

การใช้แมพรีดิวซ์เชื่อมคอลเลกชันของฐานข้อมูล โนเอสคิวแอลบนมองโกดีบี

Using MapReduce to Join Collection of NOSQL Database Based on MongoDB

นิรุทธิ์ รวยรื่น (Nirut Ruayreun)^{1*} ดร.เกรียงไกร ปอแก้ว (Dr.Kriengkrai Porkaew)**

บทคัดย่อ

ในปัจจุบันปริมาณจำนวนข้อมูลที่จัดเก็บในฐานข้อมูลมีอัตราการเติบโตที่สูง เมื่อข้อมูลที่จัดเก็บมีจำนวนมากในการจัดเก็บข้อมูลลงฐานข้อมูลแบบเชิงสัมพันธ์นั้นจะทำให้การจัดการกับข้อมูลไม่มีประสิทธิภาพเท่าที่ควรเนื่องจากรูปแบบการจัดเก็บข้อมูลมีลักษณะโครงสร้างที่ซับซ้อนทำให้ต้องใช้เวลานานในการจัดการเข้าถึงข้อมูลที่มีปริมาณมาก จึงมีการนำเทคโนโลยีฐานข้อมูลโนเอสคิวแอลเข้ามาใช้ในการจัดเก็บและจัดการข้อมูลที่มีจำนวนมากนั้น การเก็บข้อมูลในฐานข้อมูลโนเอสคิวแอลนั้นจะเป็นการเก็บข้อมูลในลักษณะไม่มีความสัมพันธ์เชิงโครงสร้าง ซึ่งฐานข้อมูลโนเอสคิวแอลไม่รองรับการเชื่อมความสัมพันธ์ของข้อมูลงานวิจัยนี้จึงได้เสนอแนวคิดสร้างความสัมพันธ์ของข้อมูลในฐานข้อมูลโนเอสคิวแอลบนฐานข้อมูลมองโกดีบีโดยนำเทคนิคหลักการเขียนโปรแกรมแบบแมพรีดิวซ์มาประยุกต์ใช้ในการเชื่อมความสัมพันธ์ของข้อมูล และดำเนินการทดลองวัดผลเพื่อนำค่ามาวิเคราะห์วัดประสิทธิภาพในการจัดการข้อมูล

ABSTRACT

At present, the amount of data stored in databases is growing at a high rate. When a lot of data stored in the data storage into relational databases to manipulate data to make it ineffective. Because the format of data storage structures are complex, they require a long time to deal with large volumes of data. It is widely NOSQL database to use to store and manage large volumes of data with it. To store data in NOSQL databases, will store data in a manner no structural relationships. This research proposes the concept of building relationships in the NOSQL database, based on MongoDB. Programming techniques Map Reduce applied to linkage data. This Research conducted experiments to measure and analyze the performance in data management.

คำสำคัญ : การเชื่อมความสัมพันธ์ มองโกดีบี ฐานข้อมูลโนเอสคิวแอล

Key Words : Join relationship, MongoDB, NOSQL database

¹ Correspondent author: r.nirut@hotmail.com

* นักศึกษา หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

** ผู้ช่วยศาสตราจารย์ คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

บทนำ

ในการสร้างความสัมพันธ์เชื่อมโยงระหว่างข้อมูลในฐานข้อมูลเชิงสัมพันธ์นั้น ทำให้ข้อมูลมีความถูกต้องสูงเพราะหลีกเลี่ยงการจัดเก็บข้อมูลที่ซ้ำซ้อน สามารถจัดการในด้านการนำข้อมูลไปใช้เพื่อบรรลุวัตถุประสงค์ในงานด้านต่างๆ ได้ง่ายและมีประสิทธิภาพ เนื่องจากฐานข้อมูลเชิงสัมพันธ์ถูกออกแบบมาให้เหมาะสมกับข้อมูลที่มีโครงสร้างค่อนข้างชัดเจน และยังรองรับการเชื่อมความสัมพันธ์ของข้อมูล ขณะที่ในยุคปัจจุบันนั้นปริมาณข้อมูลที่จัดเก็บในฐานข้อมูลมีอัตราการเติบโตที่สูงมากและมีการจัดเก็บข้อมูลที่หลากหลายในเชิงโครงสร้าง ดังนั้นจึงทำให้การบริหารจัดการกับข้อมูลทำได้ค่อนข้างยากและถ้าหากดำเนินการได้ก็มีความซับซ้อนในการจัดการข้อมูลที่มีปริมาณมาก การเข้าถึงข้อมูลก็จะช้าเพราะต้องเสียเวลาในการเชื่อมโยงข้อมูลที่แยกกันอยู่หลายส่วนทำให้ใช้ระยะเวลานาน

Sadalgae and Fowler [1] กล่าวว่าในปีคริสต์ศักราช 1998 Carlo Strozzi ได้คิดค้นฐานข้อมูลเชิงสัมพันธ์โอเพนซอร์สที่มีการทำงานแบบเบา (Light) โดยออกแบบให้ฐานข้อมูลเก็บตารางเป็นแบบแฟ้มแอสกี (ASCII File) ข้อมูลแต่ละระเบียบ (Tuple) จะเป็นการเก็บแบบบรรทัดที่มีการแบ่งฟิลด์ข้อมูลด้วยแถบช่องว่าง (Tab) ในการค้นหาหรือกระทำการใด ๆ กับข้อมูลก็จะไม่ใช้ภาษาเอสคิวแอล แต่จะใช้ Shell Script ในการดำเนินการ จึงเป็นที่มาของคำว่าฐานข้อมูลโนเอสคิวแอล ซึ่งสามารถจำแนกออกเป็น 4 แบบ คือ Key-Value Data Model, Document Data Model, Column-Family Data Model, Graph Data Model Islam [2] ได้กล่าวว่าฐานข้อมูลมอดโกตีบีเป็นฐานข้อมูลจัดอยู่ในกลุ่ม Document Data Model ของฐานข้อมูลโนเอสคิวแอล ซึ่งฐานข้อมูลมอดโกตีบีถูกพัฒนาจากองค์กร 10gen เป็นฐานข้อมูลแบบโอเพนซอร์ส (Open Source) พัฒนาการจัดการข้อมูลด้วยภาษาซี ฐานข้อมูลมอดโกตีบี มีหลักการทำงานโดยแบ่งการจัดเก็บข้อมูลออกเป็น 3 ส่วนดังต่อไปนี้

ส่วนที่ 1 ฐานข้อมูล (Database) เป็นการจัดเก็บข้อมูลเป็นกลุ่ม โดยในฐานข้อมูลประกอบด้วยกลุ่มข้อมูลคอลเลคชัน เป็นการเก็บข้อมูลคอลเลคชันเดียวหรือรวบรวมหลาย ๆ คอลเลคชันเก็บไว้ในฐานข้อมูลได้

ส่วนที่ 2 คอลเลคชัน (Collection) เป็นการจัดเก็บข้อมูลเป็นกลุ่มซึ่งอยู่ภายใต้ฐานข้อมูล โดยคอลเลคชัน ประกอบด้วยกลุ่มข้อมูลดอคิวเมนต์ (Document) ซึ่งหลักการจัดเก็บของคอลเลคชันจะคล้ายกับการจัดเก็บแบบตารางในฐานข้อมูลเชิงสัมพันธ์ การจัดเก็บต่างกับตารางในฐานข้อมูลเชิงสัมพันธ์ตรงที่คอลเลคชันไม่ต้องสร้างโครงสร้างที่จัดเก็บข้อมูลที่แน่นอนก่อนนำค่าข้อมูลเก็บลงในฐานข้อมูล

ส่วนที่ 3 ดอคิวเมนต์ (Document) เป็นการจัดเก็บข้อมูลซึ่งอยู่ภายใต้คอลเลคชันโดยในดอคิวเมนต์ประกอบไปคู่ของฟิลด์และแวลู (field-and-value pairs) โดยฟิลด์ใช้ในการอ้างอิงและแวลูเป็นค่าของข้อมูลโดยที่แวลูสามารถเป็นข้อมูลประเภทต่างๆ ได้เช่น ตัวอักษร (String), ตัวเลขจำนวนเต็ม (Integer), ตัวเลขทศนิยม (Float), อาร์เรย์ (Array) เป็นต้น อีกทั้งในค่าแวลูยังสามารถเก็บค่าข้อมูลที่เป็นแบบดอคิวเมนต์ซ้อนลงไปได้อีกด้วย

ในงานวิจัยนี้ได้ศึกษาการจัดการข้อมูลของฐานข้อมูลโนเอสคิวแอล โดยศึกษาจากฐานข้อมูลมอดโกตีบีเนื่องจากฐานข้อมูลมอดโกตีบีสามารถเข้าถึงข้อมูลได้เร็วมาก ในข้อมูลกว่า 50 กิกะไบต์ ฐานข้อมูลมอดโกตีบีสามารถเข้าถึงข้อมูลได้เร็วกว่าฐานข้อมูลมายเอสคิวแอล ถึง 10 เท่า [3] นอกจากนี้ฐานข้อมูลมอดโกตีบียังมีหลายองค์กรที่เลือกใช้ [4] และศึกษาอัลกอริทึมที่ใช้จัดการข้อมูลปริมาณมากด้วยแมพรีดิวซ์

เทคนิคแมพรีดิวซ์ (MapReduce) เป็นแนวคิดวิธีการที่ใช้ในการค้นหาข้อมูลที่มีปริมาณมาก [5] ได้กล่าวว่ารูปแบบการเขียนโปรแกรมตามแนวคิดแมพรีดิวซ์ ถูกพัฒนาขึ้นโดยองค์กรกูเกิล (Google) เพื่อเข้ามาช่วยในการค้นหาข้อมูลที่มีปริมาณมากและเข้ามาช่วยในการประมวลผลข้อความ บนที่เก็บ

ข้อมูลเว็บขนาดใหญ่ ซึ่งข้อมูลเว็บเหล่านี้ถูกเก็บใน BigTable และระบบการเก็บไฟล์แบบกระจาย (Google File System) แนวคิดการเขียนโปรแกรมแบบแมพรีดิวซ์ ถูกออกแบบเพื่อการประมวลผลและสร้างข้อมูลที่มีขนาดใหญ่ด้วยวิธีการคำนวณแบบขนานพร้อมกันของหน่วยประมวลผล (Processor) จำนวนหนึ่งพันหน่วยในเวลาเดียวกัน

ในปัจจุบันฐานข้อมูลโนเอสคิวแอลไม่รองรับการเชื่อมความสัมพันธ์ของข้อมูล ดังนั้นเพื่อเป็นการเพิ่มความถูกต้องของข้อมูลและยังคงประสิทธิภาพด้านความเร็วในการสอบถาม (Query) ข้อมูลของฐานข้อมูลโนเอสคิวแอล ดังนั้นงานวิจัยนี้จึงได้เสนอแนวคิดสร้างความสัมพันธ์ของฐานข้อมูลโนเอสคิวแอลขึ้นมา โดยใช้ฐานข้อมูลมอดโกตีบี และใช้หลักการเขียนโปรแกรมแบบแมพรีดิวซ์มาประยุกต์ใช้ สำหรับการวัดค่าประสิทธิภาพนั้นวัดจากการทดลองหาค่าเวลาที่ใช้ในการสอบถามข้อมูลจนได้ผลลัพธ์ที่ต้องการ ระหว่างวิธีการเชื่อมคอลเลคชันโดยใช้แมพรีดิวซ์กับวิธีการเชื่อมคอลเลคชันโดยการเขียนโปรแกรมเพื่อเชื่อมข้อมูลเอง (ภาพที่ 1) เป็นกระบวนการทำงานของฟังก์ชัน แมพรีดิวซ์ ในการเชื่อมความสัมพันธ์ของข้อมูลระหว่างคอลเลคชันมีขั้นตอนดังนี้

ขั้นตอนที่ 1 การสร้างฟังก์ชันแมพ เป็นการสร้างฟังก์ชันแมพขึ้นมาให้ทำงานกับคอลเลคชันแรก โดยมีอาร์กิวเมนต์ (Argument) 2 ค่าคือคีย์ (Key) และแวลู (Value) ค่า อาร์กิวเมนต์ แวลู ที่ถูกส่งโดยฟังก์ชัน Emit นั้น จะเป็นเอาต์พุต (Output) ที่มีฟิลด์ของข้อมูลตรงตามความต้องการใช้งาน หากฟิลด์ของข้อมูลที่ต้องการอยู่คอลเลคชันที่สอง ให้ทำการนำคีย์นอก (Foreign Key) ของข้อมูลภายใต้อาร์กิวเมนต์คีย์นั้น ของคอลเลคชันแรกไปค้นหาในคอลเลคชันที่สองก่อน แล้วนำผลลัพธ์ที่ได้จากการค้นหามารวมเป็นเอาต์พุตเดียว เพื่อให้รูปแบบของฟังก์ชัน Emit มีลักษณะเป็นคู่ของคีย์และแวลู

ขั้นตอนที่ 2 การสร้างฟังก์ชันรีดิวซ์ เป็นการสร้างฟังก์ชันรีดิวซ์ขึ้นมาให้มีการรับพารามิเตอร์ 2 ค่า

คือ คาคีย์และค่าแวลู สร้างตัวแปรที่จะเป็นเอาต์พุตให้ประกอบไปด้วยชื่อฟิลด์ที่ต้องการ ต่อมานำค่าจากพารามิเตอร์แวลูมาเขียนโปรแกรมวนลูปเพื่อกำหนดค่าให้กับฟิลด์ที่ต้องการเป็นลักษณะการเตรียมข้อมูลทั้งหมดกำหนดไว้ในตัวแปรที่เป็นเอาต์พุตเพื่อใช้แสดงผลลัพธ์

ขั้นตอนที่ 3 เป็นรูปแบบการเรียกใช้คำสั่งแมพรีดิวซ์ของมอดโกตีบีให้ทำงาน เมื่อคำสั่งแมพรีดิวซ์ประมวลผลการทำงาน ตามฟังก์ชันแมพและฟังก์ชันรีดิวซ์เสร็จ จะมีการสร้างผลลัพธ์ออกมาอยู่ในรูปของคอลเลคชันชั่วคราว (Temp Collection)

ในงานวิจัยของ Calil [6] นั้นได้นำเสนอวิธีการเชื่อมความสัมพันธ์ของข้อมูลบนฐานข้อมูลที่ชื่อว่า SimpleDB ที่เป็นบริการฐานข้อมูลของ Amazons เพื่อใช้จัดการข้อมูลบนระบบคลาวด์ ซึ่งเป็นฐานข้อมูลโนเอสคิวแอลแบบ Document Data Model ชนิดหนึ่ง ด้วยวิธีการสร้างอินเตอร์เฟสขึ้นมาให้รองรับการเขียนคำสั่งสอบถามข้อมูลด้วยภาษาเอสคิวแอล ซึ่งประกอบไปด้วยหลักการทำงาน 3 ส่วนดังนี้

ส่วนที่ 1 Access interface ทำหน้าที่รับคำสั่งที่เป็นภาษาเอสคิวแอล โดยคำสั่งภาษาเอสคิวแอลที่ใช้เชื่อมความสัมพันธ์ของข้อมูลจะเป็นแบบ INNER JOIN

ส่วนที่ 2 Command decomposition ทำหน้าที่แยกคำสั่งเอสคิวแอลที่รับมาให้เป็นคำสั่งที่ง่าย ไม่มีการเชื่อมความสัมพันธ์ของข้อมูล ด้วยวิธี regular expression และแปลงคำสั่งนั้นให้อยู่ในรูปแบบของภาษาสอบถามของ SimpleDB

ส่วนที่ 3 Processing and Return ทำหน้าที่ส่งคำสั่งที่แยกออกเป็นส่วนๆ ไปสอบถามข้อมูลที่ SimpleDB เมื่อได้ผลลัพธ์กลับมาแล้วจะต้องนำผลลัพธ์แต่ละส่วนมารวมกันเพื่อส่งกลับไปยังผู้ใช้

วัตถุประสงค์การวิจัย

1. เพื่อศึกษารูปแบบจัดเก็บข้อมูลของฐานข้อมูลโนเอสคิวแอลและการจัดการข้อมูลของฐานข้อมูลมอดโกตีบี

2. เพื่อศึกษารูปแบบอัลกอริทึมที่ใช้ในการจัดการข้อมูลปริมาณมากด้วยแมพรีดิวซ์

3. เพื่อศึกษาและเปรียบเทียบผลลัพธ์ที่ได้จากการพัฒนาโปรแกรมเชื่อมคอลเลกชันของฐานข้อมูลมองโกดีบีกับวิธีการเขียนโปรแกรมเชื่อมคอลเลกชันเอง

วิธีการวิจัย

ในงานวิจัยนี้ได้ทำการศึกษาและเปรียบเทียบการเชื่อมคอลเลกชันของฐานข้อมูลมองโกดีบี ที่ได้มีการออกแบบโครงสร้างข้อมูลให้มีความสัมพันธ์กัน โดยการเปรียบเทียบนั้นจะทำการเปรียบเทียบเวลาที่ใช้ในการเชื่อมคอลเลกชันเพื่อให้ได้ข้อมูลตามที่ต้องการด้วยวิธีเขียนเชื่อมคอลเลกชันในโปรแกรมประยุกต์ (Application) เองกับวิธีที่มีการพัฒนาโปรแกรมให้เป็นไลบรารีที่มีการประยุกต์ใช้แมพรีดิวซ์ในการเชื่อมคอลเลกชันและเพิ่มขั้นตอนการกรองข้อมูลก่อนที่จะเชื่อมคอลเลกชันด้วยแมพรีดิวซ์ โดยงานวิจัยนี้ได้ทำการศึกษาและเสนอวิธีการเชื่อมคอลเลกชันด้วยกันทั้งหมด 5 รูปแบบ ดังนี้

รูปแบบที่ 1 join direct เป็นวิธีที่ใช้หลักการสอบถามข้อมูลจากคอลเลกชันแรกมาเก็บไว้ในโปรแกรม แล้ววนลูปนำคีย์นอกแต่ละคีย์ไปสอบถามข้อมูลในคอลเลกชันที่สอง (ภาพที่ 2) สามารถอธิบายขั้นตอนการทำงานได้ดังนี้

ขั้นตอนที่ 1 โปรแกรมฝั่งไคลเอนต์ส่งคำสั่งเพื่อสอบถามข้อมูลที่ต้องการ ซึ่งอยู่ในคอลเลกชันที่หนึ่ง ไปยังฐานข้อมูลมองโกดีบี

ขั้นตอนที่ 2 ฐานข้อมูลมองโกดีบีทำการประมวลผลคำสั่งสอบถามข้อมูลที่ได้รับมาและส่งผลลัพธ์กลับไปยังโปรแกรมประยุกต์ของผู้ใช้งาน

ขั้นตอนที่ 3 เมื่อโปรแกรมประยุกต์ฝั่งไคลเอนต์ได้รับค่าผลลัพธ์จากฐานข้อมูลมองโกดีบี โปรแกรมจะทำการวนลูปข้อมูลโดยนำค่าฟิลด์ที่เป็นคีย์นอกแต่ละคีย์ส่งไปสอบถามข้อมูลในคอลเลกชันที่สองที่ฐานข้อมูลมองโกดีบี

ขั้นตอนที่ 4 ฐานข้อมูลมองโกดีบีประมวลผลคำสั่งสอบถามข้อมูลและส่งผลลัพธ์กลับไปยังโปรแกรมประยุกต์ฝั่งไคลเอนต์

รูปแบบการเชื่อมคอลเลกชันที่ประยุกต์ใช้แมพรีดิวซ์และพัฒนาเป็นไลบรารีที่มีการนำเสนอในงานวิจัยนั้นต้องการพารามิเตอร์ทั้งหมด 4 ค่า คือ

ค่าที่ 1 sSelect เป็นค่าฟิลด์ของข้อมูลที่ต้องการสอบถาม มีรูปแบบคือ collection name, field1, collection name.field2 เช่น student.std_id, student.std_title_th, student.std_name, student.surname, advisor.first_name, advisor.last_name

ค่าที่ 2 sFrom เป็นค่าของชื่อคอลเลกชันทั้งสองที่ต้องการเชื่อมข้อมูลเข้าด้วยกัน มีรูปแบบคือ collection name1, collection name2 เช่น student, advisor

ค่าที่ 3 sWhere เป็นค่าที่ระบุชื่อฟิลด์ที่เป็นคีย์เพื่อใช้เชื่อมสองคอลเลกชัน มีรูปแบบคือ collection name1.key_id = collection name2.key_id เช่น student.advisor_id = advisor.advisor_id

ค่าที่ 4 sCondition เป็นค่าของเงื่อนไขในการสอบถามข้อมูล มีรูปแบบคือ collection name.field <Comparison Operator> value, collection name.field <Comparison Operator> value เช่น student.extra_status='N', student.admit_proj=52, student.major_id='Q03', advisor.adv_status=3 ซึ่ง <Comparison Operator> ได้แก่ =, <, >, <=, >=, != โดยในงานวิจัยนี้จะเรียกไลบรารีที่พัฒนาขึ้นว่า MongoDB Library Join ทั้งนี้เพื่อให้การเรียกชื่อโปรแกรมที่พัฒนาขึ้นเป็นไปในรูปแบบเดียวกัน

รูปแบบที่ 2 MR join ใช้หลักการเชื่อมคอลเลกชันของข้อมูลด้วย MongoDB Library Join ที่ใช้แมพรีดิวซ์ (ภาพที่ 3) สามารถอธิบายขั้นตอนการทำงานได้ดังนี้

ขั้นตอนที่ 1 โปรแกรมฝั่งไคลเอนต์เรียกใช้ MongoDB Library Join โดยส่งค่าพารามิเตอร์ตามที่ไลบรารีระบุ คือ sSelect, sFrom, sWhere, sCondition

ขั้นตอนที่ 2 กระบวนการของ MongoDB Library Join เมื่อได้รับค่าข้อมูลตามพารามิเตอร์ที่ระบุ จะนำค่าข้อมูล sCondition มาสร้าง query string condition ซึ่งเป็นเงื่อนไขในการสอบถามข้อมูลโดยสร้างให้อยู่ในรูปแบบคำสั่งสอบถามที่ใช้ในฐานข้อมูลมองโกดีบี จากนั้นกระบวนการต่อมาจะนำค่าข้อมูล sSelect, sFrom, sWhere และ query string condition นำค่าข้อมูลทั้งหมดมาสร้างเป็น query string MapReduce และทำการนำ query string MapReduce ส่งไปประมวลผลที่ฐานข้อมูลมองโกดีบี ซึ่งผลลัพธ์ที่ได้จากแมพรีดิวซ์ จะอยู่ในรูปแบบคอลเลคชันชั่วคราว

ขั้นตอนที่ 3 MongoDB Library Join จะส่งคำสั่งสอบถามข้อมูลไปที่คอลเลคชันชั่วคราว และส่งผลลัพธ์กลับไปยังโปรแกรมฝั่งไคลเอนต์

รูปแบบที่ 3 MR filter join ใช้หลักการเชื่อมคอลเลคชันของข้อมูลด้วย MongoDB Library Join ที่มีการกรองข้อมูลทั้งสองคอลเลคชันมาสร้างเป็นคอลเลคชันชั่วคราวก่อนแล้วใช้แมพรีดิวซ์ (ภาพที่ 4) สามารถอธิบายขั้นตอนการทำงานได้ดังนี้

ขั้นตอนที่ 1 โปรแกรมฝั่งไคลเอนต์เรียกใช้ MongoDB Library Join โดยส่งค่าตามพารามิเตอร์ที่ระบุในไลบรารีคือ sSelect, sFrom, sWhere, sCondition

ขั้นตอนที่ 2 กระบวนการของ MongoDB Library Join เมื่อได้รับค่าข้อมูลตามพารามิเตอร์ที่ระบุ จะนำค่าข้อมูล sFrom, sWhere, sCondition มาแยกสร้าง query string condition ของคอลเลคชันที่หนึ่ง แล้วนำ query string condition นี้ไปประมวลผลที่ฐานข้อมูลมองโกดีบี เพื่อสร้างคอลเลคชันชั่วคราวของคอลเลคชันที่หนึ่ง และต่อมา

จะทำการสร้าง query string condition ของคอลเลคชันที่สอง แล้วนำ query string condition นี้ไปประมวลผลที่ฐานข้อมูลมองโกดีบี เพื่อสร้างคอลเลคชันชั่วคราวของคอลเลคชันที่สอง กระบวนการต่อมานำค่าข้อมูล sSelect, sWhere, ชื่อของคอลเลคชันชั่วคราวที่หนึ่งและชื่อคอลเลคชันชั่วคราวที่สอง นำค่าข้อมูลทั้งหมดมาสร้างเป็น query string MapReduce และทำการนำ query string MapReduce นี้ส่งไปประมวลผลที่ฐานข้อมูลมองโกดีบี ซึ่งผลลัพธ์ที่ได้จากแมพรีดิวซ์ จะอยู่ในรูปแบบคอลเลคชันชั่วคราว

ขั้นตอนที่ 3 MongoDB Library Join จะประมวลผลคำสั่งสอบถามข้อมูลไปที่คอลเลคชันชั่วคราวและส่งผลลัพธ์กลับไปยังโปรแกรมฝั่งไคลเอนต์

รูปแบบที่ 4 MR left filter join ใช้หลักการเชื่อมคอลเลคชันของข้อมูลด้วย MongoDB Library Join ที่มีการกรองข้อมูลของคอลเลคชันแรกมาสร้างเป็นคอลเลคชันชั่วคราวก่อนเท่านั้นแล้วใช้แมพรีดิวซ์ โดยขั้นตอนการทำงานจะคล้ายกับรูปแบบ MR filter join เพียงแต่การสร้างคอลเลคชันชั่วคราวจะสร้างจากเงื่อนไขที่มาจากคอลเลคชันที่หนึ่งเท่านั้น โดยกระบวนการเชื่อมคอลเลคชันของแมพรีดิวซ์จะใช้คอลเลคชันชั่วคราวที่สร้างขึ้นกับคอลเลคชันที่สองที่ไม่มีการกรองข้อมูล

รูปแบบที่ 5 MR right filter join ใช้หลักการเชื่อมคอลเลคชันของข้อมูลด้วย MongoDB Library Join ที่มีการกรองข้อมูลของคอลเลคชันที่สองมาสร้างเป็นคอลเลคชันชั่วคราวก่อนเท่านั้นแล้วใช้แมพรีดิวซ์ โดยขั้นตอนการทำงานจะคล้ายกับรูปแบบ MR filter join เพียงแต่การสร้างคอลเลคชันชั่วคราวจะสร้างจากเงื่อนไขที่มาจากคอลเลคชันที่สองเท่านั้น โดยกระบวนการเชื่อมคอลเลคชันของแมพรีดิวซ์จะใช้คอลเลคชันชั่วคราวที่สร้างขึ้นกับคอลเลคชันที่หนึ่งที่ไม่มีการกรองข้อมูล

ผลการวิจัย

การวัดค่าเวลาที่ใช้ในการเชื่อมคอลเลคชันนั้นผู้วิจัยได้จำลองสภาพแวดล้อมของการทดสอบโดยใช้เครื่องคอมพิวเตอร์ Processor: Intel® Core™ i5-3337U CPU @ 1.80GHz 1.80 GHz, RAM: 8.00 GB, System type: 64-bit Operating System, x64-based processor, OS: Windows 8 โปรแกรม Microsoft visual studio 2012 ใช้ภาษา C# สำหรับการเขียนโปรแกรม ใช้ฐานข้อมูลมอดโก ดีบี เวอร์ชัน 2.2.6 โปรแกรมที่ใช้วัดค่าเวลา WAPT 8.1 ข้อมูลที่ใช้ทดสอบมี 2 คอลเลคชันคือคอลเลคชัน student: 16 fields, 9,665 documents คอลเลคชัน advisor: 11 fields, 5,954 documents ทั้งสองคอลเลคชันใช้อินเด็กซ์ที่มอดโกดีบีสร้างให้ ไม่มีการสร้างอินเด็กซ์เพิ่มเติม และพัฒนาโปรแกรมสำหรับทดสอบจำนวน 5 รูปแบบ

การทดลองเชื่อมคอลเลคชันแบ่งเป็น 2 ชุด โดยแต่ละชุด มี 3 คิวรี เพื่อใช้สำหรับทดสอบการเชื่อมคอลเลคชันทั้ง 5 แบบ โดยให้คอลเลคชัน student ใช้แต่ละคำสั่งสอบถามที่มีเงื่อนไขกรองข้อมูลได้ต่างกัน ส่วนคอลเลคชัน advisor ทั้ง 3 คำสั่งสอบถาม กรองข้อมูลได้เท่ากันในแต่ละชุด ดังแสดงใน (ตารางที่ 1) และ (ตารางที่ 2) จากการทดลองเชื่อมคอลเลคชันทั้ง 5 รูปแบบ เพื่อเปรียบเทียบเวลาในการเชื่อมข้อมูลได้ผลการทดลอง โดยแบ่งเป็นผลการทดลองจากข้อมูลชุดที่ 1 และชุดที่ 2 (ภาพที่ 5) และ (ภาพที่ 6) จะเห็นได้ว่าหากข้อมูลจากเงื่อนไขในการเชื่อมคอลเลคชันสามารถกรองข้อมูลได้มากจะทำให้การเชื่อมคอลเลคชันทำได้ค่อนข้างเร็วแต่หากเงื่อนไขในการเชื่อมคอลเลคชันกรองข้อมูลได้น้อยวิธี MR filter join จะช่วยให้การเชื่อมคอลเลคชันทำได้ค่อนข้างเร็ว (ภาพที่ 7) เป็นการเปรียบเทียบเวลาที่ใช้ในการเชื่อมคอลเลคชันของข้อมูลชุดที่ 1 และชุดที่ 2 ของวิธีการเชื่อมคอลเลคชันแต่ละแบบในแต่ละคำสั่งสอบถาม จะเห็นได้ว่าข้อมูลชุดที่ 2 ซึ่งเงื่อนไขสามารถกรองข้อมูลได้มากกว่าชุดที่ 1 เมื่อนำ

มาใช้กับวิธี MR filter join จะทำให้เชื่อมคอลเลคชันได้เร็วกว่าวิธีอื่น

การเชื่อมคอลเลคชันด้วยวิธี direct join นอกจากจะใช้เวลาในการเชื่อมคอลเลคชันนานแล้วยังใช้ซีพียูมากกว่าวิธีอื่นอีกด้วย เนื่องจากช่วงของการวนลูปนำค่าคีย์นอกแต่ละคีย์ไปค้นหาข้อมูลที่อยู่ในคอลเลคชันที่สองซึ่งจะต้องมีการติดต่อกับฐานข้อมูลตลอด จนกว่าคีย์ที่ใช้ในการค้นหาข้อมูลจะหมดลง (ภาพที่ 8) เป็นการเปรียบเทียบปริมาณการใช้ซีพียูเพื่อเชื่อมคอลเลคชันแต่ละแบบตามแต่ละคำสั่งสอบถามจะเห็นว่าวิธี direct join จะใช้ซีพียูมากกว่าวิธีอื่น

อภิปรายและสรุปผลการวิจัย

ในบทความงานวิจัยนี้ได้นำเสนอแนวคิดและวิธีในการเชื่อมคอลเลคชันของฐานข้อมูลมอดโกดีบีซึ่งไม่รองรับการเชื่อมคอลเลคชันก็เพื่อเป็นการนำข้อดีของฐานข้อมูลโนเอสคิวแอลในเรื่องการอ่านและเขียนข้อมูลที่เร็วมารวมกับแนวคิดของฐานข้อมูลเชิงสัมพันธ์ที่ลดความซ้ำซ้อนของข้อมูลโดยการแยกตารางซึ่งในฐานข้อมูลมอดโกดีบีคือการแยกคอลเลคชันและประยุกต์ใช้วิธีการประมวลผลข้อมูลด้วยวิธีแมพรีดิวซ์มาช่วยในการเชื่อมคอลเลคชันของฐานข้อมูลมอดโกดีบีโดยเพิ่มขั้นตอนการกรองข้อมูลของคอลเลคชัน แล้วนำมาสร้างเป็นคอลเลคชันชั่วคราว เพื่อใช้ในการเชื่อมคอลเลคชัน

จากการทดลองจะเห็นได้ว่าการสร้างคอลเลคชันชั่วคราวขึ้นมาก่อนที่จะเชื่อมกันนั้นจะทำให้การเชื่อมคอลเลคชันทำได้เร็วขึ้น ในวิธี MR filter join จะเป็นการสร้างคอลเลคชันชั่วคราวขึ้นมาทั้งสองคอลเลคชัน วิธี MR left filter join จะสร้างคอลเลคชันชั่วคราวของพารามิเตอร์ sCondition ลำดับแรก วิธี MR right filter join จะสร้างคอลเลคชันชั่วคราวของพารามิเตอร์ sCondition ลำดับสอง แม้ว่าการทดลองจะเห็นว่าวิธี MR filter join ใช้เวลาน้อยสุด ก็ไม่ได้หมายความว่า

จะดีกว่าวิธี MR left filter join หรือ MR right filter join ทั้งนี้ขึ้นอยู่กับปริมาณข้อมูลและเงื่อนไขในการกรองข้อมูล ดังจะเห็นได้จากการทดลองของข้อมูลชุดที่ 2 ในรูปที่ 6 ตรง query 3 วิธี MR filter join กับวิธี MR right filter join ใช้เวลาในการเชื่อมคอลเลคชันใกล้เคียงกัน

ในส่วนของวิธี MR join จะไม่มีการสร้างคอลเลคชันชั่วคราวขึ้นมาก่อนแต่จะใช้แมพรีดิวซ์อย่างเดียว ดังนั้นถ้าหากเงื่อนไขในการเชื่อมคอลเลคชันกรองข้อมูลได้น้อยก็จะใช้เวลาที่ค่อนข้างนานเพราะหลักแมพรีดิวซ์จะอ่านข้อมูลทั้งหมดในคอลเลคชันสำหรับวิธี direct join จะเป็นการเขียนโปรแกรมเองซึ่งก็หลักการกรองข้อมูลจากคอลเลคชันที่หนึ่งก่อน หลังจากนั้นจะนำคีย์นอก แต่ละคีย์วนลูปไปสอบถามในคอลเลคชันที่สอง ซึ่งวิธีนี้ใช้เวลาค่อนข้างนานและเปลืองทรัพยากรซีพียู แต่ถ้าเงื่อนไขกรองข้อมูลได้น้อยการใช้ซีพียูของแต่ละแบบก็ไม่ได้แตกต่างกันมากจนเป็นนัยสำคัญ

แนวคิดและวิธีที่นำเสนอในงานวิจัยนี้เป็นแนวทางหนึ่งที่จะเชื่อมคอลเลคชันกรณีมีการออกแบบข้อมูลที่มีความสัมพันธ์กันแต่อยู่คนละคอลเลคชันให้สามารถเชื่อมกันได้ สำหรับการพัฒนางานวิจัยนี้ในอนาคตอาจพัฒนาให้โปรแกรมมีการตัดสินใจว่าจะกรองข้อมูลเพื่อสร้างคอลเลคชันชั่วคราว หรือไม่ โดยถ้าเงื่อนไขกรองข้อมูลได้น้อยก็ไม่จำเป็นต้องสร้างคอลเลคชันชั่วคราวให้เสียเวลา

เอกสารอ้างอิง

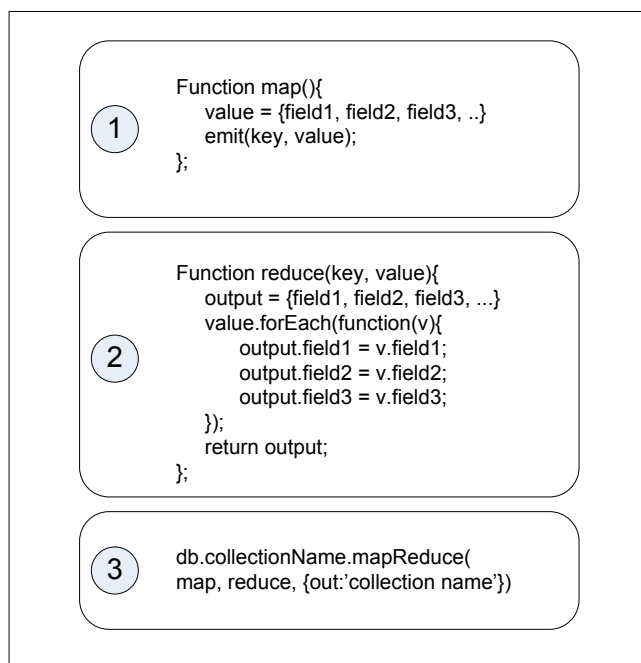
1. Pramod J. Sadalage and Martin Fowler, 2013, NoSQL distilled : a brief guide to the emerging world of polyglot, 2nded., Pearson Education Inc, United States of America, pp. 9-28.
2. Rubayeet Islam. 2011. PHP and MongoDB Web development beginner's guide, Packt Publishing, Birmingham, pp. 8-12.
3. Jing Han. 2011. Survey on NoSQL Database. pervasive computing and applications (ICPCA), 6th International Conference, 2011, pp. 363-366.
4. MongoDB Inc. Customer Success Stories. Retrieved [Online], Available : <http://www.mongodb.com/customers>. [2014, January 27].
5. Gautam Shroff. 2010. Enterprise cloud computing technology architecture application, Cambridge University Press, United Kingdom, pp. 131-142.
6. Andre Calil, Ronaldo dos Santos Mello. 2012. SimpleSQL: A relational layer for simpleDB, ADBIS 2012, LNCS 7503, 2012, pp. 99-110.

ตารางที่ 1 แสดงจำนวนดอควิเมนต์แต่ละเงื่อนไข ชุดที่ 1

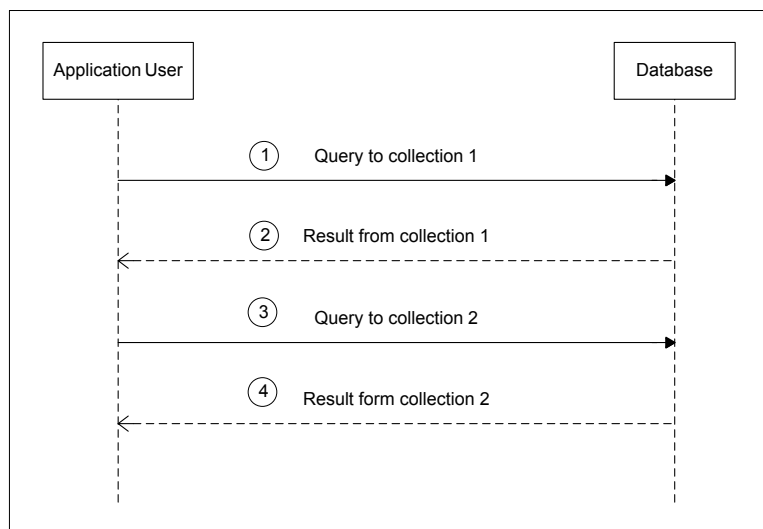
condition	student (%)	advisor (%)
sCondition query 1	146 (1.51)	3,423 (57.64)
sCondition query 2	3,746 (38.78)	3,423 (57.64)
sCondition query 3	6,867 (71.14)	3,423 (57.64)

ตารางที่ 2 แสดงจำนวนดอควิเมนต์แต่ละเงื่อนไข ชุดที่ 2

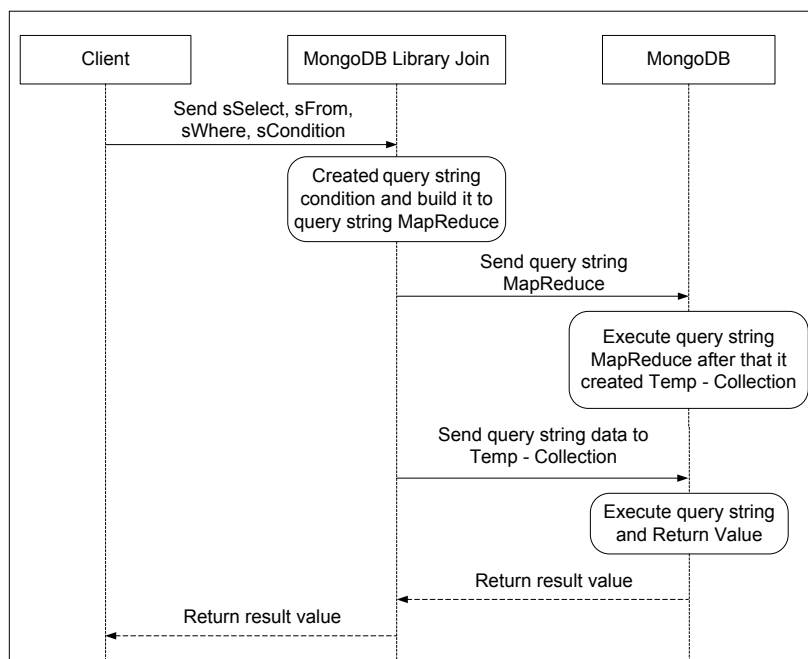
condition	student (%)	advisor (%)
sCondition query 1	146 (1.51)	1,471 (24.71)
sCondition query 2	3,746 (38.78)	1,471 (24.71)
sCondition query 3	6,867 (71.14)	1,471 (24.71)



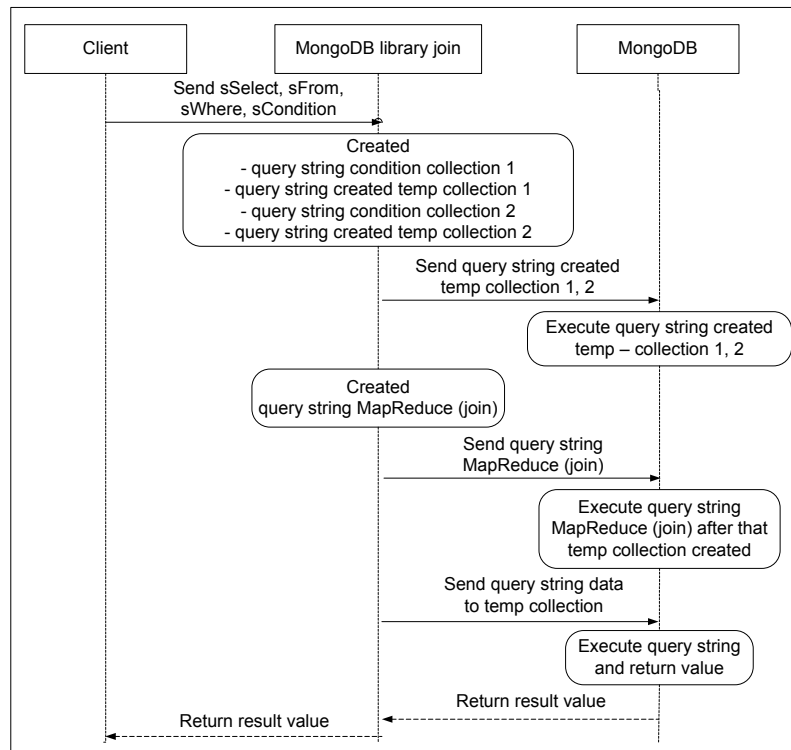
ภาพที่ 1 กระบวนการเชื่อมความสัมพันธ์ของข้อมูลในรูปแบบฟังก์ชันแมปรีดิวซ์



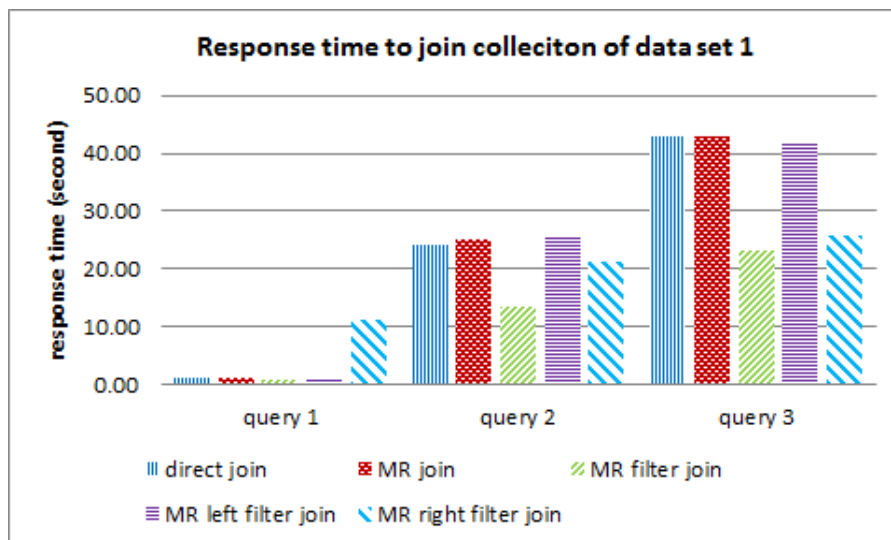
ภาพที่ 2 ขั้นตอนการเชื่อมคอลเลคชันแบบ direct join



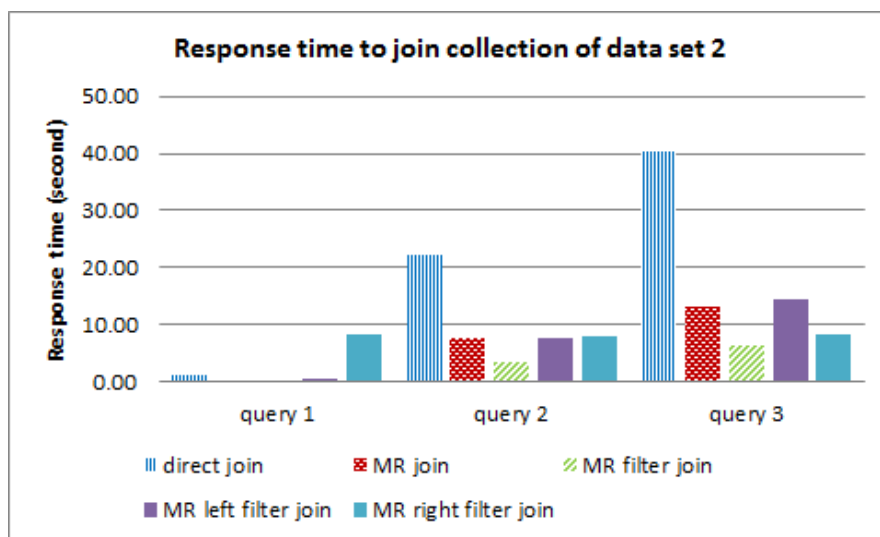
ภาพที่ 3 ขั้นตอนการเชื่อมคอลเลคชันของข้อมูลด้วย MongoDB Library Join ที่ใช้แมพรีดิวซ์



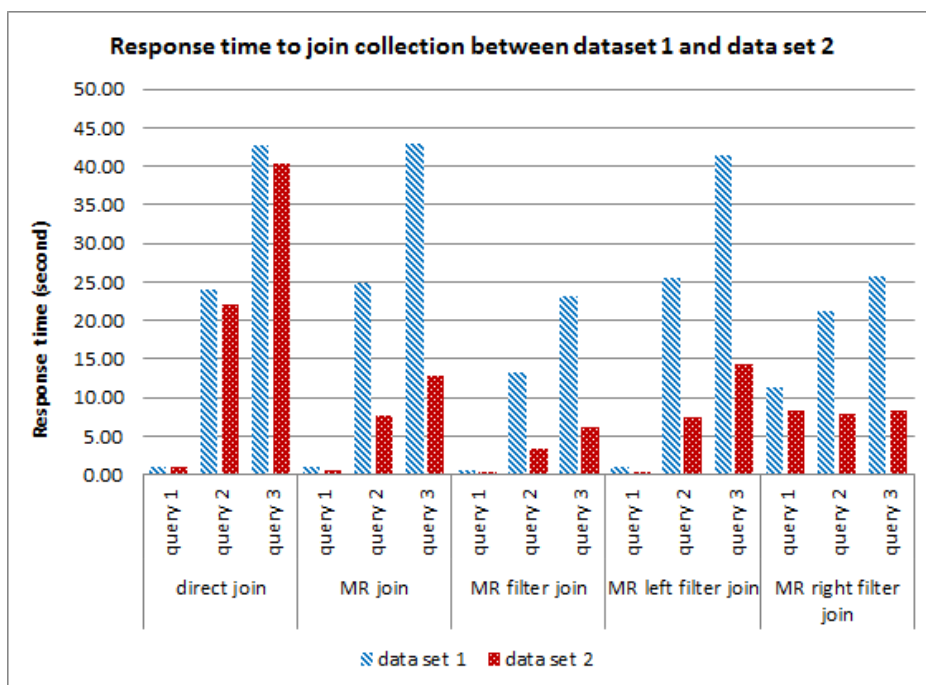
ภาพที่ 4 ขั้นตอนการเชื่อมคอลเลคชันแบบ MR filter join



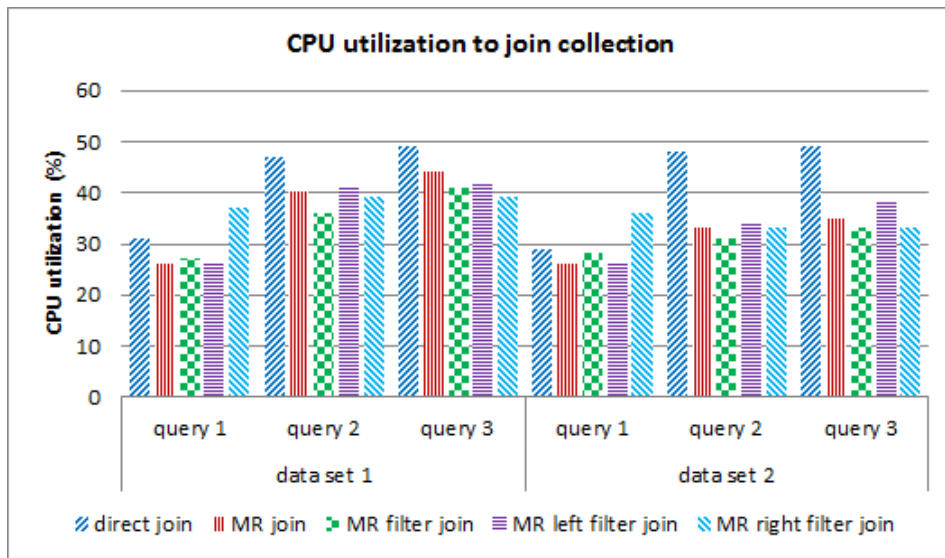
ภาพที่ 5 กราฟแสดงเวลาในการเชื่อมคอลเลคชันของข้อมูลชุดที่ 1



ภาพที่ 6 กราฟแสดงเวลาในการเชื่อมคอลเลคชันของข้อมูลชุดที่ 2



ภาพที่ 7 กราฟแสดงเวลาในการเชื่อมคอลเลคชันของข้อมูลชุดที่ 1 และชุดที่ 2



ภาพที่ 8 กราฟแสดงปริมาณการใช้งานซีพียูในการเชื่อมคอลเลคชัน