูล
Key Value	AP	สูง	สูง	สูง	ต่ำ
Column	AP	ปานกลาง	ปานกลาง	ปานกลาง	ปานกลาง
Document	CP	ปานกลาง	ปานกลาง	ปานกลาง	ปานกลาง
Graph	CP	ต่ำ	ต่ำ	สูง	สูง

ตารางที่ 1 ตารางเปรียบเทียบลักษณะของประเภทฐานข้อมูลโนเอสคิวแอล

จุฬาลงกรณ์มหาวิทยาลัย

ได้มีการศึกษาและเปรียบเทียบ (14) เพื่อเป็นถึงความแตกต่างระหว่างโนเอสคิวแอลชนิดต่างๆ โดยได้มีตัวชี้วัดที่บอกถึงความแตกต่างของแต่ละระบบเช่น ภาษาที่ใช้พัฒนา, รูปแบบข้อมูล, ชุมชนที่พัฒนาระบบ, เทคโนโลยีที่ใช้ในการจัดเก็บข้อมูล, ภาษาที่ใช้ในการสอบถามข้อมูล ซึ่งในงานวิจัยได้สรุปความเหมาะสมสำหรับระบบโนเอสคิวแอลไว้ดังตารางต่อไปนี้

NoSQL	ประเภท NoSQL	เหมาะสำหรับ
Riak	Key-Value	ต้องการความพร้อมและความคงทนต่อการแบ่งส่วนที่สูง
Redis	Key-Value	ระบบที่มีการเขียนและเปลี่ยนแปลงข้อมูลบ่อย อ่านข้อมูลน้อย
MongoDB	Document	ความหลากหลายในคำสั่งสอบถามข้อมูล สามารถทำดัชนีได้ ประสิทธิภาพดีสำหรับฐานข้อมูลขนาดใหญ่
CouchDB	Document	ข้อมูลที่มีการเปลี่ยนเป็นระยะๆ คำสั่งสอบถามข้อมูลที่มีการกำหนดไว้แล้ว เว็บไซต์และโปรแกรมประยุกต์บนมือถือ
Cassandra	Column	เมื่อระบบมีการเขียนมากกว่าการอ่าน
Neo4J	Graph	ข้อมูลที่มีรายละเอียดและความสัมพันธ์ที่สูง

ตารางที่ 2 ตารางจุดเด่นขอโนเอสคิวแอลชนิดต่างๆ

งานวิจัยนี้ได้เลือกใช้มองโกดีบีซึ่งเป็นโนเอสคิวแอลประเภทเอกสารเป็นหลัก เนื่องจากคุณลักษณะที่มาจากการเก็บไฟล์ข้อมูลในรูปแบบที่คล้ายคลึงกับ JSON ทำให้ข้อมูลที่สามารถจัดเก็บลงในระบบมีความหลากหลายและสามารถตอบโจทย์ข้อมูลบันทึกเหตุการณ์ที่บางระบบมีโครงสร้างข้อมูลที่แตกต่างและเด่นชัดได้เป็นอย่างดี รวมถึงมีจุดเด่นในด้านการสอบถามข้อมูลที่มีความหลากหลายที่สูงทำให้สามารถวิเคราะห์ปัญหาหรือผลลัพธ์ได้อย่างสะดวกมากขึ้น เมื่อเทียบกับระบบอื่นที่ใกล้เคียง

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2.4 ลักษณะภาระงาน (Workload)

ลักษณะการเข้าภาระงานสามารถจำแนกได้โดยดูจากการกระทำที่เกิดขึ้นในระบบโดยสามารถ สังเกตได้จากการดำเนินงานต่างๆในระบบเช่น read, write และ update เป็นหลัก YCSB (9) (Yahoo ! Cloud Service Benchmark) ได้กำหนด ลักษณะงานเบื้องต้นที่สามารถพบเห็นได้บ่อยในระบบทั่วไป และทางผู้ใช้ยังสามารถที่จะปรับค่า เพื่อใช้จำลองกับระบบการทำงานที่มีลักษณะพิเศษ

ลักษณะงานพื้นฐานแบ่งออกได้ดังต่อไปนี้ตามตารางที่ 3

ภาระงาน A ปรับข้อมูลหนัก อ่าน 50% ปรับ 50% ระบบที่ใช้เก็บSessionหรือระบบที่มีการเปลี่ยนแปลงบ่อย

ภาระงาน **B** อ่านหนัก อ่าน 95% ปรับ 5% การปรับป้ายข้อมูลให้กับรูปถ่ายใหม่ ซึ่งมีการอ่านข้อมูล ป้ายเป็นส่วนใหญ่ ก่อนที่จะเลือกเพิ่มป้ายข้อมูล

ภาระงาน **C** อ่านอย่างเดียว อ่าน 100% ระบบประเภทแคช ที่ข้อมูลจะถูกรวบรวมที่อื่น

ภาระงาน **D** อ่านล่าสุด อ่าน 95% ใส่ 5% ข้อความของระบบสนทนาและสถานะตามระบบเครือข่าย

ภาระงาน **E** ระยะพิสัยเล็ก กราดตรวจ 95% ใส่5% ข้อความการสนทนาตามเว็บบอร์ดต่างๆ ที่จะมีการ กราดตรวจข้อมูลทุกอันบนระบบ

Workload	Operations	Record selection	Application example
A—Update heavy	Read: 50% Update: 50%	Zipfian	Session store recording recent actions in a user session
B—Read heavy	Read: 95% Update: 5%	Zipfian	Photo tagging; add a tag is an update, but most operations are to read tags
C—Read only	Read: 100%	Zipfian	User profile cache, where profiles are constructed elsewhere (e.g., Hadoop)
D—Read latest	Read: 95% Insert: 5%	Latest	User status updates; people want to read the latest statuses
E—Short ranges	Scan: 95% Insert: 5%	Zipfian/Uniform*	Threaded conversations, where each scan is for the posts in a given thread (assumed to be clustered by thread id)

ตารางที่ 3 รูปแบบภาระงานเบื้องต้นที่ถูกกำหนดโดย YCSB (9)

นอกเหนือจากรูปแบบงานพื้นฐานของ YCSB งานวิจัย (5) ยังได้แก้ไขและเพิ่มรูปแบบลักษณะงานเข้าไปซึ่งมีรูปแบบงานหนึ่งที่สามารถใช้อธิบายลักษณะภาระงานของระบบบันทึกเหตุการณ์ได้ดีและยังตรงกับลักษณะข้อมูลที่มีอยู่นั่นคือ ใส่ข้อมูลหนัก (Insert Heavy) โดยที่มีส่วนประกอบเป็น อ่าน5% และใส่ 95%

การเลือกข้อมูลเพื่อที่จะใช้ในการจำลองการกระจายภาระงานในระบบ YCSHB นั้นจะมีหลักการอยู่สี่แบบซึ่งจะมีรายละเอียดดังต่อไปนี้

Uniform การเลือกที่จะกระจายงานแบบแบบแผนโดยการสุ่มไปทั่วๆ อย่างเช่นเมื่อเอกสารถูกหยิบ ออกมาทุกๆเอกสารจะมีสิทธิเท่ากันที่จะถูกเลือก

Zipfian ใช้แนวการกระจายแบบ Zipfian ในการเลือกข้อมูล โดยข้อมูลที่มีความนิยมสูงจะถูกเลือก

Latest จะคล้ายกับวิธีการ Zipfian แต่จะแตกต่างกันที่ ข้อมูลในวิธีดึงกล่าวจะถูกเลือกจากลำดับล่าสุดที่ เข้ามาในระบบ

Multinomial โอกาสที่ข้อมูลแต่ละชิ้นจะถูกเลือกนั้นสามารถกำหนดได้เป็นอัตราความน่าจะเป็น

2.5 การแบ่งส่วนข้อมูล (Data partitioning)

การที่จะจัดสรรข้อมูลนั้นจะต้องคำนึงถึงวิธีที่ใช้ในการแบ่งข้อมูลของระบบฐานข้อมูลที่เลือก เนื่องจากหน่วยข้อมูลที่จะใช้ในการเคลื่อนย้ายจัดเรียง จะส่งผลกระทบต่อผลลัพธ์ และวิธีการในการจัดสรรข้อมูลใหม่โดยตรง วิธีในการแบ่งข้อมูลแบ่งเป็นสองวิธีดังต่อไปนี้

2.5.1 การแบ่งข้อมูลแบบแนวตั้ง (Vertical Partitioning)

เป็นการแบ่งส่วนข้อมูลโดยแบ่งข้อมูลแยกข้อมูลในคอลัมน์ออกจาก กันวิธีนี้จะใช้ได้ต่อเมื่อข้อมูลมีการกำหนดคอลัมน์และความหมายที่ชัดเจน เป้าหมายของการ แบ่งส่วนข้อมูลวิธีนี้คือการแยกข้อมูลส่วนที่มีแนวโน้มที่จะถูกนำไปใช้บ่อยครั้งแยกออกจากข้อมูล ส่วนที่มีการใช้งานน้อย ซึ่งจะทำให้ตัวงานสามารถแบ่งส่วนที่จำเป็นต้องใช้ทรัพยากรสูงออกจาก ตัวส่วนที่ไม่จำเป็นต้องเรียกใช้ตลอดเวลาได้

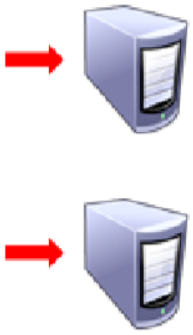
User	password	email	profile	photo
Admin	1234	admin@proposal.com	10kb	1mb
User	password	User@proposal.com	15kb	1mb
Mick	password	mick@proposal.com	20kb	1mb

รูปที่ 2 รูปแบบการแบ่งข้อมูลแบบแนวตั้ง

2.5.2 การแบ่งข้อมูลแบบแนวนอน (Horizontal Partitioning)

หรือการทำ **ชาร์ตดิ้ง (Sharding)** เป็นการแบ่งข้อมูลโดยการแบ่งจากระยะแถว ข้อมูลจากแถว เนื่องจากข้อมูลของโนเอสคิวแอลมีการเก็บเป็น คีย์โดยที่ไม่มีตัวคอลัมน์มากำหนดความหมายข้อมูล การแบ่งชิ้นส่วนข้อมูลบนระบบโนเอสคิวแอลส่วนใหญ่จึงอยู่ในรูปแบบนี้

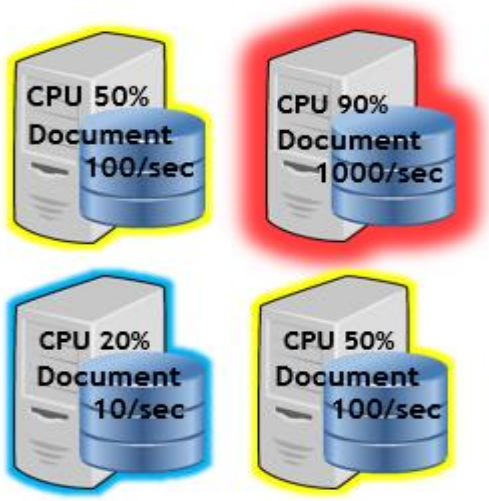
Key	Data
0475830271	22065 INFO Authentication: Max sessions policy passed 2014-10-05T00:00:01.000+07:00
0475830272	22064 INFO Authentication: New accounting session created in Session cache 2014-10-05T00:00:01.000+07:00
0475830289	22069 INFO Authentication: The accounting session was updated in Session cache 2014-10-05T00:00:01.000+07:00
0475830298	22078 INFO Authentication: Audit session was not found. 2014-10-05T00:00:01.000+07:00
0475830459	22079 Authentication: Accounting start was received for non-existing session 2014-10-05T00:00:01.000+07:00



รูปที่ 3 รูปแบบการแบ่งข้อมูลแบบแนวนอน

2.6 ปัญหาฮอตสปอต (Hotspot Problem)

ปัญหาฮอตสปอตเป็นปัญหาที่พบได้บ่อยในระบบแบบกระจาย ซึ่งปัญหาดังกล่าวสามารถแปลตรงตัวได้ว่า จุดที่ร้อน ซึ่งเป็นการอ้างอิงถึงสถานการณ์ในระบบแบบกระจายเมื่อเครื่องสมาชิกเครื่องใดเครื่องหนึ่ง มีภาระการทำงานที่สูงเกินกว่าภาพรวมของระบบเป็นอย่างมาก ทำให้ส่งผลกระทบต่อการทำงานโดยรวมของระบบตกลงอย่างมาก (12) โดยเฉพาะการทำงานที่จำเป็นที่จะต้องดึงทรัพยากรจากเครื่องที่เกิดปัญหาฮอตสปอต



รูปที่ 4 ตัวอย่างระบบที่เกิดปัญหาฮอตสปอต

ตัวชี้วัดการเกิดฮอตสปอตนั้นสามารถจำแนกได้เป็น ทรัพยากรที่ใช้ในการประมวลผล เนื้อที่จัดเก็บข้อมูล รวมถึงจำนวนภาระงานที่เข้าถึงระบบด้วย จากรูปที่ 4 เป็นการจำลองเหตุการณ์ที่เมื่อ

ระบบเกิดปัญหาฮอตสปอตขึ้นโดยจะใช้ ค่าซีพียูกับจำนวนเอกสารขาเข้าต่อวินาทีเป็นหลัก สิ่งที่ใช้เป็นตัวชี้วัดฮอตสปอตนั้นสามารถจำแนกได้ตามนี้คือ

2.7 **แนวทางการจัดสรรข้อมูล (Data Allocation)**

จากการศึกษาแนวทางจากงานวิจัยที่เกี่ยวข้อง ทำให้สามารถจำแนกเป้าหมายและกฎเกณหลักของการ
เคลื่อนย้ายข้อมูลออกได้เป็นสี่กลุ่มหลักๆคือ ค่าต้นทุนในการเคลื่อนย้าย ความสมดุลของตัวจัดเก็บข้อมูล ลักษณะรูปแบบการเข้าถึงข้อมูล และการกระจายตามภาระงาน

2.7.1 **พิจารณาค่าต้นทุนในการเคลื่อนย้าย (Migration Cost)**

คำนึงถึงค่าต้นทุนที่จะใช้ในการเคลื่อนย้ายข้อมูลเป็นหลัก ระยะเวลาที่ใช้ใน การเคลื่อนย้ายข้อมูลและทรัพยากรที่จะถูกนำไปใช้ในระหว่างการเคลื่อนย้ายข้อมูล เนื่องจากโดยส่วนใหญ่ค่าต้นทุนในการเคลื่อนย้ายนั้นถ้าค่ามีค่าน้อยหมายความว่าเครื่องที่จะทำการย้าย ข้อมูลและเครื่องที่จะทำการรับข้อมูลมีตำแหน่งที่อยู่ใกล้กัน ทำให้สามารถสรุปได้ว่าการเน้น เป้าหมายนี้เป็นหลักจะทำให้ข้อมูลกระจายอยู่กันเป็นกลุ่มที่มีตำแหน่งใกล้เคียงกันไม่กระจาย ออกไปทั่วๆทั้งระบบ ในทางกลับกันถ้าค่าต้นทุนในการเคลื่อนย้ายมีค่ามากอาจจะส่งผลกับประสิทธิภาพการทำงานหลายๆอย่าง ซึ่งจะมีปัจจัยสำคัญที่เกี่ยวข้องอีกเช่น ขนาดของชิ้นส่วนข้อมูล เสถียรภาพของระบบเครือข่าย งานวิจัย (11) เป็นตัวอย่างการจัดเรียงเคลื่อนย้ายข้อมูลด้วยแนวทางนี้ซึ่งจะใช้วิธีการเคลื่อนย้ายไปเครื่องที่ใกล้ก่อนหรือเครื่องที่พร้อมรับข้อมูลแต่อยู่ในระยะไกล

2.7.2 **เพื่อความสมดุลของตัวจัดเก็บข้อมูล (Storage Balance)**

คำนึงถึงความเท่าเทียมของข้อมูลในระบบเป็นหลักเพื่อให้สามารถใช้ประโยชน์จากความสามารถของโอเอสคิวแอลอย่างเช่นการคำนวณแบบคู่ขนานให้มากที่สุด มีตัวชี้วัดคือข้อมูลที่ถูกจัดเก็บอยู่ในแต่ละเครื่องบนระบบจะต้องมีปริมาณที่ใกล้เคียงกับค่าเฉลี่ยจากค่าภาระรวมกันของทุกเครื่องให้มากที่สุด แต่วิธีนี้อาจ ส่งผลให้ข้อมูลที่มีความใกล้เคียงกันถูกกระจายออกไปด้วย ทำให้เวลาการนำข้อมูลออกมาใช้มี ความซับซ้อนมากยิ่งขึ้นเนื่องจากเวลาทำการค้นหาจะต้องค้นหาจากเครื่องในระบบทั้งหมดซึ่ง อาจจะส่งผลให้การตอบรับมีความล่าช้ากว่าปกติ แนวทางการกระจายข้อมูลที่มีเป้าหมายดังกล่าวเป็นสำคัญคือการกระจายแบบแฮชที่มีการคำนวณค่าแฮชขึ้นมาจากข้อมูลที่ให้เพื่อให้ค่าที่ใช้ในการกระจายอยู่ในลักษณะที่ความความสุ่มที่สูง

2.7.3 **คำนึงถึงลักษณะรูปแบบการเข้าถึงข้อมูล (Data Access Pattern)**

ในแต่ละระบบจะมีลักษณะการเข้าถึงข้อมูลที่แตกต่างกันออกไปซึ่งส่งผลถึงภาระงานที่จะเกิดขึ้นในระบบที่ได้มีการจำแนกโดย YSCB (9) ทำให้การใช้ทรัพยากรในระบบและปัญหาที่จะตามมาจะมีความแตกต่างกันออกไป โดยในแต่ละงานวิจัยก็มีวิธีการสังเกตุลักษณะข้อมูลและการแก้ไขปัญหาที่แตกต่างกันออกไป อย่างเช่นระบบ MeT (5) แสดงถึงแนวทางระบบที่มีลักษณะการเข้าถึงข้อมูลที่หลากหลาย ก่อนที่จะจัดกลุ่มข้อมูลด้วยการศึกษาถึงชนิดของงานที่เกิดขึ้นในระบบ โดยระบบดังกล่าวใช้ประโยชน์จากการกำหนดและปรับค่าหน่วยความจำและขนาดแคชเพื่อให้เข้ากับภาระงาน และ แนวทางการจัดเรียงชิ้นส่วนข้อมูลของ (8) ที่ใช้ hypergraph เข้ามาช่วยการทำให้ข้อมูลที่มีความสัมพันธ์กันเกาะกลุ่มกันมากขึ้น

2.7.4 **เน้นกระจายภาระงานของข้อมูล (Workload Allocation)**

เป็นแนวทางที่มุ่งเน้นไปที่การตอบสนองของภาระงานเป็นหลัก เป้าหมายในการแก้ไขปัญหาประสิทธิภาพแบบฮอตสปอตเป็นแนวทางที่หลายงานวิจัยได้เลือกเป็นเป้าหมายสำคัญในการแก้ไขปัญหา โดยที่ระบบที่เน้นแนวทางดังกล่าวนั้นมีส่วนที่คล้ายคลึงกันคือ ระบบเหล่านั้นจะทำการรวบรวมตัวชี้วัดต่างๆที่เกิดขึ้นกับระบบ และทำการเคลื่อนย้ายข้อมูลโดยการอ้างอิงจากค่าเหล่านั้นเป็นสำคัญ อย่างเช่นงานของ Yahoo! (10) ที่ได้นำเสนอการคำนวณความร้อนเครื่องและชิ้นส่วนข้อมูลซึ่งมาจากค่าเวลาแฝงคำนวณกับชิ้นส่วนข้อมูลผลลัพธ์จากคำขอทั้งหมด ก่อนที่จะคำนวณความร้อนของชิ้นส่วนข้อมูลแยกออกมาอีกทีหนึ่งเพื่อใช้ในการเคลื่อนย้าย ส่วนงานวิจัย (12) ซึ่งนำเสนอการแก้ปัญหาการกระจายข้อมูลของมองโกดีบีนั้นจะเน้นไปที่การตรวจรวบรวมตัวชี้วัดจากทรัพยากรเครื่องซึ่งในงานนี้กล่าวถึง ซีพียู เพื่อที่จะทำการเคลื่อนย้ายชิ้นส่วนข้อมูลจากเครื่องที่มีภาระการทำงานหนักก่อนย้ายไปเครื่องที่มีการทำงานน้อยโดยไม่สนว่าเครื่องปลายทางจะมีชิ้นส่วนมากอยู่เพียงใดก็ตาม

2.8 **ฐานข้อมูลโนเอสคิวแอล มองโกดีบี (MongoDB)**

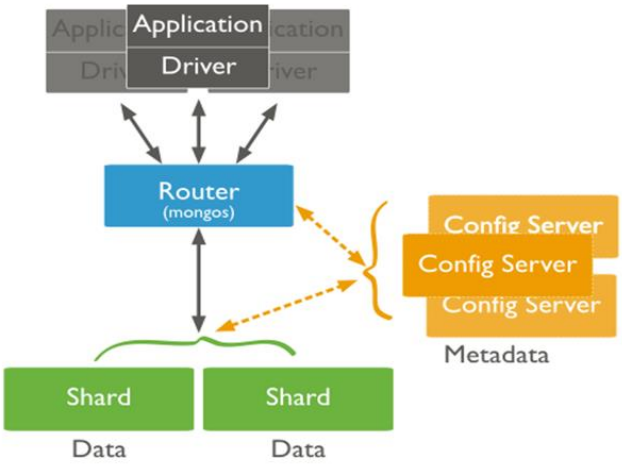
มองโกดีบี(3)เป็นฐานข้อมูลแบบโนเอสคิวแอลประเภทที่ขึ้นกับเอกสาร ที่มีการจัดเก็บข้อมูลอยู่ในรูปแบบเอกสาร BSON หรือ Binary JSON ซึ่งเป็นเอกสารประเภท JSON ที่ถูกเข้ารหัสให้อยู่ในลักษณะ Binary ทำให้รายละเอียดข้อมูลมีความยืดหยุ่นสูงและสามารถปรับใช้ได้กับข้อมูลหลายประเภทมากขึ้นทำให้มองโกดีบีได้รับความนิยมอย่างสูงในกลุ่มโนเอสคิวแอลด้วยกัน และยังมีระบบต่างๆที่เอื้ออำนวยต่อระบบที่มีข้อมูลขนาดใหญ่เช่นการประมวลผลแบบคู่ขนาน Map Reduce และการทำซ้ำข้อมูล

```
{
  name: "sue",                          ← field: value
  age: 26,                              ← field: value
  status: "A",                          ← field: value
  groups: [ "news", "sports" ]          ← field: value
}
```

รูปที่ 5 ลักษณะเอกสารข้อมูล BSON บน MongoDB

2.8.1 โครงสร้างของระบบมองโกดีบี

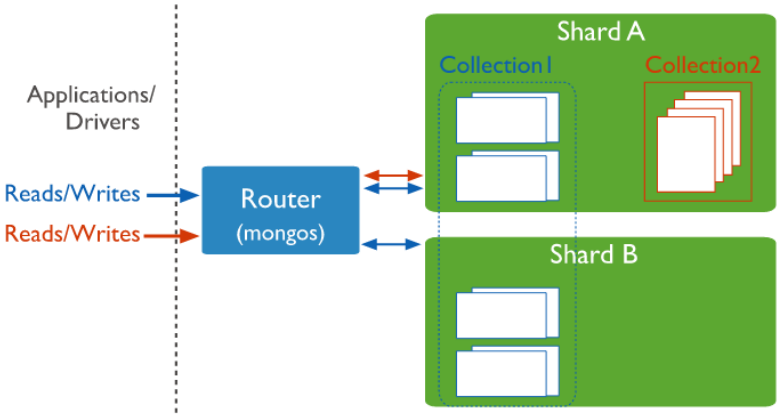
ส่วนประกอบสำคัญในระบบมองโกดีบี ประกอบไปด้วย เราเตอร์ คอนฟิกเซอเวอร์ และ ตัวจัดเก็บข้อมูล คอนฟิคเซอเวอร์จะทำการเก็บรายละเอียดเมทาดาตาของระบบรวมถึงรายละเอียดการกำหนดค่าต่างๆและตำแหน่งของข้อมูลที่ถูกจัดเก็บลงไปในตัวจัดเก็บข้อมูล โดยที่จะมีการติดต่อกันผ่านเราเตอร์ซึ่งจะเป็นตัวทำหน้าที่ประสานงานและรับคำขอจากทางฝั่งผู้ใช้ โดยที่มาตรฐานที่ถูกกำหนดไว้จากทางมองโกดีบี ในระบบมองโกดีบีหนึ่งระบบจำเป็นที่จะต้องประกอบไปด้วย ตัวจัดเก็บข้อมูลจำนวนสี่เครื่องเป็นขั้นต่ำ เนื่องจากการที่มีตัวจัดเก็บข้อมูลที่น้อยกว่ามาตรฐานจะทำให้ลักษณะการทำงานบางอย่างมีความผิดปกติไปจากลักษณะที่ควรจะเป็นอย่างเช่นการกระจายของชิ้นส่วนข้อมูลเพื่อปรับสมดุลชิ้นส่วนข้อมูลจะขึ้นกับความแตกต่างของปริมาณชิ้นส่วนข้อมูลในตัดจัดเก็บข้อมูลในระบบทั้งหมด เมื่อจำนวนตัวจัดเก็บข้อมูลมีน้อยจะยิ่งทำให้เกิดปัญหาอย่างเช่นฮอตสปอตได้ง่ายยิ่งขึ้นอีกด้วย



รูปที่ 6 ส่วนประกอบของระบบมองโกดีบี

2.8.2 การทำชาร์ตดิ้งบนมองโกดีบี

ชาร์ตดิ้งเป็นฟังชั่นที่ทำให้มองโกดีบีเข้าสู่สภาพระบบแบบกระจายและเริ่มทำการกระจายข้อมูล และจะทำให้ตัวจัดเก็บข้อมูลที่ถูกกำหนดให้เป็นสมาชิกในระบบชาร์ตดิ้งจะถูกเปลี่ยนเป็นเครื่องชาร์ต ในแต่ละฐานข้อมูลในหนึ่งชาร์ตจะสามารถจัดเก็บกลุ่มข้อมูลที่เรียกว่า Collection ได้หลายกลุ่มซึ่งในแต่ละกลุ่มนั้นจะมีชิ้นส่วนข้อมูล (Chunk) ย่อยลงมาอีกทีหนึ่งซึ่งเป็นตัวที่ใช้ในการใส่เอกสารที่เข้ามาในระบบ โดยแต่ละเอกสารจะมีระยะพิสัยเป็นตัวกำหนดซึ่งระยะพิสัยนี้จะมาจากการกำหนดค่าชาร์ตคีย์ (Shard Key) โดยการเลือกจากฟิลด์ของเอกสารซึ่งจะต้องกำหนดก่อนการทำชาร์ตดิ้งทุกครั้ง



รูปที่ 7 ส่วนประกอบของชาร์ตดิ้ง

2.8.3 การเลือกชาร์ตคีย์ที่เหมาะสมสำหรับระบบที่พัฒนา

การเลือกชาร์ตคีย์ที่ถูกต้องเป็นขั้นตอนแรกสุดของการกำหนดรูปแบบการกระจายข้อมูลในมองโกดีบี ซึ่งตัวชาร์ตคีย์จะทำหน้าที่เป็นตัวบ่งชี้ถึงเอกสารและชิ้นส่วนข้อมูล โดยข้อมูลชาร์ตคีย์นั้นจะเลือกมาจากฟิลด์ของเอกสาร ดังนั้นการเลือกชาร์คคีย์จึงจำเป็นที่จะต้องเลือกให้เข้ากับสภาพการใช้งานระบบให้มากที่สุด ดังนั้นแล้วเป้าหมายของการเลือกชาร์ตคีย์จึงสามารถแบ่งได้ออกเป็นสองกลุ่มดังนี้ได้แก่ เน้นสืบค้นข้อมูลและเน้นด้านกระจายข้อมูลขึ้นอยู่กับธรรมชาติของระบบนั้นๆ โดยหลักการเลือกชาร์ตคีย์จมีให้เลือกพิจารณาดังต่อไปนี้

2.8.3.1 เลือกค่าที่สามารถแยกชิ้นส่วนได้ง่าย

เพื่อให้ข้อมูลสามารถที่จะแยกชิ้นส่วนได้ง่าย ค่าที่มีข้อจำกัดในด้านตัวเลขหรือสมาชิกนั้นเป็นสิ่งที่ไม่ควรอย่างยิ่งเนื่องจากว่าจะทำให้ชิ้นส่วนข้อมูลไม่สามารถที่จะแยกชิ้นส่วนได้ ซึ่งจะส่งผลต่อเนื่องถ้าปริมาณเอกสารในชิ้นส่วนหลักเกินขีดจำกัด ชิ้นส่วนนี้จะไม่สามารถเคลื่อนย้ายได้อีก

2.8.3.2 การเลือกค่าที่มีอัตราการสุ่มที่สูง
เพื่อให้ระบบสามารถกระจายงานออกไปเก็บได้ตามชาร์ตได้อย่างเท่าเทียม

2.8.3.3 เลือกชาร์ตคีย์ให้เหมาะสมกับลักษณะการสอบถามข้อมูล
สำหรับการสอบถามข้อมูลบนมองโกดีบีแล้ว ระบบจะสามารถส่งคำตอบกลับมาได้ไวที่สุดต่อเมื่อสิ่งที่ค้นหานั้นอยู่บนชาร์ตคีย์ ในกรณีที่ข้อมูลที่ค้นหาไม่อยู่บนชาร์ตคีย์ระบบจะทำการกราดตรวจ ข้อมูลทุกชาร์ตเพื่อให้ได้มาซึ่งคำตอบ ซึ่งหมายความว่าระบบจะต้องทำการอ่านทุกเอกสารนั่นเอง

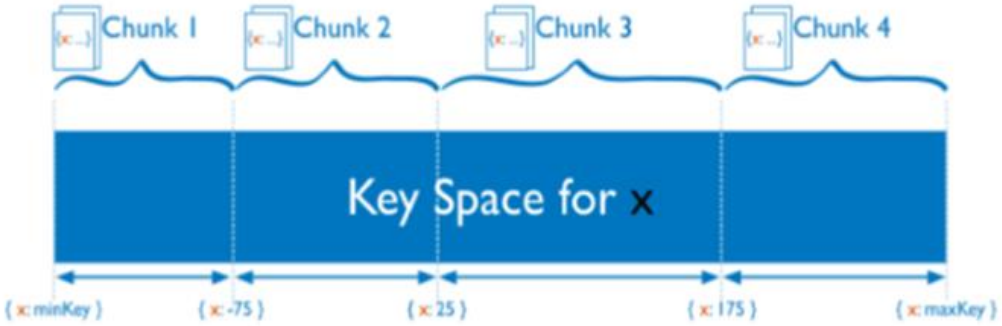
2.8.3.4 การใช้ชาร์ตคีย์ที่ประกอบเข้าด้วยกัน (Compound Shard Key)
ในกรณีที่ระบบนั้นไม่มีข้อมูลที่เหมาะสมที่สุดค่าเดียว การเลือกค่าใดค่าหนึ่งจึงไม่ใช่คำตอบที่ ดีที่สุด ดังนั้นแล้วการที่จะเลือกข้อมูลได้สองถึงสามฟิลด์นั้นจะช่วยลดปัญหาต่างๆได้บ้างอย่างเช่นการที่ใช้ค่าแรกเป็นค่าที่มีความเป็นคณิตสูง ในขณะที่ค่าที่สองเป็นค่าที่ตั้งเพื่อสำหรับความสะดวกในการสอบถามข้อมูล

2.9 **กลไกการทำงานของชาร์ตดิ้งกับแผนการกระจายข้อมูล**

การที่จะออกคำสั่งเปิดใช้งานชาร์ตดิ้ง จะต้องเริ่มจากการกำหนดแบบแผนการกระจายข้อมูลเสมอ โดยเริ่มจากการเลือกฟิลด์ข้อมูลมาใช้มาเป็นค่าคีย์ ซึ่งจะมาเป็นตัวกำหนดระยะพิสัยและลักษณะการแบ่งชิ้นส่วนข้อมูล แผนการกระจายข้อมูลที่สามารถเลือกมาใช้ได้ในระบบมองโกดีบีมีสองวิธีคือ แผนการกระจายข้อมูลแบบพิสัย และ แผนการกระจายข้อมูลแบแฮช ลักษณะพฤติกรรมของระบบชาร์ตดิ้งภายใต้แผนการกระจายทั้งสองมีความแตกต่างอย่างมากซึ่งจะมีลักษณะดังต่อไปนี้

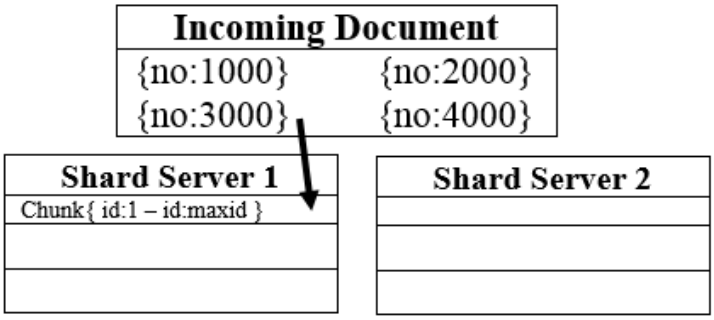
2.9.1 ชาร์ตดิ้งกับการกระจายข้อมูลแบบพิสัย (Range Sharding)

แผนการกระจายข้อมูลแบบพิสัยเป็นแผนการกระจายข้อมูลโดยขึ้นจากระยะของคีย์ วิธีดังกล่าวข้อมูลจะถูกจัดเก็บเรียงตามอันดับตัวเลขและอักษร ทำให้ข้อมูลถูกจัดเรียงตามลำดับการเข้าก่อนหลังและข้อมูลที่อยู่ใกล้เคียงกันจะอยู่ในระยะที่ใกล้เคียงกันไม่โดนกระจายออกไป



รูปที่ 8 แผนการกระจายข้อมูลแบบพิสัย

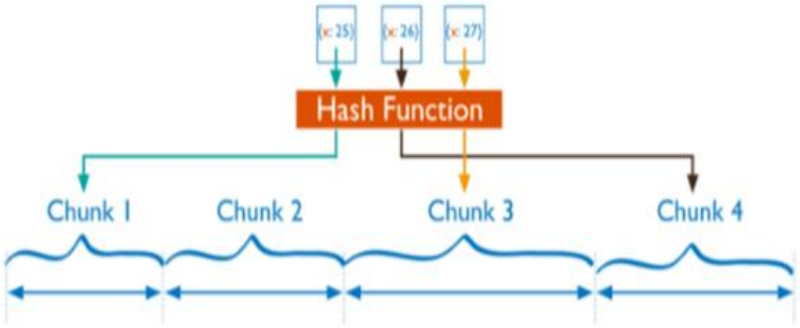
ชาร์ตดิ้งที่เลือกแนวทางการกระจายแบบพิสัย จะเริ่มทำการสร้างชิ้นส่วนข้อมูลขึ้นมาเพื่อรองรับ ข้อมูลต่อเมื่อได้รับเอกสารข้อมูลเป็นครั้งแรก ชิ้นส่วนข้อมูลที่มีค่า maxKey จะทำหน้าที่รับเอกสารใหม่ เสมอตราบใดที่ข้อมูลคีย์มีลักษณะที่เพิ่มขึ้นตลอดเวลาทำให้เอกสารใหม่จะถูกจัดเก็บลงไปเพียงชาร์ตที่ แบกรับชิ้นส่วนนี้เท่านั้น ชิ้นส่วนข้อมูลจะทำการแตกตัวออกมาเป็นชิ้นส่วนใหม่เมื่อขนาดชิ้นส่วนข้อมูลถึง ขีดจำกัดที่กำหนดไว้



รูปที่ 9 ตัวอย่างการกระจายข้อมูลแบบพิสัย

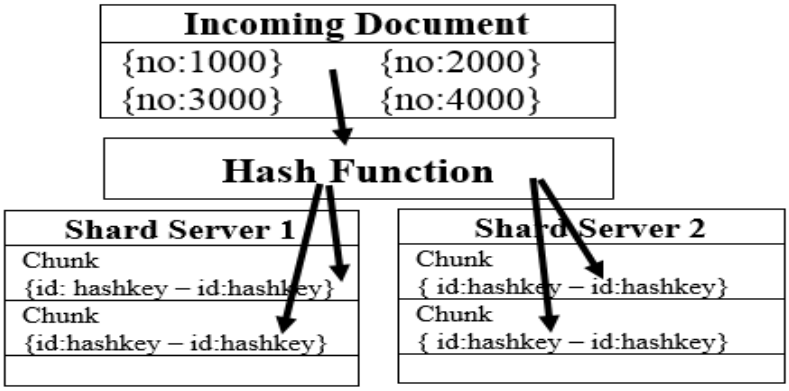
2.9.2 ชาร์ตดิ้งกับการกระจายข้อมูลแบบแฮช (Hash Sharding)

แผนการกระจายข้อมูลแบบแฮชมีเป้าหมายหลักคือการทำให้ข้อมูลถูกสุ่มกระจายออกไปตามชาร์ตในระบบอย่างเท่าเทียมกัน โดยการนำข้อมูลที่มีอยู่มาเข้าฟังชั่น hash ซึ่งจะมีการแปลงค่าคีย์เพื่อให้ค่าคีย์มีการเปลี่ยนแปลงขึ้นและไม่เกาะติดกันจนสามารถทำให้กระจายออกไปสู่เครื่องชาร์ตในระบบได้ในลักษณะแบบสุ่ม โดย ฟังชั่นแฮช ในกรณีมองโกดีบีจะใช้การเข้ารหัสแบบ MD5 ในการแปลงข้อมูล



รูปที่ 10 การแบ่งข้อมูลแบบแฮช

กลุ่มชิ้นส่วนข้อมูลจำนวนหนึ่งจะถูกสร้างขึ้นมาทันทีที่เปิดระบบชาร์ตดิ้งด้วยแผนการกระจายข้อมูลแบบแฮช โดยที่ระยะพิสัยของแต่ละชิ้นส่วนนั้นจะอยู่ในรูปแบบของค่าแฮชเพื่อรองรับข้อมูลใหม่ที่ชาร์ตคีย์จะต้องผ่านฟังก์ชั่นแฮชก่อนเพื่อเป็นการสุ่มกระจายข้อมูลลงไปในชาร์ตที่มีระยะคีย์เหล่านั้น



รูปที่ 11 การกระจายข้อมูลแบบแฮช

2.10 ระบบปรับสมดุลอัตโนมัติบนมองโกดีบี

ในสภาวะการทำงานของชาร์ตดิ้ง เมื่อชิ้นส่วนข้อมูลในชาร์ตหนึ่งมีปริมาณที่มากกว่าขีดแบ่งของระบบ ชิ้นส่วนข้อมูลในชาร์ตนั้นชิ้นที่มีความใหม่ที่สุดจะถูกเคลื่อนย้ายไปยังชาร์ตใหม่ที่มีชิ้นส่วนข้อมูลน้อยกว่าชาร์ตต้นทางหรือชาร์ตอื่น โดยเกณฑ์ของค่าขีดแบ่งในการเคลื่อนย้ายจะวัดจากปริมาณชิ้นส่วนในระบบทั้งหมด เมื่อชิ้นส่วนมีปริมาณมากขีดแบ่งปริมาณความต่างของชิ้นส่วนข้อมูลที่ใช้เป็นตัววัดจะยิ่งมีปริมาณที่มาก ทำให้ระยะห่างการเคลื่อนย้ายแต่ละครั้งก็จะยาวมากขึ้นไปด้วย ซึ่งหมายความว่าในกรณีที่ชิ้นส่วนข้อมูลที่ทำหน้าที่เป็น maxKey อยู่ที่ชาร์ตเครื่องใดเครื่องนั้นจะเป็นจุดที่มีการทำงานหนักในระยะเวลายาวและแบกรับข้อมูลที่มากกว่าเครื่องอื่น จากรูปที่6จะเห็นได้ว่าเมื่อชิ้นส่วนยังมีต่ำกว่า 20 ชิ้นระบบจะมีค่าขีดแบ่งอยู่ที่เพียง 2 ชิ้นส่วนเท่านั้นซึ่ง

จะขยับระยะห่างออกเป็น8ชิ้นส่วนเมื่อจำนวนชิ้นส่วนมีสูงถึง 80 ชิ้นสามารถสรุปได้ว่าเมื่อข้อมูลมีปริมาณที่มากถึงจุดหนึ่งการปรับสมดุลและเคลื่อนย้ายก็จะลดลงตามไปด้วย ระบบดังกล่าวนั้นจำเป็นที่จะต้องปิดในกรณีที่นักพัฒนาได้พัฒนาระบบปรับสมดุลเพื่อมาจัดการเรื่องเคลื่อนย้ายข้อมูลเอง

Number of Chunks	Migration Threshold
Fewer than 20	2
20-79	4
80 and greater	8

รูปที่ 12 ขีดแบ่งของการเคลื่อนย้ายชิ้นส่วนข้อมูล

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 3
การออกแบบการจัดสรรชิ้นส่วนข้อมูลบันทึกเหตุการณ์

ในบทนี้ จะกล่าวถึงการออกแบบการจัดสรรชิ้นส่วนข้อมูลที่เหมาะสมแก่การบริหารจัดการข้อมูลบันทึกเหตุการณ์ โดยจะเริ่มจากการพิจารณาลักษณะของแบบแผนการวิเคราะห์ข้อมูล เพื่อนำไปสู่การออกแบบการกระจายข้อมูลที่เหมาะสม

3.1 แบบแผนการวิเคราะห์ข้อมูลบันทึกเหตุการณ์

การนำระบบโนเอสคิวแอลอย่างเช่น มองโกดีบี มาบริหารจัดการข้อมูลบันทึกเหตุการณ์อย่างมีประสิทธิภาพนั้น จำเป็นอย่างยิ่งที่จะต้องทำการพิจารณารูปแบบการสอบถามข้อมูลบันทึกเหตุการณ์เพื่อนำมาวิเคราะห์ โดยในงานวิจัยนี้จะใช้ข้อมูลที่จำลองมาจากระบบบันทึกเหตุการณ์การใช้งานอินเตอร์เน็ตภายในจุฬาลงกรณ์มหาวิทยาลัย ซึ่งจะเป็นข้อมูลที่แสดงถึงพฤติกรรมผู้ใช้เหตุการณ์ต่างๆที่เกิดขึ้นภายในเครือข่าย ระบบอินเตอร์เน็ตของมหาวิทยาลัย จากการศึกษารูปแบบการใช้งานข้อมูลเบื้องต้นพบว่า การค้นหาจะเริ่มจากการกำหนดช่วงระยะพิสัยเวลาที่ต้องการจะตรวจสอบ โดยการทำการค้นหาแบบสอบถามระยะพิสัยเมื่อได้ข้อมูลในช่วงระยะเวลามาก็จะทำการกำหนดเงื่อนไขอื่นๆที่จะใช้ในการค้นหาเพิ่มเติมอาทิเช่น ในกรณีที่ต้องการพิจารณาผู้ใช้งานที่มีพฤติกรรมการใช้งานที่ผิดปกติบางอย่าง เช่นการเข้าใช้งานพอร์ทหรือการโจมตีระบบ โดยหลังจากที่ทำการกำหนดช่วงระยะเวลาเพื่อทำการค้นหาข้อมูลเหตุการณ์ที่เกิดขึ้นในช่วงระยะเวลานั้นแล้ว ผู้ดูแลระบบจะทำการระบุไอพีที่เกี่ยวข้อง เพื่อที่ผู้ดูแลระบบจะสามารถทำการค้นหาข้อมูลที่เกี่ยวข้องเช่น ชื่อผู้ใช้งาน รวมถึงเวลาเข้าและออกระบบต่อไป

จะสังเกตได้ว่าการสอบถามข้อมูลบันทึกเพื่อนำมาวิเคราะห์มีหลายขั้นตอนและมีการใช้การค้นหาหลายรูปแบบโดยเฉพาะการค้นหาข้อมูลประเภทระยะพิสัยเวลาและการค้นหาข้อมูลรายละเอียดเหตุการณ์ ซึ่งทำให้การอาศัยความเร็วในการค้นหาข้อมูลบางประเภทเป็นไปได้ยาก ในกรณีที่ต้องการข้อมูลที่ไม่ได้ถูกกำหนดอยู่ใน ชาร์ตคีย์และผลกระทบต่อประสิทธิภาพการค้นหาข้อมูลจะยิ่งมากขึ้นเมื่อปริมาณข้อมูลมีขนาดที่ใหญ่มากขึ้น

3.2 **การทำชาร์ตดิ้งกับข้อมูลประเภทบันทึกเหตุการณ์**

ปริมาณโอเปอเรชั่นของข้อมูลบันทึกเหตุการณ์นั้นสามารถแปลผันได้ตามเวลาซึ่งจะขึ้นกับคำขอที่ส่งมาจากไคลเอนต์ในระยะเวลานั้นๆ ถ้าภาระที่เกิดขึ้นในระบบจากการโอเปอเรชั่นเหล่านี้ไม่ถูกกระจายออกไปอย่างเหมาะสม ก็จะมีโอกาสที่ปัญหาแบบฮอตสปอตจะเกิดขึ้นโดยเฉพาะช่วงเวลาที่มีปริมาณโอเปอเรชั่นบันทึกข้อมูลที่สูง โดยเฉพาะเมื่อข้อมูลใหม่นั่นถูกรวมไว้ที่ชาร์ตเครื่องเดียว ปัญหาฮอตสปอตสามารถที่จะส่งผลร้ายแรงต่อสภาพโดยรวมของระบบได้เนื่องจากว่าปัญหาดังกล่าวจะทำให้โอเปอเรชั่นต่างๆใช้ทรัพยากรเครื่องมาเกินความจำเป็นและทำให้ประสิทธิภาพการทำงานตกลงอย่างมากเมื่อทรัพยากรระบบถูกนำไปใช้งานจนถึงขีดจำกัด (12) เพื่อที่จะทำความเข้าใจถึงผลกระทบของภาระงานที่เข้ามาในระบบและการท่านเหล่านั้นทำให้เกิดปัญหาฮอตสปอต จะต้องทราบถึงความแตกต่างในด้านประสิทธิภาพระหว่างแผนการรับมือข้อมูลขนาดใหญ่แบบต่างๆ

เมื่อข้อมูลบันทึกเหตุการณ์ถูกจัดเก็บลงไปในระบบแล้วการสืบค้นข้อมูลแบบพิสัย เป็นวิธีที่จำอย่างยิ่งในการนำข้อมูลบันทึกเหตุการณ์ออกมาวิเคราะห์ ซึ่งหมายความว่าปริมาณเอกสารที่ได้รับแต่ละครั้งจากการสืบค้นจะมาจากความกว้างของพิสัยที่ใช้ในการสืบค้นซึ่งจะครอบคลุมชิ้นส่วนข้อมูลเป็นจำนวนมาก และ ทำให้การประมวลผลเป็นไปได้ช้าลงด้วย เนื่องจากว่ามองโกดีบีมีระบบการประมวลแบบคู่ขนานทำให้การกระจายข้อมูลออกไปอย่างทัดเทียมกันทุกชาร์ตจะจะเป็นแนวทางที่เหมาะสมที่สุดแต่ยังไงก็ตามความเร็วในการประมวลผลของการสืบค้นแต่ละครั้งนั้นไม่สามารถรับประกันได้ว่าจะเท่าเทียมกันทุกชาร์ตแม้ว่าปริมาณของเอกสารที่ถูกส่งกลับ มาจะมีจำนวนเท่ากัน เนื่องจากความแตกต่างระหว่าง การใช้งานทรัพยากรของระบบอย่างเช่นซีพียูเมื่อข้อมูลเข้ามาเป็นปริมาณที่มากขั้นตอนการเขียนข้อมูลจะใช้ทรัพยากรซีพียูเป็นปริมาณมากตามไปด้วย

3.3 **ข้อดีและข้อเสียจากแผนการกระจายข้อมูลบนมองโกดีบี**

จากการศึกษาพฤติกรรมการทำงานของระบบมองโกดีบีที่ใช้แผนการกระจายทั้งแบบพิสัยและแบบแฮช ทำให้พบว่าทั้งข้อดีและข้อเสียมาจากกลไกการทำงานของทั้งสองวิธี ส่งผลให้แผนการกระจายข้อมูลทั้งสองยังไม่สามารถตอบโจทย์ได้อย่างเต็มที่นักเมื่อระบบมีลักษณะข้อมูลที่จำเป็นต้องใช้จุดเด่นของทั้งสองวิธี

การกระจายข้อมูลแบบพิสัยจะเหมาะสมกับกรณีที่ต้องการจะค้นหาข้อมูลที่เฉพาะเจาะจงซึ่งการสืบค้นข้อมูลบนมองโกดีบีอย่างมีประสิทธิภาพนั้นจะขึ้นกับสองหลักคือ ข้อมูลนั้นถูกใช้งานบ่อยครั้งทำให้แคชยังคงอยู่บนหน่วยความจำ และ ข้อมูลนั้นสามารถมองเห็นชาร์ตคีย์ได้ซึ่งรวมถึง

ระยะพิสัยข้อมูลที่ปรากฏบนชาร์ตคีย์ด้วยทำให้เข้ากันได้ดีกับลักษณะงานของข้อมูลบันทึกเหตุการณ์ที่จำเป็นต้องใช้การค้นหาข้อมูลด้วยระยะพิสัย แต่เนื่องจากว่าข้อมูลไม่ได้ถูกกระจายออกไปอย่างทัดเทียมทำให้การค้นหาแต่ละครั้งไม่สามารถใช้ความสามารถด้านการคำนวณแบบคู่ขนานได้ดีนักและยังรวมถึงในกรณีที่ได้รับผลกระทบจากฮอตสปอตโดยเฉพาะพื้นที่ๆมีข้อมูลเป้าหมายเก็บอยู่เป็นปริมาณมาก การกระจายข้อมูลแบบพิสัยยังก่อให้เกิดปัญหาคอขวดอีกด้วยเนื่องจากข้อมูลจะถูกใส่เข้าไปที่จุดเดียวเสมอทำให้ชาร์ตดังกล่าวแบกรับภาระหนักจนกว่าชิ้นส่วนข้อมูลที่ทำหน้าที่รับข้อมูลใหม่จะถูกย้ายไปชาร์ตอื่น

การกระจายข้อมูลแบบแฮชสามารถใช้ความสามารถของมองโกดีบีในด้านการกระจายข้อมูลได้ดีกว่าแบบพิสัยอย่างมาก ข้อมูลจะถูกกระจายออกไปอย่างทัดเทียมทำให้สามารถใช้ความสามารถด้านการคำนวณแบบคู่ขนานได้อย่างเต็มที่ การที่ระบบสร้างชิ้นส่วนข้อมูลขึ้นมาก่อนเพื่อใช้เป็นตัวบอกตำแหน่งที่ข้อมูลใหม่จะถูกจัดเก็บ ทำให้การแตกตัวของชิ้นส่วนข้อมูลเกิดได้น้อยไม่เกิดภาระเพิ่มเติมจากกลไกการแตกชิ้นส่วน แต่เนื่องจากชาร์ตคีย์ถูกทำการเข้ารหัสแฮชไว้ทำให้การค้นหาแต่ละครั้งจะจัดอยู่ในรูปแบบกราดตรวจ ซึ่งจะทำการกราดตรวจข้อมูลทั้งหมดที่มีทุกครั้งก่อนที่จะส่งผลลัพท์ออกมา ทำให้ใช้เวลาในการค้นหานานกว่าปกติ ซึ่งปัญหานี้สามารถแก้ได้ด้วยการสร้างดัชนีให้กับข้อมูลทำให้ประสิทธิภาพการค้นหาของแฮชหลังผ่านกระบวนการทำดัชนีแล้วมีความเร็วที่ค่อนข้างมากจากการคำนวณแบบคู่ขนาน แต่ก็ต้องแลกกับการที่ระบบจะต้องทำการเก็บและอัพเดทดัชนีทุกครั้งที่มีข้อมูลใหม่เพิ่มเติมด้วยรวมถึงเนื้อที่ที่จะถูกใช้ในการเก็บค่าดัชนี ผลกระทบจากฮอตสปอตในแผนการนี้มีน้อยกว่าแบบพิสัยแต่เนื่องจากการค้นหาแต่ละครั้งจะครอบคลุมทุกชาร์ตทำให้การค้นหาข้อมูลจะหลีกเลี่ยงจุดที่เป็นฮอตสปอตไม่ได้

จะสังเกตได้ว่าทั้งสองวิธีต่างก็มีสิ่งที่จะทำให้ระบบประเภทบันทึกเหตุการณ์สามารถทำงานได้เต็มประสิทธิภาพยิ่งขึ้น การสละข้อใดข้อหนึ่งไปจะทำให้เกิดปัญหาที่ส่งผลกระทบต่อการทำงานของระบบได้ ซึ่งจุดดี จุดเสียของทั้งสองแผนการได้ถูกสรุปลงมาในตารางที่ 4 ดังนี้

แผนการกระจายข้อมูล	ข้อดี	ข้อเสีย
แบบพิสัย	1. สามารถค้นหาข้อมูลจากชาร์ตคีย์ได้ 2. การค้นหาแบบพิสัยนั่นจำเป็นสำหรับข้อมูลบันทึกเหตุการณ์ 3. สามารถทำการเคลื่อนย้ายจัดเรียงชิ้นส่วนได้เอง	1. ข้อมูลไม่ได้ถูกกระจายอย่างเต็มที่ทำให้ไม่สามารถใช้งาน 2. เกิดปัญหาคอขวดจากลักษณะการใส่ข้อมูล 3. ผลกระทบจากฮอตสปอตสูง
แบบแฮช	1. ข้อมูลถูกกระจายอย่างเท่าเทียมที่สุด 2. สามารถใส่เอกสารได้พร้อมๆกันทุกชาร์ต 3. สามารถใช้ประโยชน์จากการคำนวณแบบคู่ขนานได้เต็มที่ทำให้การค้นหามีความเร็วสูงในกรณีที่ทำดัชนี	1. ต้องทำดัชนีข้อมูลเพื่อให้ระบบสามารถค้นหาแบบพิสัยได้ 2. การค้นหาแต่ละครั้งจะไม่สามารถหลีกเลี่ยง ตำแหน่งที่เกิดฮอตสปอตได้ 3. ข้อมูลถูกใส่รหัสและกระจายออกไป ทำให้ไม่สามารถที่จะรู้ถึงตำแหน่งชิ้นส่วนที่ เก็บข้อมูลที่ต้องการได้

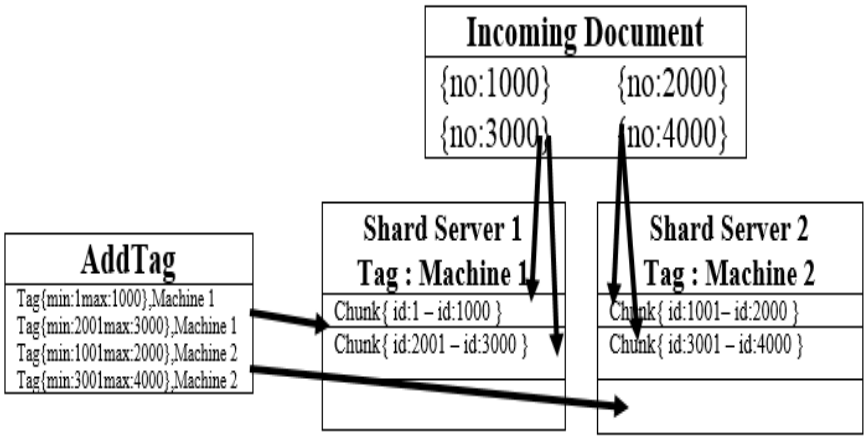
ตารางที่ 4 ข้อดีข้อเสียของแผนการกระจายข้อมูล

3.4 ระบบชาดดิ้งที่ตระหนักถึงป้ายบอกข้อมูล (Tag Aware Shading)

จากปัญหาที่ได้เจอในระหว่างการศึกษาแผนการกระจายข้อมูล ทำให้พบว่า การที่จะแก้ไขปัญหาเหล่านี้จำเป็นที่จะต้องเลือกกลไกเพื่อมาสนับสนุนการทำงานของชาร์ตดิ้งและเพื่ออำนวยสะดวกในการจัดสรรข้อมูลให้สามารถใช้ประโยชน์จากความสามารถของมองโกดีบีได้เต็มที่ เพื่อเป้าหมายนี้จึงได้ประยุกต์ใช้ระบบป้ายบอกข้อมูลกับการทำชาร์ตดิ้งแบบปกติ เดิมทีกลไกดังกล่าวถูกออกแบบมาเพื่อการจำแนกและจัดเก็บให้กับข้อมูลที่สามารถแยกลักษณะได้ด้วยรายละเอียดที่เด่นชัด โดยการการจับกลุ่มชิ้นส่วนข้อมูลเข้ากับชาร์ตใดชาร์ตหนึ่งตามรายละเอียดข้อมูลที่ทำการติดป้ายไว้ ระบบสามารถที่จะติดป้ายให้กับระยะพิสัยใหม่หรือชิ้นส่วนข้อมูลที่มีอยู่แล้วได้และผูกเข้ากับชาร์ตที่ติดป้ายที่เหมือนกันชิ้นส่วนข้อมูลที่มีป้ายกำหนดไว้จะเคลื่อนย้ายไปที่ชาร์ตปลายทางที่ติดป้ายเหมือนกัน ถ้าไม่ปรากฏชิ้นส่วนข้อมูลที่ตรงกับระยะที่ติดป้ายใหม่ระบบจะทำการสร้างชิ้นส่วนนั้นขึ้นมาและทำการเคลื่อนย้ายไปยังชาร์ตปลายทาง

จากลักษณะกลไกของระบบติดป้ายข้อมูลข้างต้นทำให้พบว่า ระบบสามารถที่จะสร้างชิ้นส่วนข้อมูลเพื่อรองรับข้อมูลที่จะเข้ามาในอนาคตได้แบบเดียวกันกับการกระจายข้อมูลแบบแฮช ซึ่ง

หมายความว่าถ้าใช้วิธีดังกล่าวในการจัดสรรชิ้นส่วนข้อมูลก่อนจะสามารถลดขั้นตอนการสร้างและเคลื่อนย้ายชิ้นส่วนข้อมูลได้ทำให้ลดระยะเวลาและลดการทำงานลงซึ่งจะเป็นประโยชน์มากในกรณีที่ต้องการจะหลีกเลี่ยงช่วงเวลาที่ระบบทำงานหนัก แต่กระนั้นแล้วก็ต้องคำนึงถึงระยะเวลาที่ระบบจะทำการเคลื่อนย้ายข้อมูลไปยังที่หมายด้วยเนื่องจาก การสร้างและการเคลื่อนย้ายแต่ละครั้งจะใช้เวลานาน ถ้ามีชิ้นส่วนที่รอการเคลื่อนย้ายเป็นจำนวนมาก



รูปที่ 13 ลักษณะการกระจายข้อมูลด้วยป้ายบอกข้อมูล

ในการที่จะประยุกต์ใช้ระบบติดป้ายข้อมูลให้เข้ากับข้อมูลบันทึกเหตุการณ์ ในงานวิจัยนี้ได้ใช้ตราเวลาเป็นชาร์ตคีย์และทำการเปิดระบบชาร์ตดิ้งด้วยแผนการกระจายแบบพิสัยเพื่อที่จะทำให้เข้ากับลักษณะการสืบค้นข้อมูลบันทึกเหตุการณ์ ระยะพิสัยของตราเวลาจะถูกคำนวณขึ้นมาให้อยู่ในระดับที่ชิ้นส่วนไม่ใหญ่หรือเล็กเกินไป เพื่อให้ผลกระทบจากฮอตสปอตมีน้อยที่สุดในขณะที่ยังคงไว้ซึ่งความสามารถในการทำการค้นหาข้อมูลแบบพิสัยและสามารถใช้ประโยชน์จากการคำนวณแบบคู่ขนานให้มากที่สุด ชิ้นส่วนข้อมูลจะถูกติดป้ายและกระจายลงไปในชาร์ตอย่างเท่าเทียมกัน และเพื่อเป็นการลดภาระการทำงานที่เกิดขึ้นจากขั้นตอนการแยกตัวของชิ้นส่วนข้อมูล จึงต้องมีการสร้างชิ้นส่วนข้อมูลเพิ่มขึ้นมาก่อนที่จะรับข้อมูลที่กำลังจะเข้าใหม่

3.5 การปรับชาร์ตให้เหมาะสมกับโอเปอเรชั่นของข้อมูลบันทึกเหตุการณ์

เพื่อเป็นการให้ระบบสามารถที่จะกระจายภาระงานออกไปในกลุ่มชาร์ตได้อย่างเท่าเทียมที่สุด ในงานวิจัยนี้จึงได้นำเสนออัลกอริทึมในการที่จะกำหนดระยะพิสัยของแต่ละป้ายข้อมูล โดยที่เป้าหมายหลักคือการทำให้การกระจายข้อมูลสามารถที่จะเข้ากันกับลักษณะรูปแบบการสืบค้นข้อมูล

เพื่อที่จะให้สามารถใช้งานการคำนวณแบบคู่ขนานได้อย่างเต็มประสิทธิภาพที่สุดและลดโอกาสและระยะการสอบถามข้อมูลจะกระทบชาร์ตที่เป็นฮอตสปอต เพื่อให้แต่ละชิ้นส่วนข้อมูลมีขนาดที่สมเหตุสมผล จึงจะต้องใช้ อัตราการใส่ข้อมูลมาใช้ในการคำนวณค่าพิสัยของตราเวลา ซึ่งตัวชิ้นส่วนข้อมูลยังมีข้อจำกัดที่สามารถรองรับเอกสารได้เพียง 250,000 (3) เท่านั้น ไม่ฉะนั้นชิ้นส่วนข้อมูลที่มีขนาดใหญ่เกินข้อกำหนดจะไม่สามารถที่จะเคลื่อนย้ายได้ดังนั้นแล้วจึงต้องใช้ 250,000 เป็นขีดจำกัดของจำนวนเอกสารบันทึกเหตุการณ์ที่หนึ่งชิ้นส่วนข้อมูลจะมีได้

Algorithm 1 Tag Range Determination Algorithm

1. **let** q be the expected range of query window (in seconds)
2. **let** D be the expected log arrival rate (in log document per second)
3. **Formula** Tag range $T_r = T + t$
4. $$t = \begin{cases} q \ \ if \ \ (q * D \leq 250000) \\ \frac{250000}{D} \ \ otherwise \end{cases}$$
5. **return** t

อัลกอริทึม 1 อัลกอริทึมในการกำหนดระยะพิสัยของป้ายข้อมูล

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

แนวคิดหลักสำหรับการคำนวณหาระยะพิสัยเวลา คือเพื่อที่จะทำให้ปริมาณเอกสารที่เข้ามาในระบบสามารถที่จะเข้ากันได้พอดีกับระยะเวลาที่ใช้สอบถามข้อมูลภายในชิ้นส่วนข้อมูลจำนวนหนึ่งชิ้น อัลกอริทึมข้างต้นสามารถอธิบายได้ดังนี้

เมื่อ q เป็น การคาดการณ์ระยะเวลาที่ใช้ในการสอบถามข้อมูล โดยจะมีหน่วยเป็นวินาที

D เป็น การคาดการณ์อัตราการเข้ามาของข้อมูลบันทึกเหตุการณ์ โดย คิดเป็น เอกสารต่อวินาที

Tr เป็นเป็น ระยะพิสัยเวลาที่จะถูกนำไปติดป้ายข้อมูลซึ่งจะเป็นผลลัพจาก เวลาที่จะเริ่มทำการติด ป้าย T บวกกับ ระยะพิสัยเวลา t

ในบรรทัดที่ 4 ได้นำ q ไปคูณกับ Dเพื่อให้ได้ปริมาณเอกสารที่คาดว่าจะอยู่ในช่วงระยะเวลา q ถ้าเอกสารมีจำนวนต่ำกว่า 250,000 ให้ t เป็น q ในขณะที่ ถ้ากรณีที่เอกสารที่คาดเดาในระยะเวลาดังกล่าวมีจำนวนที่เกิน250,000ทำให้ไม่สามารถนำเก็บในชิ้นส่วนเดียวได้ จะทำการนำ 250,000 มา

หารกับD เพื่อให้ได้ระยะเวลาที่เล็กลงมาสำหรับการแยกข้อมูลนั้นลงไปเก็บตัวอย่างเช่น เมื่ออัตราข้อมูลเข้ามีทั้งหมด 1000ต่อวินาทีเป็นเวลา15นาทีหรือ900วินาทีซึ่งจะทำให้เอกสารทั้งหมดมีจำนวนทั้งสิ้น 900,000 ซึ่งผลลัพจากการนำ 250,000/1000 ทำให้ได้ 250 วินาที ต่อ 1 ชิ้นส่วนข้อมูล

และเพื่อเป็นการเพิ่มประสิทธิภาพการทำงาน ในงานดังกล่าวยังรวมกลไกที่จะทำการจัดสรรและสร้างชิ้นส่วนข้อมูลขึ้นมาก่อนเพื่อให้ระบบมีเวลามากพอที่จะใช้ในการเคลื่อนย้ายชิ้นส่วนข้อมูล โดยการกำหนด Nprecale เป็นค่าเริ่มต้นปริมาณชิ้นส่วนข้อมูลที่จะต้องสร้าง เมื่อสร้างเสร็จแล้วจะทำการติดป้ายและกระจายออกไปเพื่อให้สามารถใช้งานการคำนวณแบบคู่ขนานได้เต็มที่เมื่อข้อมูลถูกใส่จนถึง Nthreshold ที่กำหนดไว้ ระบบจึงจะเริ่มทำการคำนวณเพิ่มชิ้นส่วนข้อมูลกลุ่มใหม่เข้าไป

แต่ก็ยังมีปัญหาเกี่ยวกับด้านการคาดเดาอัตราข้อมูลขาเข้า ซึ่งในความเป็นจริงแล้วอัตราขาเข้าจะไม่กำหนดไว้ตายตัวและจะมีการเปลี่ยนแปลงอยู่เรื่อยๆอย่างเช่นระบบจะมีข้อมูลขาเข้าที่มากในช่วงระยะเวลากลางวันที่มีผู้เข้าใช้ระบบเป็นปริมาณมากกว่าช่วงกลางคืน ดังนั้งแล้ว การเปลี่ยนแปลงในอัตราข้อมูลขาเข้าจะส่งผลให้ประสิทธิภารการทำงานมีการเปลี่ยนแปลงตามไปด้วย การเปลี่ยนระบบให้สามรถตอบสนองได้ไวมากขึ้นสามารถทำได้โดยการลดขนาด Nprecale กับ Nthreshold เพื่อให้ระบบทำการคำนวณเวลาในการเพิ่มซึ่งก็จะส่งผลให้ประสิทธิภาพการทำงานลดลงตามไปด้วยจากการที่ ระบบมีการสร้างและเคลื่อนย้ายข้อมูลที่บ่อยขึ้น

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 4
การทดลองและสรุปผลงานวิจัย

4.1 การทดสอบประสิทธิภาพ

4.1.1 การเตรียมระบบที่ใช้ในการทดลอง

ระบบที่ใช้ในการทำการวิจัยนั้นจะอ้างอิงจากเกณฑมาตรฐานของมองโกดีบีคือประกอบไปด้วยชาร์ตจำนวน 4เครื่องและแต่ละเครื่องใช้ระบบปฏิบัติการณ์ Ubuntu LTS 12.04 โดยแต่ละเครื่องจะมี CPU รุ่น Intel(R) Xeon(R) CPU E5-2670 0@ 2.60GHz และได้รับการติดตั้งมองโกดีบีเวอชั่นล่าสุด

ระบบมองโกดีบีที่จะใช้ทดลองจะแบ่งออกเป็นสามรูปแบบขึ้นกับแบบแผนการกระจายข้อมูล ได้แก่ การทำชาร์ตดิ้งแบบพิสัย การทำชาร์ตดิ้งแบบแฮช และ แบบที่นำเสนอในงานวิจัยการทำชาร์ตดิ้งด้วยป้ายข้อมูล ซึ่งต้องมีการล้างข้อมูลและทำใหม่ทุกครั้งที่มีการเปลี่ยรูปแบบการกระจายข้อมูล โดยในกรณีของแผนการรูปแบบที่เสนอนั้นจำเป็นที่จะต้องทำการปิดระบบปรับสมดุลอัตโนมัติของโมโกดีบีด้วยซึ่งสามารถออกคำสั่ง sh.stopBalancer() ผ่านเราเตอร์เพื่อป้องกันไม่ให้ชิ้นส่วนข้อมูลเกิดการเคลื่อนย้ายไปแบบแผนการที่วางไว้

ในการทดลองแนวทางการกระจายข้อมูลแบบติดป้ายข้อมูลนั้นร

4.1.2 ระบบจำลองข้อมูลบันทึกเหตุการณ์

ข้อมูลบันทึกเหตุการณ์ที่นำมาใช้ในงานวิจัยนี้ได้รับการอ้างอิงโครงสร้างมาจากข้อมูลบันทึกเหตุการณ์จากระบบไอทีจุฬาลงกรณ์มหาวิทยาลัย ระบบจำลองข้อมูลพัฒนาโดยใช้ภาษา JavaScript โดยระบบจะมีการกำหนดตราเวลาที่คงที่เพื่อให้สะดวกต่อการจำลองเหตุการณ์จริง ข้อมูลในแต่ละหนึ่งวัน ก่อนที่จะทำการสุ่มค่าเพื่อจำลองข้อมูลภายในเอกสารบันทึกเหตุการณ์หนึ่งเอกสาร

ในการใส่ข้อมูลทุกครั้งระบบจะต้องทำการติดต่อกับเราเตอร์ของมองโกดีบีและเชื่อมต่อเข้ากับฐานข้อมูลที่สร้างไว้เพื่อรองรับข้อมูลบันทึกเหตุการณ์ ข้อมูลแต่ละส่วนของบันทึกเหตุการณ์จะถูกผูกเข้ากับฟิลด์ที่กำหนดไว้ใน ก่อนที่จะทำการใส่ข้อมูลลงไปในระบบ

ข้อมูลดั้งเดิม	แปลงเป็นเอกสาร
2016-10-05T00:00:00.00 acs1121-cen59-01 INFO Max sessions policy passed...	{ timestamp : 2016-10-05T00:00:00.00, Machine : "acs1121-cen59-01 ", type : "INFO", details : "Max sessions policy passed..." }

รูปที่ 14 ตัวอย่างข้อมูลบันทึกเหตุการณ์ที่ถูกแปลงให้เป็นเอกสารมองโกดีบี

4.1.3 เคสที่ใช้ในการทดลอง

งานวิจัยนี้จะแบ่งกลุ่มการทดลองออกเป็นสองกลุ่มหลักได้แก่การประเมินประสิทธิภาพการทำของของการใส่ข้อมูล และ การอ่านข้อมูล โดยการทดลองทั้งสองกลุ่มจะต้องทำการทดลองกับแผนการกระจายข้อมูลทั้งสามแบบได้แก่ การกระจายข้อมูลแบบพิสัย, การกระจายข้อมูลแบบแฮช และ วิธีติดป้ายข้อมูล เพื่อให้ผลการทดลองมีความเที่ยงตรงมากที่สุด จึงได้มีการทำการการทดลองซ้ำสิบครั้งก่อนที่จะนำมาหาค่าเฉลี่ย รายละเอียดเคสที่จะต้องทดลองมีดังต่อไปนี้

ประสิทธิภาพการใส่ข้อมูล (Write Performance)

เพื่อเป็นการหาขีดจำกัดของประสิทธิภาพการทำงานด้านการใส่ข้อมูล จึงต้องมีการทำการทดลองด้วยการกำหนดให้แต่ละแนวทางต้องใส่เอกสารลงไปทั้งหมดเป็นปริมาณทั้งสิ้น 1,000,000 เอกสาร เพื่อที่จะวัดความแตกต่างของเวลาที่ใช้ในการประมวลข้อมูลทั้งหมดสำหรับแผนการกระจายข้อมูลแบบต่างๆ

ประสิทธิภาพการอ่านข้อมูล (Read Performance)

เพื่อให้การวัดประสิทธิภาพการทำงานของรูปแบบการกระจายข้อมูลให้เข้ากับสถานการณ์จริง ในการทดลองนี้จะแบ่งออกมาเป็นกรณีย่อยสามกรณีได้แก่ ไม่มีการใส่ข้อมูล (no write) มีใส่ข้อมูลปานกลาง (normal write) และ มีการใส่ข้อมูลอย่างหนัก (heavy write) เพื่อเป็นการจำลองสภาพความพร้อมในด้านทรัพยากรของระบบ ทั้งสามกรณีมีรายละเอียดดังนี้

ไม่มีการใส่ข้อมูล (no write) หมายความว่าเป็นกรณีที่ระบบจะไม่มีการรับเอกสารใหม่ใดๆ ทั้งสิ้นทำให้มีทรัพยากรเครื่องว่างใช้งานมากและไม่เกิดปัญหาใดๆแทรกซ้อน

มีการใส่ข้อมูลปานกลาง (normal write) เป็นกรณีที่จำลองระบบในสถานการณ์ปกติ ที่มีทรัพยากรเพียงพอต่อทั้งการใส่เอกสารและอ่านข้อมูล เอกสารที่ถูกใส่เข้าไปในการทดลองนี้จะมีปริมาณอยู่ที่ 1000 เอกสารต่อต่อวินาที

มีการใส่ข้อมูลอย่างหนัก (heavy write) สถานการณ์ที่เครื่องทำงานอย่างหนักจนเกิดฮอตสปอตขึ้น ทรัพยากรเครื่องจะถูกนำไปใช้กับการเขียนข้อมูลจนส่งผลกระทบต่อการสอบถามข้อมูลอย่างมาก ปริมาณเอกสารที่ใส่เข้าระบบจะอยู่ที่ 10,000 เอกสารต่อวินาที นอกเหนือจากสถานการต่างๆในการทดลองแล้ว ยังต้องทำการล้างแคชในระบบทุกครั้งเนื่องจากมองโกดีบีจะจำข้อความและฐานข้อมูลที่ถูกค้นหาไว้ในหน่วยความจำเสมอทำให้การสอบถามข้อมูลครั้งต่อไปมีจะมีความเร็วที่สูงมากทำให้ไม่สามารถวัดประสิทธิภาพได้

4.2 การรวบรวมข้อมูลเพื่อประมวลผล

4.2.1 การรวมรวบข้อมูลโอเปอเรชั่น

เพื่อให้สามารถทำการการรวบรวมข้อมูลเกี่ยวกับโอเปอเรชั่นที่เกิดขึ้นเพื่อที่จะใช้ในการวิเคราะห์ภาระงานที่เข้ามาในระบบนั้น เบื้องต้นมองโกดีบีมีระบบ oplog (operation log) เป็นเครื่องมือในการเฝ้าสังเกตุพฤติกรรมของโอเปอเรชั่นที่เกิดขึ้นในระบบ โดยสามารถกำหนดรูปแบบความถี่ในการบันทึก เช่นทุกๆ5วินาทีจากตัวอย่างข้อมูลโดยระบบจะบันทึกเฉพาะเหตุการณ์ที่เกิดขึ้นในระยะเวลานั้นทำให้การที่จะนำไปใช้ จำเป็นที่จะต้องกำหนดการบันทึกค่อนข้างถี่ รูปที่ 15 เป็นตัวอย่างข้อมูลที่ได้จากฟังก์ชั่น oplog

```
connected to: localhost:27017
insert  query update delete    db         netIn   netOut   conn    time
   918     *0     *0     *0   Collection1  117k    52k     2   13:09:55
  1805     *0     *0     *0   Collection1  231k   101k     2   13:10:00
  2614      1     *0     *0   Collection1  334k   147k     2   13:10:05
  2606      2     *0     *0   Collection2  333k   146k     2   13:10:10
  2604      2     *0     *0   Collection2  333k   146k     3   13:10:15
  1011     *0     *0     *0   Collection2  129k    57k     3   13:10:20
```

รูปที่ 15 ตัวอย่างข้อมูลที่ได้จากบันทึก oplog

แต่เนื่องจากว่าฟังก์ชั่น oplog นั้นถูกพัฒนาสำเร็จรูปมาจากมองโกดีบีทำให้การเข้าไปปรับแก้ไขเพื่อประยุกต์ใช้ในการรวบรวมข้อมูลเป็นไปได้ด้วยความลำบาก ทำให้จำเป็นที่จะต้องหันไปเลือกระบบอื่นซึ่งในที่นี่ได้เลือกระบบ Ganglia ซึ่งเป็นระบบโอเพนซอร์ส(open source)ที่ถูกพัฒนามาเพื่อการเฝ้าสังเกตุระบบขนาดใหญ่ที่มีลักษณะแบบกระจายโดยเฉพาะระบบดังกล่าวรอบรับระบบปฏิบัติการต่างๆ ทำให้สะดวกในการใช้งานนำเข้าใช้งานโดยใน Ganglia ยังได้มีส่วนขยายที่ถูกพัฒนาขึ้นมาเพิ่มเติมเพื่อให้สามารถใช้งานกับระบบอื่นๆรวมถึงมองโกดีบีได้ด้วย ซึ่งส่วนขยายดังกล่าวถูกพัฒนาโดยใช้ภาษา Python ทำให้สามารถที่จะปรับแต่งให้เข้ากับงานวิจัยนี้

ได้ง่าย ซึ่งหลังจากได้ข้อมูลที่ต้องการแล้วผลที่ได้จากจุดนี้จะนำไปใช้ในการประเมินผลการทดลอง สำหรับประสิทธิภาพการใส่ข้อมูล ทำให้ค่าที่ต้องการเป็นหลักคือ insert หรือ opcounters_insert ในตัวอย่างข้อมูลที่รวบรวมได้จากส่วนขยายมองโกดีบีดังรูปที่ 16

```
09:53:58 mongodb_opcounters_insert 25610.131293
09:53:58 mongodb_opcounters_query 0.000000
09:53:58 mongodb_opcounters_update 0.000000
09:53:58 mongodb_opcounters_delete 0.000000
09:53:58 mongodb_opcounters_getmore 0.000000
09:53:58 mongodb_opcounters_command 3.274115
09:53:58 mongodb_backgroundFlushing_flushes 0.000000
09:53:58 mongodb_mem_mapped 0
09:53:58 mongodb_mem_virtual 149
09:53:58 mongodb_mem_resident 9
09:53:58 mongodb_extra_info_page_faults 0.000000
09:53:58 mongodb_globalLock_ratio 0.000000
09:53:58 mongodb_indexCounters_btree_miss_ratio 0.000000
09:53:58 mongodb_globalLock_currentQueue_total 0
09:53:58 mongodb_globalLock_currentQueue_readers 0
09:53:58 mongodb_globalLock_currentQueue_writers 0
09:53:58 mongodb_globalLock_activeClients_total 0
09:53:58 mongodb_globalLock_activeClients_readers 0
09:53:58 mongodb_globalLock_activeClients_writers 0
09:53:58 mongodb_connections_current 4
09:53:58 mongodb_connections_current_ratio 0.490798
09:53:58 mongodb_slave_delay 0
09:53:58 mongodb_asserts_total 0.000000
```

รูปที่ 16 ตัวอย่างข้อมูลที่ได้จากส่วนขยายมองโกดีบีบนGanglia

4.2.2 การรวมรวบข้อมูลการใช้ประโยชน์ซีพียู

ขั้นตอนที่สองเพื่อให้สามารถศึกษาประสิทธิภาพการทำงานและการใช้ประโยชน์จากซีพียูของแต่ละชาร์ตจึงได้มีการเขียนคำสั่งเพื่อให้ระบบทำการคอยสังเกตุและบันทึก

top -bc -d 1 | grep 'mongo' | awk '{print $1 " " $9 " " $10 " " $12}' >> cpu.txt

คำสั่งดังกล่าวสามารถอธิบายได้ดังนี้ top -bc เป็นคำสั่งที่ใช้แสดงถึงรายการของโดยให้ข้อมูลแสดงรายละเอียดที่เกิดขึ้นและอยู่ และพารามิเตอร์ -d 1 เป็นการกำหนดให้ระบบมีความหน่วง 1วินาทีก่อนที่จะทำการเก็บค่าต่อไป โดยจะทำการเลือกเฉพาะระบบที่เกี่ยวข้องกับมองโกดีบีเท่านั้นด้วยการ grep ‘mongo’และท้ายที่สุดเพื่อกรองข้อมูลสำหรับความสะดวกในการประเมินผล จึงมีการเลือกเฉพาะข้อมูลเกี่ยวข้องกับกระบวนการทำงานซึ่งในทีนี้จะประกอบไป

ด้วย รหัสของกระบวนกวน (pid) คีซีพียู CPU(%) เวลาที่ระบบทำงาน (runtime) และ กรรมวิธี (process) ดังที่จะเห็นได้ในรูป 17

pid	CPU(%)	Runtime	process
12145	0	00:06.1	mongod
12145	0	00:06.1	mongod
12145	26	00:06.4	mongod
12145	100	00:07.4	mongod
12145	93	00:08.3	mongod
12145	100	00:09.3	mongod
12145	92	00:10.2	mongod
12145	98	00:11.2	mongod
12145	89	00:12.1	mongod
12145	72	00:12.8	mongod
12145	98	00:13.8	mongod
12145	87	00:14.7	mongod
12145	69	00:15.4	mongod
12145	61	00:16.0	mongod
12145	76	00:16.8	mongod

รูปที่ 17 ตัวอย่างข้อมูลทรัพยากรซีพียู

4.2.3 การเก็บค่าประสิทธิภาพการสอบถามข้อมูล

การใช้คำสั่งนับเวลาในโปรแกรมต่างๆนั้น จะทำให้ได้แค่ระยะเวลาที่สอบถามข้อมูลเสร็จเพียงอย่างเดียวซึ่งไม่มากพอที่ใช้ในการศึกษาพฤติกรรมของการสอบถามข้อมูล เพื่อให้ได้ข้อมูลการสอบถามแต่ละครั้งอย่างละเอียดที่สุด จึงได้มีการใช้คำสั่ง .explain() ต่อท้ายคำสั่งสอบถามข้อมูล ซึ่งจะออกมาในรูปแบบดังนี้ **db.collection.find().explain()** โดยที่คำสั่งดังกล่าวจะเป็นการทำให้มองโกดีบีโชว์รายละเอียดทั้งหมดที่เกิดขึ้นระหว่างการสอบถามข้อมูล ซึ่งมีส่วนที่สำคัญสำหรับใช้ในการวัดผลดังนี้ จำนวนเอกสารที่ค้นเจอในแต่ละชาร์ตและเวลาที่ใช้ และจำนวนเอกสารทั้งหมดที่ค้นเจอและเวลา โดยตัวอย่างรูปที่1จะเป็นส่วนที่อธิบายถึงรายละเอียดที่การสืบค้นข้อมูลไปตกกระทบแต่ละชาร์ต และรูปที่2จะเป็นรายละเอียดสรุปประสิทธิภาพการค้นหาในรอบนั้นโดยที่ n แทนจำนวนเอกสารที่ค้นหาได้ และ ค่า millis แทนเวลาที่ใช้ในการดำเนินงานมีหน่วยเป็นมิลลิวินาที โดยที่เวลาที่ใช้ในการดำเนินงานในส่วนภาพรวมการทำงั้นนั้นจะมาจากชาร์ตที่ทำงานช้าที่สุดเสมอ

```
"clusteredType" : "ParallelSort",
"shards" : {
        "s1/161.200.135.202:27019" : [
                {
                        "cursor" : "BtreeCursor timestamp_1 reverse",
                        "isMultiKey" : false,
                        "n" : 360734,
                        "nscannedObjects" : 360734,
                        "nscanned" : 360734,
                        "nscannedObjectsAllPlans" : 360734,
                        "nscannedAllPlans" : 360734,
                        "scanAndOrder" : false,
                        "indexOnly" : false,
                        "nYields" : 2111,
                        "nChunkSkips" : 0,
                        "millis" : 38989,
                        "indexBounds" : {
                                "timestamp" : [
                                        [
                                                "2016-06-07T22:35:00:0000",
                                                "2016-06-07T20:35:00:0000"
                                        ]
                                ]
                        },
                        "server" : "hadoov03:27019"
                }
        ],
```


รูปที่ 18 ตัวอย่างผลลัพท์เฉพาะส่วนชาร์ตจากการทำexplain()

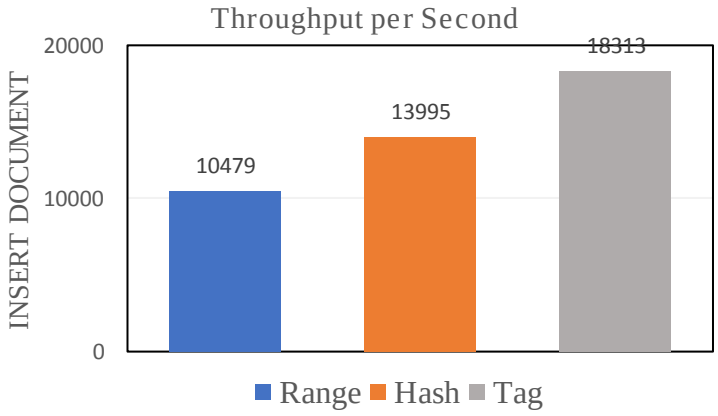

```
"cursor" : "BtreeCursor timestamp_1 reverse",
"n" : 1440000,
"nChunkSkips" : 0,
"nYields" : 8336,
"nscanned" : 1440000,
"nscannedAllPlans" : 1440000,
"nscannedObjects" : 1440000,
"nscannedObjectsAllPlans" : 1440000,
"millisShardTotal" : 178635,
"millisShardAvg" : 44658,
"numQueries" : 4,
"numShards" : 4,
"millis" : 65583
```


รูปที่ 19 ตัวอย่างผลลัพธ์ภาพรวมการสืบค้นข้อมูลจากการทำexplain()

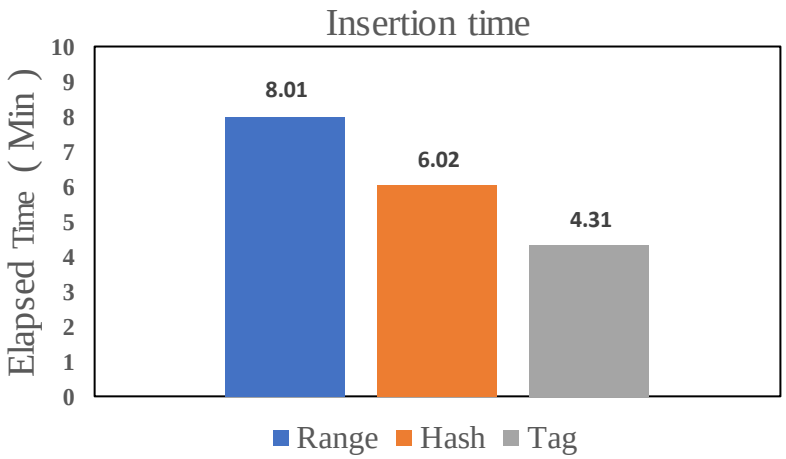
4.3 ประสิทธิภาพของการบันทึกข้อมูล (Write Performance)

ในการทดลองนี้จะเป็นการเปรียบเทียบประสิทธิภาพของการใส่ข้อมูลจากชาร์ตดิ้งที่เลือกได้แก่ แบบพิสัย แบบแฮช และ แบบติดป้ายข้อมูล ผลการทดลองประสิทธิภาพของการใส่ข้อมูลแบ่งออกเป็นสองส่วนโดยแบบแรกรูปที่ 20 จะวัดปริมาณเอกสารที่ระบบสามารถทำได้ต่อวินาที กับแบบที่สองรูปที่ 21 วัดระยะเวลาที่ใช้ในการใส่เอกสารทั้งหมดซึ่งมีหน่วยเป็นนาที จากผลทดลองจะสังเกตุได้ว่าแบบพิสัยจะใช้เวลาในการทำงานช้าที่สุด จากการใส่เอกสารจำนวนทั้งสิ้น 1,000,000 ถ้าเทียบกับวิธีอื่นซึ่งสาเหตุหลักมาจากการที่เอกสารที่เข้าระบบทั้งหมดจะไปรวมอยู่ที่จุดเดียวจนทำให้เกิดปัญหาคอขวดกับประสิทธิภาพการทำงาน ซึ่งเป็นผลมาจากการที่แบบพิสัยจะเกิดขั้นตอนการแตกชิ้นส่วนข้อมูลใหม่ เคลื่อนย้ายข้อมูลที่บ่อยครั้ง ซึ่งสองขั้นตอนนี้จะเป็นขั้นตอนที่เป็นภาระกับระบบที่ค่อนข้างสูงในขณะที่รูปแบบแฮชสามารถทำได้รวดเร็วกว่าพิสัยอย่างมาก

ในขณะที่รูปแบบที่นำเสนอในงานวิจัยนี้มีประสิทธิภาพการทำงานที่ดีกว่าอีกสองแบบ อันเนื่องมาจากสาเหตุหลักสองอย่างคือ หนึ่ง การที่สร้างชิ้นส่วนข้อมูลไว้รองรับข้อมูลใหม่ก่อนโดนมีการคำนวณระยะพิสัยและจำนวนที่แต่ละชิ้นส่วนจะสามารถเก็บได้ ทำให้การแตกชิ้นส่วนและการเคลื่อนย้ายข้อมูลไม่เกิดขึ้นในวิธีนี้ และ สอง การกระจายชิ้นส่วนและจัดสรรแต่แรกในทีนี้จะทำให้ปัญหาฮอตสปอตไม่อยู่ที่ชาร์ตเดียวเป็นระยะเวลานานๆ



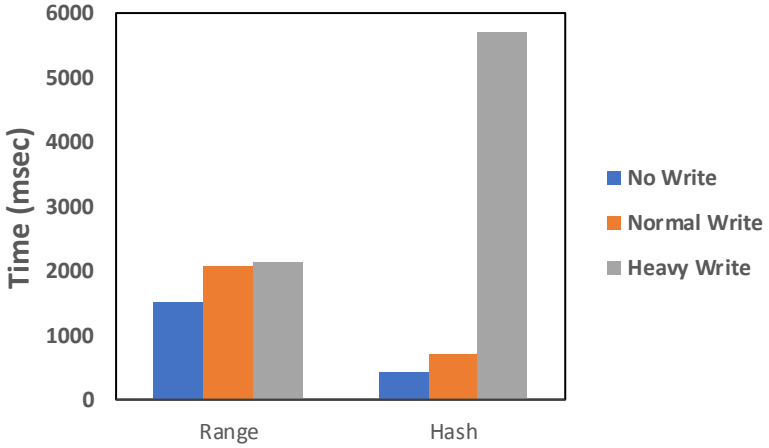
รูปที่ 20 ปริมาณเอกสารที่ประมวลผลได้ต่อวินาที



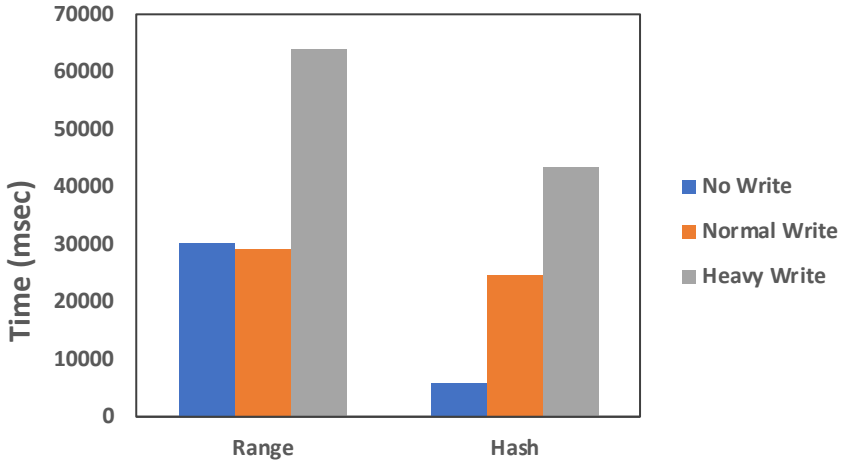
รูปที่ 21 เปรียบเทียบความเร็วของการบันทึกเอกสาร

4.4 **ประสิทธิภาพการสอบถามข้อมูลของการกระจายแบบพิสัยและแฮช**

เพื่อพิสูจน์และทำความเข้าใจถึงความแตกต่างระหว่างประสิทธิภาพการทำงานของระบบแบบพิสัยและแบบแฮช จึงได้มีการทำการทดสอบประสิทธิภาพการสอบถามข้อมูล โดยวัดจากเวลาที่ใช้ในการทำสอบถามข้อมูลแต่ละครั้ง ภายใต้สถานการณ์สามแบบได้แก่ ไม่มีการบันทึกข้อมูล (No Write) มีการบันทึกข้อมูลในภาวะปกติ (Normal Write) และเมื่อมีการบันทึกข้อมูลอย่างหนัก (Heavy Write) โดยข้อมูลบันทึกเหตุการณ์จะถูกจำลองออกมาที่ 17,000,000 บันทึกโดยประมาณ และมีช่วงระยะเวลาในการสืบค้นที่ 5 นาทีกับ 2 ชั่วโมง ซึ่งจะทำการค้นหาซ้ำ 10 ครั้ง ในการค้นหาแต่ละครั้ง จะมีการคืนเอกสารกลับมาที่ 60000 และ 1,440,000 บันทึก ตามลำดับ ก่อนที่จะนำมาคำนวณเวลาเฉลี่ยในการทำเสร็จ และเพื่อให้เกิดความทัดเทียมกัน ระบบกระจายแบบแฮชจะมีการสร้างดัชนีเตรียมไว้ก่อน เพื่อที่จะทำให้สามารถทำการค้นหาแบบระยะพิสัยได้



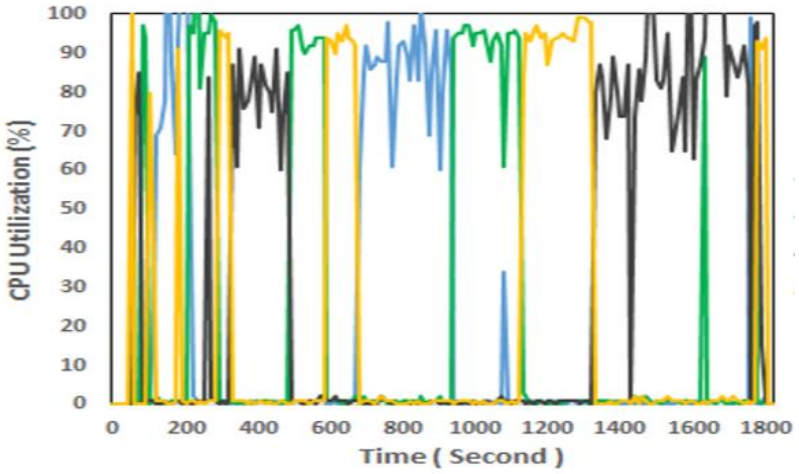
รูปที่ 22 ประสิทธิภาพการสอบถามข้อมูล 5 นาที ของพิสัยและแฮช



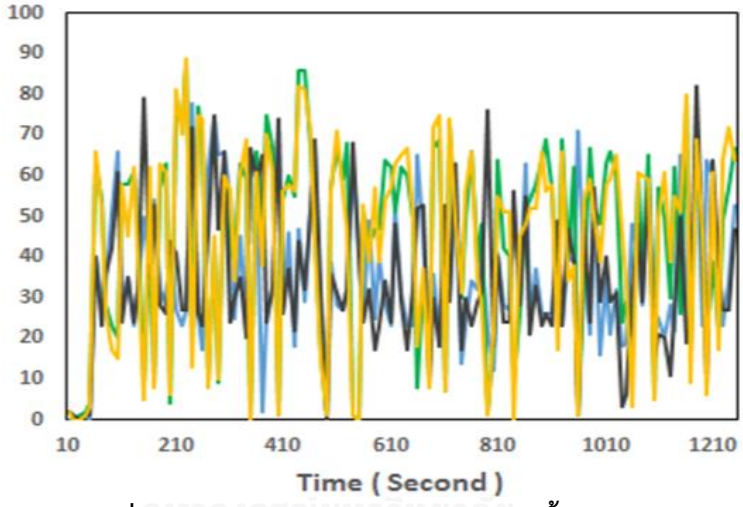
รูปที่ 23 ประสิทธิภาพการสอบถามข้อมูล 2 ขั่วโมง ของพิสัยและแฮช

ในรูปที่ 22 และ รูปที่ 23 จะเป็นการแสดงถึงผลการทดลองของประสิทธิภาพการสอบถามข้อมูลสำหรับการสอบถามแบบ 5 นาทีและ 2 ชั่วโมงตามลำดับ จากผลการทดลองจะเห็นว่า การกระจายแบบแฮชนั้น สามารถทำได้ดีกว่าแบบพิสัยเป็นอย่างมาก โดยเฉพาะอย่างยิ่ง ในกรณีที่ไม่มีการบันทึกข้อมูลในระบบ หรือมีการบันทึกข้อมูลในภาวะปกติ เนื่องจากว่าการที่ระบบแบบแฮชได้ทำกระจายข้อมูลออกไปทัดเทียมกันนั้น ทำให้ในการสอบถามข้อมูลแต่ละครั้ง จำนวนเอกสารที่ส่งกลับคืนมาจากแต่ละชาร์ต จะมีปริมาณที่น้อยและไม่ใหญ่มากพอที่จะได้รับผลกระทบจากปัญหาฮอตสปอต แต่เมื่ออยู่สภาวะที่มีการบันทึกเอกสารจำนวนมาก จะพบว่าทั้งสองวิธีต่างใช้เวลาในการทำงานเพิ่มขึ้นเป็นอย่างมาก ซึ่งเกิดมาจากผลกระทบของฮอตสปอต ที่ส่งผลเด่นชัดขึ้นเป็นอย่างมาก โดยเฉพาะในกรณีที่ระบบมีปริมาณงานที่สูงมาก ซึ่งในการสอบถามข้อมูลแบบแฮชนั้น จะมีการใช้งานทุกชาร์ตเสมอกัน ทำให้ฮอตสปอตเป็นเรื่องที่หลีกเลี่ยงไม่ได้

เพื่อที่จะทำความเข้าใจในแบบแผนและผลกระทบของฮอตสปอตให้มากขึ้น จึงได้ทำการตรวจเช็คและรวบรวมข้อมูลการนำซีพียูไปใช้ประโยชน์ ซึ่งข้อมูลดังกล่าวจะถูกเก็บรวบรวมในขณะที่กำลังอยู่ในกลไกสร้างข้อมูลไว้ใช้การทดลอง โดยที่รูปที่ 24 และ 25 แสดงถึงลักษณะการใช้งานซีพียูของแผนการกระจายแบบพิสัยและแบบแฮชตามลำดับ



รูปที่ 24 การใช้ประโยชน์ซีพียูของชาร์ตดิ้งแบบพิสัย



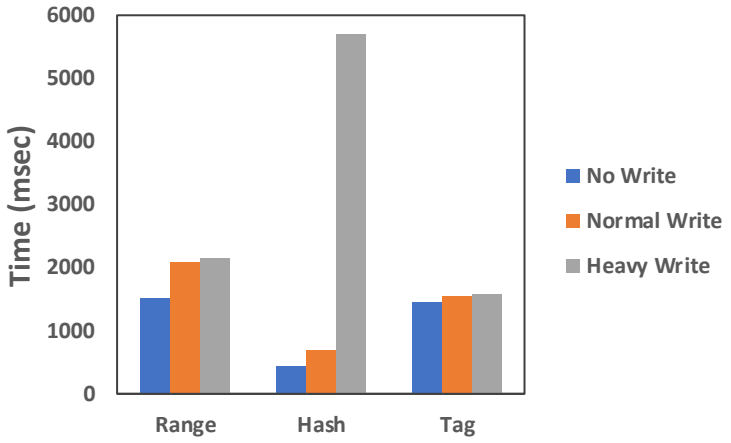
รูปที่ 25 การใช้ประโยชน์ซีพียูของชาร์ตดิ้งแบบแฮช

Chulalongkorn University

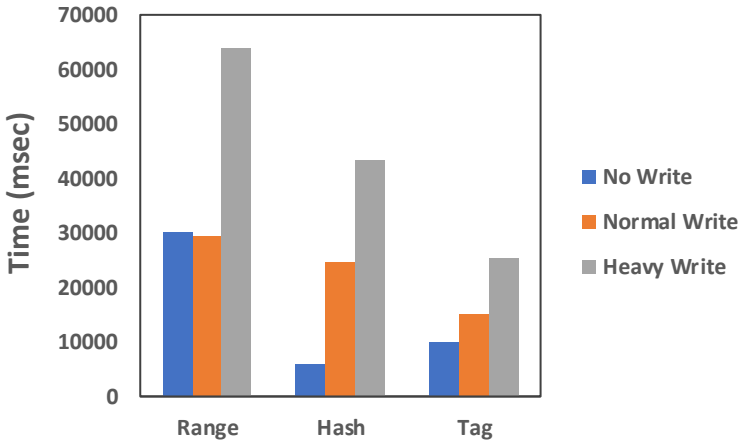
ข้อมูลข้างต้นสามารถใช้อธิบายลักษณะการเกิดฮอตสปอตของทั้งสองแนวทางการกระจายข้อมูลได้เป็นอย่างดี ซึ่งทั้งสองวิธีต่างก็มีลักษณะเกิดฮอตสปอตที่แตกต่างกันออกไป ตามความเคลื่อนไหวของข้อมูลที่ถูกจัดเก็บลงในแต่ละชาร์ต โดยเฉพาะแบบพิสัย ที่มีการใช้ประโยชน์จากซีพียูสูงในระดับ 80% - 90% ทำให้ผลกระทบต่อครั้งค่อนข้างสูง ในขณะที่ค่าเฉลี่ยการใช้ซีพียูของระบบแบบแฮชจะอยู่ที่ 40% - 60% เป็นส่วนใหญ่ แต่แลกกับการที่ทุกชาร์ตจะมีการทำงานตลอดเวลา ทั้งนี้ เกิดมาจากการที่การกระจายข้อมูลในรูปแบบพิสัยนั้น ภาระงานทั้งหมดจะถูกส่งไปรวมกันที่เครื่องชาร์ตใดเพียงชาร์ตเดียว รวมไปถึงการที่เกิดการเพิ่มชิ้นส่วนและการเคลื่อนย้ายชิ้นส่วนบ่อยครั้ง ในขณะที่รูปแบบแฮชนั้น มีภาระที่น้อยกว่า เนื่องจากทุกๆชิ้นส่วนที่ครอบคลุมถึงระยะพิสัยของค่าแฮชนั้น ได้ถูกสร้างไว้ก่อนแล้ว และกระจายอย่างเท่าเทียมกันออกไปทุกเครื่องชาร์ต ทำให้โอกาสที่ชิ้นส่วนจะแตกตัวมีน้อยลง รวมถึงการเคลื่อนย้ายข้อมูลที่น้อยมากจนถึงไม่มี

4.5 ประสิทธิภาพของการกระจายแบบติดป้ายข้อมูล

ในการทดลองเพื่อประเมินผลแนวทางกระจายที่นำเสนอในงานวิจัย การทำชาร์ตดิ้งด้วยการติดป้ายข้อมูลนั้น จะทำการด้วยข้อมูลชุดเดียวกันกับการทดลองประสิทธิภาพของระบบแบบพิสัยและระบบแบบแฮช โดยผลการทดลองจะถูกนำมาเปรียบเทียบกับสองแนวทางเดิม ตามรูปที่ 26 และ รูปที่ 27



รูปที่ 26 ประสิทธิภาพการสอบถามข้อมูล 5 นาที ของแนวทางการกระจายข้อมูล

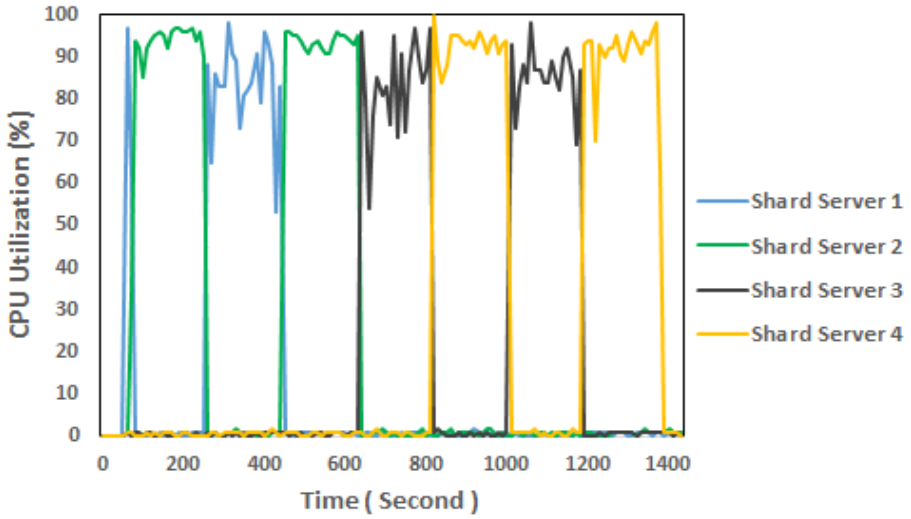


รูปที่ 27 ประสิทธิภาพการสอบถามข้อมูล 2 ขั่วโมง ของแนวทางการกระจายข้อมูล

จากผลการทดลองในรูปที่ 26 หรือกรณีสอบถามข้อมูลในระยะ 5 นาทีนั้น แนวคิดของงานวิจัยนี้นั้นมีประสิทธิภาพใกล้เคียงกับการกระจายแบบพิสัยเนื่องมาจากว่า แนวทางของงานวิจัยนั้นใช้ลักษณะการกระจายแบบพิสัยเป็นพื้นฐาน ทำให้เมื่อข้อมูลมีขนาดเล็กและยังไม่ครอบคลุมชิ้นส่วนข้อมูลจำนวนมากพอที่จะทำให้การใช้งานคำนวณแบบคู่ขนานส่งผลเด่นชัดซึ่งเป็นเหตุผลเดียวกับที่ทำไมประสิทธิภาพการทำงานยังช้ากว่าแบบแฮชอยู่มากในสองกรณีแรก แต่จะสังเกตุได้ว่า

เมื่อเครื่องชาร์ตทำงานหนัก ความเร็วในการประมวลผลจะเริ่มมีการพัฒนาไปในแนวโน้มที่ดีขึ้นเมื่อเทียบกับพิสัยและแฮช ผลของแนวทางที่นำเสนอจะยิ่งแสดงออกอย่างเด่นชัดที่สุดในการกรณีสอบถามข้อมูลแบบ 2 ชั่วโมงในรูปที่ 27 ซึ่งเมื่อผลลัพมาจากหลายๆชาร์ตทำให้ปริมาณเอกสารที่ตอบรับกลับมาจะยิ่งมากตามไปด้วย ส่งผลให้การที่ระบบทำการค้นหาข้อมูลในชาร์ตที่เกิดปัญหาจะยิ่งช้าลงไปด้วย ซึ่งจะสังเกตุได้ว่า ผลกระทบของฮอตสปอตที่เกิดกับแนวทางที่นำเสนอนั้นจะมีน้อยกว่าอีกสองวิธีอย่างเห็นได้ชัดโดยเฉพาะกรณีที่ทำการสืบค้นในสถาณการณ์ที่ระบบมีข้อมูลเข้าอย่างหนัก ในขณะที่แบบพิสัยมีโอกาศสูงมากที่ข้อมูลที่อยู่ใกล้กันจะอยู่รวมอยู่ในชาร์ตที่ยังคงรับข้อมูลอยู่ ซึ่งถึงแม้กระจายออกไปก็ไม่สามารถกระจายได้มากพอทำให้ผลใช้ประโยชน์การคำนวณแบบคู่ขนานได้ ในขณะที่รูปแบบแฮชจะไม่สามารถหลีกเลี่ยงปัญหาได้เลย

ส่วนด้านการใช้งานซีพียูนั้นจะมีความคล้ายคลึงกับรูปแบบพิสัยที่มีค่าเฉลี่ยตกอยู่ที่ 70-90% แต่ส่วนที่ต่างกับรูปแบบพิสัยคือช่วงเวลาที่เกิดฮอตสปอตนั้นสามารถที่จะทำนายได้เพราะข้อมูลชิ้นส่วนที่กระจายนั้นมีเลขเป็นตราเวลาและการกระจายนั้นถูกกำหนดไว้ก่อนแล้วด้วยป้ายบอกข้อมูล ในแนวทางที่นำเสนอนั้นทำให้แต่ละชาร์ตผลัดกันขึ้นมาเพื่อรับภาระงานอย่างเป็นแบบแผน และภาระที่เกิดจากการสร้างชิ้นส่วนข้อมูลและกระจายก็ถูกลดทอนลงด้วยเนื่องจากสิ่งเหล่านี้ถูกทำไปก่อนหน้านี้แล้วในช่วงการสร้างชิ้นส่วนข้อมูลจากป้ายบอกข้อมูล



รูปที่ 28 การใช้ประโยชน์ซีพียูของชาร์ตดิ้งแบบติดป้ายข้อมูล

บทที่ 5
ข้อสรุปงานวิจัย

จากการศึกษาหลักการและกลไกการทำงานของแนวทางการกระจายข้อมูลบนระบบฐานข้อมูลมองโกดีบี ทำให้พบว่าแนวทางการกระจายข้อมูลดั้งเดิมของมองโกดีบีนั้น ยังไม่มีแนวทางไหนที่สามารถตอบโจทย์ข้อมูลบันทึกเหตุการณ์ได้อย่างครอบคลุมและมีประสิทธิภาพมากที่สุด จากการศึกษาพบว่าทั้งสองแนวทางการกระจายข้อมูลซึ่งได้แก่แบบพิสัยและแบบแฮชนั้น ต่างก็มีข้อดีที่มีประโยชน์ และจำเป็นต่อการจัดการข้อมูลบันทึกเหตุการณ์ที่แตกต่างกัน แนวทางการกระจายแบบพิสัยนั้นมีความสามารถในการสืบค้นข้อมูลที่ถูกกำหนดไว้เป็นช่วงของเวลาได้อย่างมีประสิทธิภาพ ซึ่งเป็นรูปแบบการค้นหาที่ผู้ดูแลระบบนิยมใช้ในการเข้าถึงข้อมูลบันทึกเหตุการณ์ ในขณะที่ผลเสียนั้นจะเกิดจากการที่ระบบไม่สามารถกระจายงานการบันทึกข้อมูลออกไปได้อย่างสมดุล ซึ่งในทางกลับกัน แนวทางการกระจายข้อมูลแบบแฮชนั้น สามารถกระจายข้อมูลและภาระงานของการบันทึกข้อมูล ออกไปได้อย่างเท่าเทียม แต่แลกมาด้วยกับการที่จะไม่สามารถทำการสืบค้นข้อมูลได้อย่างมีประสิทธิภาพ ถ้าไม่ทำชะนีข้อมูล ซึ่งการทำดัชนีก็จะไปส่งผลต่อประสิทธิภาพการบันทึกข้อมูลอีกต่อหนึ่ง จากข้อดีข้อเสียข้างต้นทำให้การออกแบบแนวคิดสำหรับการกระจายข้อมูลที่เหมาะสมสำหรับข้อมูลบันทึกเหตุการณ์นั้น จะต้องหาแนวทางแบบผสมผสาน ที่นำเอาจุดเด่นของแนวทางการกระจายข้อมูลทั้งสองวิธีมารวมกัน โดยการลดกลไกการทำงานในส่วนที่เป็นภาระหนักต่อการทำงานอย่างการแตก และเคลื่อนย้ายชิ้นส่วนข้อมูล เพื่อให้ได้ระบบที่มีการทำงานที่เหมาะสมกับข้อมูลบันทึก

ในงานวิจัยนี้ได้นำเสนอวิธีการที่อาศัยลักษณะเด่นจากระบบติดป้ายข้อมูล ในการสร้างชิ้นส่วนข้อมูลขึ้นมาก่อนรับข้อมูล และยังทำให้ผู้ใช้สามารถที่จะกำหนดลักษณะกระจายได้ด้วยตามป้ายข้อมูลที่ถูกกระจายไว้ตามชาร์ตต่างๆ ซึ่งทำให้ระบบมีประสิทธิภาพที่ดีขึ้น อย่างไรก็ตาม การประยุกต์ใช้ระบบข้างต้นเพียงอย่างเดียวนั้น ยังไม่สามารถแก้ไขปัญหาอย่างฮอตสปอตได้อย่างสมบูรณ์แบบ เพื่อให้ระบบสามารถทำงานได้เต็มประสิทธิภาพกับข้อมูลบันทึกเหตุการณ์นั้น ในงานวิจัยนี้ ยังได้ประยุกต์แนวคิดดังกล่าวเข้ากับลักษณะของข้อมูลบันทึกเหตุการณ์ ซึ่งในที่นี่ ข้อมูลจะถูกจัดเรียงตามค่าประทับเวลา และแบ่งกระจายออกไปอย่างทัดเทียมในระบบ เพื่อให้สามารถกระจายภาระงานได้อย่างมีประสิทธิภาพ และสะดวกต่อการสืบค้นข้อมูล จากผลการทดลองในแนวทางภายใต้สภาวะภาระงานระดับต่างๆ ทำให้สามารถพิสูจน์ได้ว่าแนวทางที่นำเสนอ สามารถลดผลกระทบจากปัญหาฮอตสปอตต่อการสืบค้นข้อมูลได้ดีกว่าแนวทางการกระจายข้อมูลทั้งพิสัยและแฮช โดยเฉพาะกรณีที่ข้อมูลมีปริมาณมาก

ยังมีอีกหลายประเด็นที่สามารถนำไปศึกษาต่อยอดจากงานวิจัยนี้ ประเด็นแรกคือการศึกษาเกี่ยวกับความผิดพลาดในการคาดคะเนปริมาณงานที่เกิดขึ้นในระบบ โดยเฉพาะระบบที่ปริมาณงานไม่คงที่และมีการเปลี่ยนแปลงรูปแบบของภาระงานบ่อยครั้ง ซึ่งจะนำไปสู่การที่กระจายข้อมูลได้อย่างไม่เท่าเทียมส่งผลให้ใช้งานทรัพยากรได้อย่างไม่เต็มประสิทธิภาพ ประเด็นที่สอง คือการเคลื่อนย้ายข้อมูลโดยอิงจากการตรวจหาและใช้ประโยชน์จากที่เครื่องชาร์ตที่มีรายละเอียดทรัพยากรระบบ พื้นที่จัดเก็บข้อมูลและความสามารถในการแบกภาระงานที่มากกว่าเครื่องอื่น ในกรณีที่ประสิทธิภาพของเครื่องชาร์ตระบบดังกล่าวไม่เป็นแบบเดียวกันไปหมด

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

รายการอ้างอิง

1. Ward JS, Barker A. Undefined by data: a survey of big data definitions. arXiv preprint arXiv:13095821. 2013.
2. Han J, Haihong E, Le G, Du J, editors. Survey on NoSQL database. Pervasive computing and applications (ICPCA), 2011 6th international conference on; 2011: IEEE.
3. MongoDB. http://www.mongodb.org/ 2017 [
4. DB-Engines. http://db-engines.com/en/ranking 2017 [
5. Cruz F, Maia F, Matos M, Oliveira R, Paulo J, Pereira J, et al., editors. Met: workload aware elasticity for nosql. Proceedings of the 8th ACM European Conference on Computer Systems; 2013: ACM.
6. Konstantinou I, Angelou E, Tsoumakos D, Boumpouka C, Koziris N, Sioutas S, editors. Tiramola: elastic nosql provisioning through a cloud management platform. Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data; 2012: ACM.
7. Chen Z, Yang S, He L, Tan S, Zhang L, Yang H, et al., editors. Load-Aware Fragment Allocation Strategy for NoSQL Database. Proceedings of International Conference on Internet Multimedia Computing and Service; 2014: ACM.
8. Chen Z, Yang S, Tan S, He L, Yin H, Zhang G. A new fragment re-allocation strategy for NoSQL database systems. Frontiers of Computer Science. 2015;9(1):111-27.
9. Cooper BF, Silberstein A, Tam E, Ramakrishnan R, Sears R, editors. Benchmarking cloud serving systems with YCSB. Proceedings of the 1st ACM symposium on Cloud computing; 2010: ACM.
10. Klems M, Silberstein A, Chen J, Mortazavi M, Albert SA, Narayan P, et al., editors. The yahoo!: Cloud datastore load balancer. Proceedings of the fourth international workshop on Cloud data management; 2012: ACM.
11. Konstantinou I, Tsoumakos D, Mytilinis I, Koziris N, editors. DBalancer: distributed load balancing for NoSQL data-stores. Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data; 2013: ACM.

12. Lin S, Chen H, Hu F, editors. A Workload-Driven Approach to Dynamic Data Balancing in MongoDB. Smart City/SocialCom/SustainCom (SmartCity), 2015 IEEE International Conference on; 2015: IEEE.

13. Brewer E. CAP twelve years later: How the" rules" have changed. Computer. 2012;45(2):23-9.

14. Oussous A, Benjelloun F-Z, Lahcen AA, Belfkih S, editors. Comparison and classification of nosql databases for big data. Proceedings of International Conference on Big Data, Cloud and Applications; 2015.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ภาคผนวก

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ประวัติผู้เขียนวิทยานิพนธ์

นายบุญญฤทธิ์ อุณหะวัฒน์ เกิดเมื่อวันที่ 15 มีนาคม พ.ศ. 2532 ที่จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาระดับชั้นปริญญาบริหารธุรกิจบัณฑิต สาขาวิชาธุรกิจสารสนเทศจากมหาวิทยาลัยอัสสัมชัญ เมื่อปีการศึกษา พ.ศ. 2554 และได้เข้าการศึกษาต่อในระดับมหาบัณฑิต คณะวิศวกรรมศาสตร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ที่มหาวิทยาลัยจุฬาลงกรณ์ เมื่อปีการศึกษา 2556

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University