

การใช้แมพรีดิวซ์เชื่อมคอลเลกชันของฐานข้อมูลโนเอสคิวแอลบนมอดโกดีบี

Using MapReduce to join collection of NOSQL Database based on MongoDB

นิรุทธิ์ รวยริน (Nirut Ruayreun)* ดร.เกรียงไกร ปอแก้ว (Dr.Kriengkrai Porkaew)**

บทคัดย่อ

ในปัจจุบันปริมาณจำนวนข้อมูลที่จัดเก็บในฐานข้อมูลมีอัตราการเติบโตที่สูง เมื่อข้อมูลที่จัดเก็บมีจำนวนมากในการจัดเก็บข้อมูลลงฐานข้อมูลแบบเชิงสัมพันธ์นั้นจะทำให้การจัดการกับข้อมูลไม่มีประสิทธิภาพเท่าที่ควรเนื่องจากรูปแบบการจัดเก็บข้อมูลมีลักษณะโครงสร้างที่ซับซ้อนทำให้ต้องใช้เวลานานในการจัดการเข้าถึงข้อมูลที่มีปริมาณมาก จึงมีการนำเทคโนโลยีฐานข้อมูลโนเอสคิวแอลเข้ามาใช้ในการจัดเก็บและจัดการข้อมูลที่มีจำนวนมากนั้น การเก็บข้อมูลในฐานข้อมูลโนเอสคิวแอลนั้นจะเป็นการเก็บข้อมูลในลักษณะไม่มีความสัมพันธ์เชิงโครงสร้าง ซึ่งฐานข้อมูลโนเอสคิวแอลไม่รองรับการเชื่อมความสัมพันธ์ของข้อมูล งานวิจัยนี้จึงได้เสนอแนวคิดสร้างความสัมพันธ์ของข้อมูลในฐานข้อมูลโนเอสคิวแอลบนฐานข้อมูลมอดโกดีบี โดยนำเทคนิคหลักการเขียนโปรแกรมแบบแมพรีดิวซ์มาประยุกต์ใช้ในการเชื่อมความสัมพันธ์ข้อมูล และดำเนินการทดลองวัดผลเพื่อนำค่ามาวิเคราะห์หาค่าประสิทธิภาพในการจัดการข้อมูล

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 มอดโกดีบี ฐานข้อมูลโนเอสคิวแอล

Key Words: Join relationship, MongoDB, NOSQL database

* นักศึกษา หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

พระจอมเกล้าธนบุรี

** ผู้ช่วยศาสตราจารย์ คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

บทนำ

ในการสร้างความสัมพันธ์เชื่อมโยงระหว่างข้อมูลในฐานข้อมูลเชิงสัมพันธ์นั้น ทำให้ข้อมูลมีความถูกต้องสูงเพราะหลีกเลี่ยงการเก็บข้อมูลที่ซ้ำซ้อน สามารถจัดการในด้านการนำข้อมูลไปใช้เพื่อประมวลผลได้สะดวกในงานด้านต่างๆ ได้ง่ายและมีประสิทธิภาพ เนื่องจากฐานข้อมูลเชิงสัมพันธ์ถูกออกแบบมาให้เหมาะสมกับข้อมูลที่มีโครงสร้างค่อนข้างชัดเจน และยังรองรับการเชื่อมความสัมพันธ์ของข้อมูล ขณะที่ในยุคปัจจุบันนั้นปริมาณข้อมูลที่จัดเก็บในฐานข้อมูลมีอัตราการเติบโตที่สูงมากและมีการจัดเก็บข้อมูลที่หลากหลายในเชิงโครงสร้าง ดังนั้นจึงทำให้การบริหารจัดการกับข้อมูลทำได้ค่อนข้างยากและถ้าหากดำเนินการได้ก็มีความซับซ้อนในการจัดการข้อมูลที่มีปริมาณมาก การเข้าถึงข้อมูลก็จะช้าเพราะต้องเสียเวลาในการเชื่อมโยงข้อมูลที่แยกกันอยู่หลายส่วนทำให้ใช้ระยะเวลานาน

Sadalage and Fowler (2013) กล่าวว่า ในปีคริสต์ศักราช 1998 Carlo Strozzi ได้คิดค้นฐานข้อมูลเชิงสัมพันธ์โอเพนซอร์สที่มีการทำงานแบบเบา (Light) โดยออกแบบให้ฐานข้อมูลเก็บตารางเป็นแบบแฟ้มแอสกี (ASCII File) ข้อมูลแต่ละระเบียบ (Tuple) จะเป็นการเก็บแบบบรรทัดที่มีการแบ่งฟิลด์ข้อมูลด้วยแถบช่องว่าง (Tab) ในการค้นหาหรือกระทำการใดๆ 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 (2011) ได้กล่าวว่าฐานข้อมูลมอดโกดีบีเป็นฐานข้อมูลจัดอยู่ในกลุ่ม Document Data Model ของฐานข้อมูลโนเอสคิวแอล ซึ่งฐานข้อมูลมอดโกดีบี ถูกพัฒนาจากองค์กร 10gen เป็นฐานข้อมูลแบบโอเพนซอร์ส (Open Source) พัฒนาการจัดการข้อมูลด้วยภาษาซี ฐานข้อมูลมอดโกดีบี มีหลักการทำงานโดยแบ่งการจัดเก็บข้อมูลออกเป็น 3 ส่วนดังต่อไปนี้

ส่วนที่ 1 ฐานข้อมูล (Database) เป็นการจัดเก็บข้อมูลเป็นกลุ่ม โดยในฐานข้อมูลประกอบด้วยกลุ่มข้อมูลคอลเลกชัน เป็นการเก็บข้อมูลคอลเลกชันเดียวหรือรวบรวมหลายๆ คอลเลกชันเก็บไว้ในฐานข้อมูลได้

ส่วนที่ 2 คอลเลกชัน (Collection) เป็นการจัดเก็บข้อมูลเป็นกลุ่มซึ่งอยู่ภายใต้ฐานข้อมูล โดยคอลเลกชันประกอบด้วยกลุ่มข้อมูลคอคิวเมนต์ (Document) ซึ่งหลักการจัดเก็บของคอลเลกชันจะคล้ายกับการจัดเก็บแบบตารางในฐานข้อมูลเชิงสัมพันธ์ การจัดเก็บต่างกับตารางในฐานข้อมูลเชิงสัมพันธ์ตรงที่คอลเลกชันไม่ต้องสร้างโครงสร้างที่จัดเก็บข้อมูลที่แน่นอนก่อนนำค่าข้อมูลเก็บลงในฐานข้อมูล

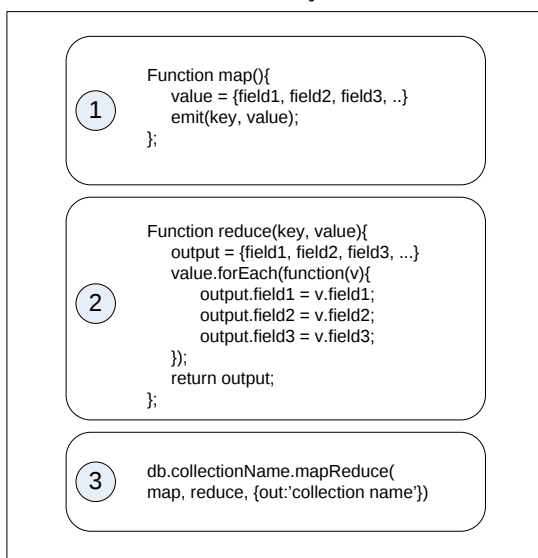
ส่วนที่ 3 คอคิวเมนต์ (Document) เป็นการจัดเก็บข้อมูลซึ่งอยู่ภายใต้คอลเลกชัน โดยในคอคิวเมนต์ประกอบไปด้วยฟิลด์และแวลู (field-and-value pairs) โดยฟิลด์ใช้ในการอ้างอิงและแวลูเป็นค่าของข้อมูล โดยที่แวลูสามารถเป็นข้อมูลประเภทต่างๆ ได้เช่น ตัวอักษร (String), ตัวเลขจำนวนเต็ม (Integer), ตัวเลขทศนิยม (Float), อาร์เรย์ (Array) เป็นต้น อีกทั้งในค่าแวลูยังสามารถเก็บค่าข้อมูลที่เป็นแบบคอคิวเมนต์ซ้อนลงไปได้อีกด้วย

ในงานวิจัยนี้ได้ศึกษาการจัดการข้อมูลของฐานข้อมูลโนเอสคิวแอล โดยศึกษาจากฐานข้อมูลมอดโกดีบีเนื่องจากฐานข้อมูลมอดโกดีบีสามารถเข้าถึงข้อมูลได้เร็วมาก ในข้อมูลกว่า 50 กิกะไบต์ ฐานข้อมูลมอดโกดีบีสามารถเข้าถึงข้อมูลได้เร็วกว่าฐานข้อมูลมายเอสคิวแอล ถึง 10 เท่า (Han, 2011) นอกจากนี้ฐานข้อมูลมอดโกดีบียังมีหลายองค์กรที่เลือกใช้ (MongoDB Inc, 2014) และศึกษาอัลกอริทึมที่ใช้จัดการข้อมูลปริมาณมากด้วยแมพรีดิวซ์

เทคนิคแมพรีดิวซ์ (MapReduce) เป็นแนวคิดวิธีการที่ใช้ในการค้นหาข้อมูลที่มีปริมาณมาก Shroff (2010) ได้กล่าวว่า รูปแบบการเขียนโปรแกรมตามแนวคิดแมพรีดิวซ์ ถูกพัฒนาขึ้นโดยองค์กรกูเกิล (Google) เพื่อเข้ามาช่วยในการค้นหาข้อมูลที่มีปริมาณมาก และเข้ามาช่วยในการประมวลผลข้อมูล บนที่เก็บข้อมูลเว็บขนาดใหญ่ ซึ่งข้อมูลเว็บเหล่านี้ถูกเก็บใน BigTable และระบบ

การเก็บไฟล์แบบกระจาย (Google File System) แนวคิด
การเขียนโปรแกรมแบบแมพรีดิวซ์ ถูกออกแบบเพื่อการ
ประมวลผลและสร้างข้อมูลที่มีขนาดใหญ่ด้วยวิธีการ
คำนวณแบบขนานพร้อมกันของหน่วยประมวลผล
(Processor) จำนวนหนึ่งพันหน่วยในเวลาเดียวกัน

ในปัจจุบันฐานข้อมูลโนเอสคิวแอลไม่รองรับการ
เชื่อมความสัมพันธ์ของข้อมูล ดังนั้นเพื่อเป็นการเพิ่ม
ความถูกต้องของข้อมูลและยังคงประสิทธิภาพด้าน
ความเร็วในการสอบถาม (Query) ข้อมูลของฐานข้อมูลโน
เอสคิวแอล ดังนั้นงานวิจัยนี้จึงได้เสนอแนวคิดสร้าง
ความสัมพันธ์ของฐานข้อมูลโนเอสคิวแอลขึ้นมา โดยใช้
ฐานข้อมูลมอโกดีบี และใช้หลักการเขียนโปรแกรมแบบ
แมพรีดิวซ์มาประยุกต์ใช้ สำหรับการวัดค่าประสิทธิภาพ
นั้นวัดจากการทดลองหาค่าเวลาที่ใช้ในการสอบถาม
ข้อมูลจนได้ผลลัพธ์ที่ต้องการ ระหว่างวิธีการเชื่อมคอล
เลกชัน โดยใช้แมพรีดิวซ์กับวิธีการเชื่อมคอลเลกชันโดย
การเขียนโปรแกรมเพื่อเชื่อมข้อมูลเอง



รูปที่ 1 กระบวนการเชื่อมความสัมพันธ์ของข้อมูลใน
รูปแบบฟังก์ชันแมพรีดิวซ์

จากรูปที่ 1 เป็นกระบวนการทำงานของฟังก์ชัน
แมพรีดิวซ์ ในการเชื่อมความสัมพันธ์ของข้อมูล
ระหว่างคอลเลกชันมีขั้นตอนดังนี้

ขั้นตอนที่ 1 การสร้างฟังก์ชันแมพ เป็นการสร้าง
ฟังก์ชันแมพขึ้นมาให้ทำงานกับคอลเลกชันแรก โดยมี
อาร์กิวเมนต์ (Argument) 2 ค่าคือคีย์ (Key) และแวลู
(Value) ค่า อาร์กิวเมนต์ แวลู ที่ถูกส่งโดยฟังก์ชัน Emit
นั้น จะเป็นเอาต์พุต (Output) ที่มีฟิลด์ของข้อมูลตรง
ตามความต้องการใช้งาน หากฟิลด์ของข้อมูลที่ต้องการอยู่
คอลเลกชันที่สอง ให้ทำการนำคีย์นอก (Foreign Key)
ของข้อมูลภายใต้อาร์กิวเมนต์คีย์นั้น ของคอลเลกชัน
แรกไปค้นหาในคอลเลกชันที่สองก่อน แล้วนำผลลัพธ์
ที่ได้จากการค้นหามารวมเป็นเอาต์พุตเดียว เพื่อให้
รูปแบบของฟังก์ชัน Emit มีลักษณะเป็นคู่ของคีย์
และแวลู

ขั้นตอนที่ 2 การสร้างฟังก์ชันรีดิวซ์ เป็นการสร้าง
ฟังก์ชันรีดิวซ์ขึ้นมาให้มีการรับพารามิเตอร์ 2 ค่าคือ คาคีย์
และค่าแวลู สร้างตัวแปรที่จะเป็นเอาต์พุตให้ประกอบ
ไปด้วยชื่อฟิลด์ที่ต้องการ ต่อมานำค่าจากพารามิเตอร์
แวลูมาเขียนโปรแกรมวนลูปเพื่อกำหนดค่าให้กับฟิลด์ที่
ต้องการเป็นลักษณะการเตรียมข้อมูลทั้งหมดกำหนดไว้
ในตัวแปรที่เป็นเอาต์พุตเพื่อใช้แสดงผลลัพธ์

ขั้นตอนที่ 3 เป็นรูปแบบการเรียกใช้คำสั่ง
แมพรีดิวซ์ของมอโกดีบีให้ทำงาน เมื่อคำสั่งแมพรีดิวซ์
ประมวลผลการทำงาน ตามฟังก์ชันแมพและฟังก์ชัน
รีดิวซ์เสร็จ จะมีการสร้างผลลัพธ์ออกมาอยู่ในรูปของ
คอลเลกชันชั่วคราว (Temp Collection)

ในงานวิจัยของ Calil (2012) นั้นได้นำเสนอวิธีการ
เชื่อมความสัมพันธ์ของข้อมูลบนฐานข้อมูลที่ชื่อว่า
SimpleDB ที่เป็นบริการฐานข้อมูลของ Amazons เพื่อใช้
จัดการข้อมูลบนระบบคลาวด์ ซึ่งเป็นฐานข้อมูลโนเอสคิว
แอลแบบ Document Data Model ชนิดหนึ่ง ด้วยวิธีการ
สร้างอินเตอร์เฟซขึ้นมาให้รองรับการเขียนคำสั่งสอบถาม
ข้อมูลด้วยภาษาเอสคิวแอล ซึ่งประกอบไปด้วยหลักการ
ทำงาน 3 ส่วนดังนี้

ส่วนที่ 1 Access interface ทำหน้าที่รับคำสั่งที่เป็น
ภาษาเอสคิวแอล โดยคำสั่งภาษาเอสคิวแอลที่ใช้เชื่อม
ความสัมพันธ์ของข้อมูลจะเป็นแบบ INNER JOIN

ส่วนที่ 2 Command decomposition ทำหน้าที่แยกคำสั่งเอสคิวแอลที่รับมาให้เป็นคำสั่งที่ง่าย ไม่มีการเชื่อมความสัมพันธ์ของข้อมูล ด้วยวิธี regular expression และแปลงคำสั่งนั้นให้อยู่ในรูปแบบของภาษาสอบถามของ SimpleDB

ส่วนที่ 3 Processing and Return ทำหน้าที่ส่งคำสั่งที่แยกออกเป็นส่วนๆ ไปสอบถามข้อมูลที่ SimpleDB เมื่อได้ผลลัพธ์กลับมาแล้วจะต้องนำผลลัพธ์แต่ละส่วนมารวมกันเพื่อส่งกลับไปยังผู้ใช้

วัตถุประสงค์การวิจัย

1. เพื่อศึกษารูปแบบจัดเก็บข้อมูลของฐานข้อมูลโนเอสคิวแอลและการจัดการข้อมูลของฐานข้อมูลมอโกดีบี
2. เพื่อศึกษารูปแบบอัลกอริทึมที่ใช้ในการจัดการข้อมูลปริมาณมากด้วยแมพรีดิวซ์
3. เพื่อศึกษาและเปรียบเทียบผลลัพธ์ที่ได้จากการพัฒนาโปรแกรมเชื่อมคอลเลกชันของฐานข้อมูลมอโกดีบีกับวิธีการเขียนโปรแกรมเชื่อมคอลเลกชันเอง

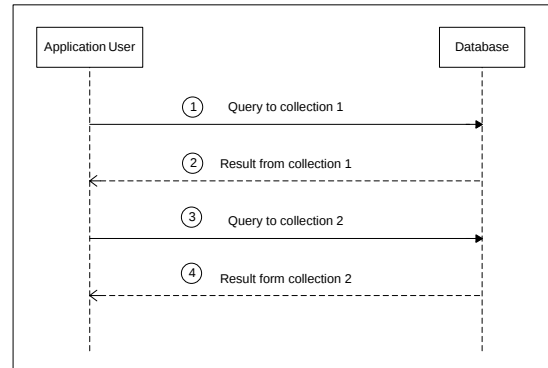
วิธีการวิจัย

ในงานวิจัยนี้ได้ทำการศึกษาและเปรียบเทียบการเชื่อมคอลเลกชันของฐานข้อมูลมอโกดีบี ที่ได้มีการออกแบบโครงสร้างข้อมูลให้มีความสัมพันธ์กัน โดยการเปรียบเทียบนั้นจะทำการเปรียบเทียบเวลาที่ใช้ในการเชื่อมคอลเลกชันเพื่อให้ได้ข้อมูลตามที่ต้องการ ด้วยวิธีเขียนเชื่อมคอลเลกชันในโปรแกรมประยุกต์ (Application) เองกับวิธีที่มีการพัฒนาโปรแกรมให้เป็นไลบรารีที่มีการประยุกต์ใช้แมพรีดิวซ์ในการเชื่อมคอลเลกชันและเพิ่มขั้นตอนการกรองข้อมูลก่อนที่จะเชื่อมคอลเลกชันด้วยแมพรีดิวซ์

โดยงานวิจัยนี้ได้ทำการศึกษาและเสนอวิธีการเชื่อมคอลเลกชันด้วยกันทั้งหมด 5 รูปแบบ ดังนี้

รูปแบบที่ 1 join direct เป็นวิธีที่ใช้หลักการสอบถามข้อมูลจากคอลเลกชันแรกมาเก็บไว้ใน

โปรแกรม แล้ววนลูปนำคีย์นอกแต่ละคีย์ไปสอบถามข้อมูลในคอลเลกชันที่สอง



รูปที่ 2 ขั้นตอนการเชื่อมคอลเลกชันแบบ direct join

จากรูปที่ 2 สามารถอธิบายขั้นตอนการทำงานได้ดังนี้

ขั้นตอนที่ 1 โปรแกรมฝั่งไคลเอนต์ส่งคำสั่งเพื่อสอบถามข้อมูลที่ต้องการ ซึ่งอยู่ในคอลเลกชันที่หนึ่งไปยังฐานข้อมูลมอโกดีบี

ขั้นตอนที่ 2 ฐานข้อมูลมอโกดีบีทำการประมวลผลคำสั่งสอบถามข้อมูลที่ได้รับมาและส่งผลลัพธ์กลับไปยังโปรแกรมประยุกต์ของผู้ใช้งาน

ขั้นตอนที่ 3 เมื่อโปรแกรมประยุกต์ฝั่งไคลเอนต์ได้รับผลลัพธ์จากฐานข้อมูลมอโกดีบี โปรแกรมจะทำการวนลูปข้อมูลโดยนำค่าฟิลด์ที่เป็นคีย์นอกแต่ละคีย์ส่งไปสอบถามข้อมูลในคอลเลกชันที่สองที่ฐานข้อมูลมอโกดีบี

ขั้นตอนที่ 4 ฐานข้อมูลมอโกดีบีประมวลผลคำสั่งสอบถามข้อมูลและส่งผลลัพธ์กลับไปยังโปรแกรมประยุกต์ฝั่งไคลเอนต์

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 เป็นค่าของชื่อคอลเลกชันทั้งสองที่
ต้องการเชื่อมข้อมูลเข้าด้วยกัน มีรูปแบบคือ collection
name1, collection name2 เช่น student, advisor

ค่าที่ 3 sWhere เป็นค่าที่ระบุข้อผิดพลาดที่เป็นคีย์เพื่อ
ใช้เชื่อมสองคอลเลกชัน มีรูปแบบคือ

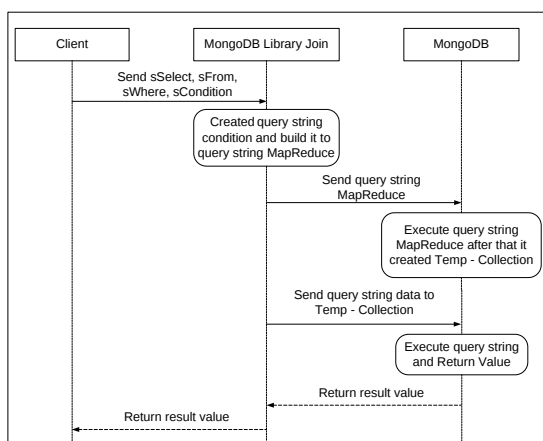
collection name1.key_id = collection name2.key_id
เช่น student.advisor_id = advisor.advisor_id

ค่าที่ 4 sCondition เป็นค่าของเงื่อนไขในการ
สอบถามข้อมูล 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 ทั้งนี้เพื่อให้การเรียกชื่อ
โปรแกรมที่พัฒนาขึ้นเป็นไปในรูปแบบเดียวกัน

รูปแบบที่ 2 MR join ใช้หลักการเชื่อมคอลเลกชัน
ของข้อมูลด้วย MongoDB Library Join ที่ใช้แมพรีดิวซ์



รูปที่ 3 ขั้นตอนการเชื่อมคอลเลกชันของข้อมูลด้วย
MongoDB Library Join ที่ใช้แมพรีดิวซ์

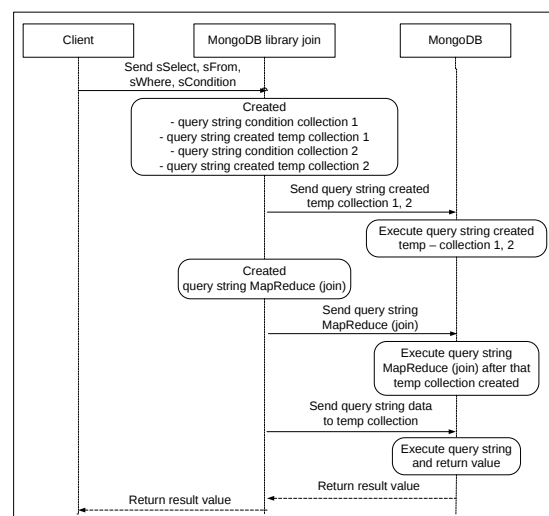
จากรูปที่ 3 สามารถอธิบายขั้นตอนการทำงานได้
ดังนี้

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

ขั้นตอนที่ 2 กระบวนการของ MongoDB Library
Join เมื่อได้รับค่าข้อมูลตามพารามิเตอร์ที่ระบุ จะนำค่า
ข้อมูล sCondition มาสร้าง query string condition ซึ่ง
เป็นเงื่อนไขในการสอบถามข้อมูลโดยสร้างให้อยู่ใน
รูปแบบคำสั่งสอบถามที่ใช้ในฐานข้อมูลมอโกดีบี
จากนั้นกระบวนการต่อมาจะนำค่าข้อมูล sSelect,
sFrom, sWhere และ query string condition นำค่าข้อมูล
ทั้งหมดมาสร้างเป็น query string MapReduce และทำ
การนำ query string MapReduce ส่งไปประมวลผลที่
ฐานข้อมูลมอโกดีบี ซึ่งผลลัพธ์ที่ได้จากแมพรีดิวซ์ จะ
อยู่ในรูปแบบคอลเลกชันชั่วคราว

ขั้นตอนที่ 3 MongoDB Library Join จะส่งคำสั่ง
สอบถามข้อมูลไปที่คอลเลกชันชั่วคราว และส่งผลลัพธ์
กลับไปยังโปรแกรมฝั่งไคลเอนต์

รูปแบบที่ 3 MR filter join ใช้หลักการเชื่อมคอล
เลกชันของข้อมูลด้วย MongoDB Library Join ที่มีการ
กรองข้อมูลทั้งสองคอลเลกชันมาสร้างเป็นคอลเลกชัน
ชั่วคราวก่อนแล้วใช้แมพรีดิวซ์



รูปที่ 4 ขั้นตอนการเชื่อมคอลเลกชันแบบ MR filter join

จากรูปที่ 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 ของคอลเลกชันที่หนึ่ง แล้วนำ query string condition นี้ไปประมวลผลที่ฐานข้อมูลมอดโกดบี เพื่อสร้างคอลเลกชันชั่วคราวของคอลเลกชันที่หนึ่ง และต่อมาจะทำการสร้าง query string condition ของคอลเลกชันที่สอง แล้วนำ query string condition นี้ไปประมวลผลที่ฐานข้อมูลมอดโกดบี เพื่อสร้างคอลเลกชันชั่วคราวของคอลเลกชันที่สอง กระบวนการต่อมาจะนำค่าข้อมูล sSelect, sWhere, ชื่อของคอลเลกชันชั่วคราวที่หนึ่งและชื่อคอลเลกชันชั่วคราวที่สอง นำค่าข้อมูลทั้งหมดมาสร้างเป็น query string MapReduce และทำการนำ query string MapReduce นี้ส่งไปประมวลผลที่ฐานข้อมูลมอดโกดบี ซึ่งผลลัพธ์ที่ได้จากแมพรีดิวซ์จะอยู่ในรูปแบบคอลเลกชันชั่วคราว

ขั้นตอนที่ 3 MongoDB Library Join จะประมวลผลคำสั่งสอบถามข้อมูลไปที่คอลเลกชันชั่วคราวและส่งผลลัพธ์กลับไปยังโปรแกรมฝั่งไคลเอนต์

รูปแบบที่ 4 MR left filter join ใช้หลักการเชื่อมคอลเลกชันของข้อมูลด้วย MongoDB Library Join ที่มีการกรองข้อมูลของคอลเลกชันแรกมาสร้างเป็นคอลเลกชันชั่วคราวก่อนเท่านั้นแล้วใช้แมพรีดิวซ์ โดยขั้นตอนการทำงานจะคล้ายกับรูปแบบ MR filter join เพียงแต่การสร้างคอลเลกชันชั่วคราวจะสร้างจากเงื่อนไขที่มาจากคอลเลกชันที่หนึ่งเท่านั้น โดยกระบวนการเชื่อมคอลเลกชันของแมพรีดิวซ์จะใช้คอลเลกชันชั่วคราวที่สร้างขึ้นกับคอลเลกชันที่สองที่ไม่มีการกรองข้อมูล

รูปแบบที่ 5 MR right filter join ใช้หลักการเชื่อมคอลเลกชันของข้อมูลด้วย MongoDB Library Join ที่มี

การกรองข้อมูลของคอลเลกชันที่สองมาสร้างเป็นคอลเลกชันชั่วคราวก่อนเท่านั้นแล้วใช้แมพรีดิวซ์ โดยขั้นตอนการทำงานจะคล้ายกับรูปแบบ MR filter join เพียงแต่การสร้างคอลเลกชันชั่วคราวจะสร้างจากเงื่อนไขที่มาจากคอลเลกชันที่สองเท่านั้น โดยกระบวนการเชื่อมคอลเลกชันของแมพรีดิวซ์จะใช้คอลเลกชันชั่วคราวที่สร้างขึ้นกับคอลเลกชันที่หนึ่งที่ไม่มีการกรองข้อมูล

ผลการวิจัย

การวัดค่าเวลาที่ใช้ในการเชื่อมคอลเลกชันนั้น ผู้วิจัยได้จำลองสภาพแวดล้อมของการทดสอบ โดยใช้เครื่องคอมพิวเตอร์ 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 ใช้ฐานข้อมูลมอดโกดบี เวอร์ชัน 2.2.6 โปรแกรมที่ใช้วัดค่าเวลา WAPT 8.1 ข้อมูลที่ใช้ทดสอบมี 2 คอลเลกชัน คือคอลเลกชัน student: 16 fields, 9,665 documents คอลเลกชัน advisor: 11 fields, 5,954 documents ทั้งสองคอลเลกชันใช้อินเด็กซ์ที่มอดโกดบีสร้างให้ ไม่มีการสร้างอินเด็กซ์เพิ่มเติม และพัฒนาโปรแกรมสำหรับทดสอบจำนวน 5 รูปแบบ

การทดลองเชื่อมคอลเลกชันแบ่งเป็น 2 ชุด โดยแต่ละชุด มี 3 คิวรี เพื่อใช้สำหรับทดสอบการเชื่อมคอลเลกชันทั้ง 5 แบบ โดยให้คอลเลกชัน student ใช้แต่ละคำสั่งสอบถามที่มีเงื่อนไขกรองข้อมูลได้ต่างกัน ส่วนคอลเลกชัน advisor ทั้ง 3 คำสั่งสอบถาม กรองข้อมูลได้เท่ากันในแต่ละชุด ดังแสดงในตารางที่ 1 และตารางที่ 2

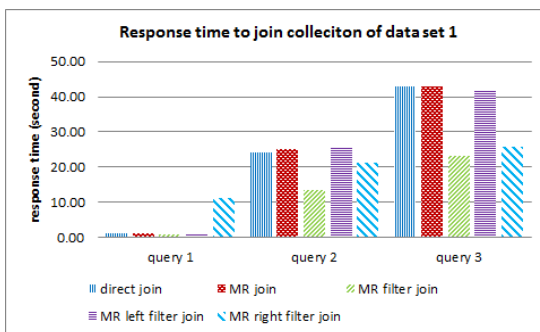
ตารางที่ 1 แสดงจำนวนคอคิวเมนต์แต่ละเงื่อนไข ชุดที่ 1

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)

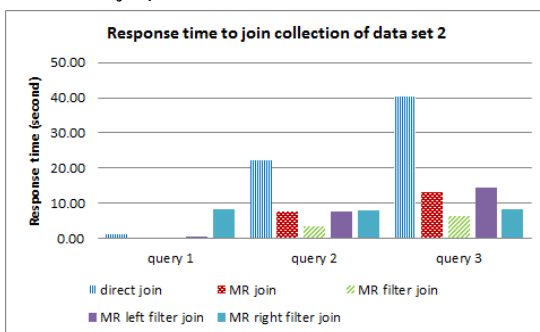
ตารางที่ 2 แสดงจำนวนคอคิวเมนต์แต่ละเงื่อนไข ชุดที่ 2

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)

จากการทดลองเชื่อมคอลเลกชันทั้ง 5 รูปแบบ เพื่อเปรียบเทียบเวลาในการเชื่อมข้อมูลได้ผลการทดลอง โดยแบ่งเป็นผลการทดลองจากข้อมูลชุดที่ 1 และชุดที่ 2

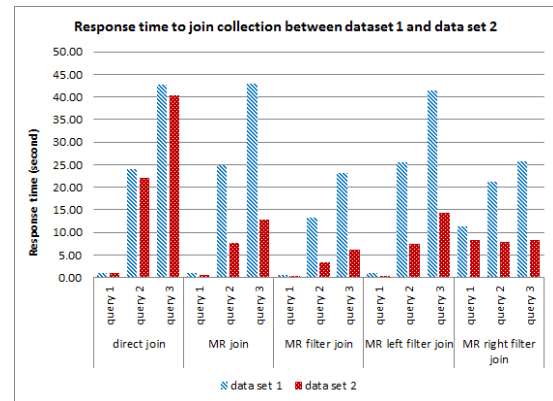


รูปที่ 5 กราฟแสดงเวลาในการเชื่อมคอลเลกชันของข้อมูลชุดที่ 1



รูปที่ 6 กราฟแสดงเวลาในการเชื่อมคอลเลกชันของข้อมูลชุดที่ 2

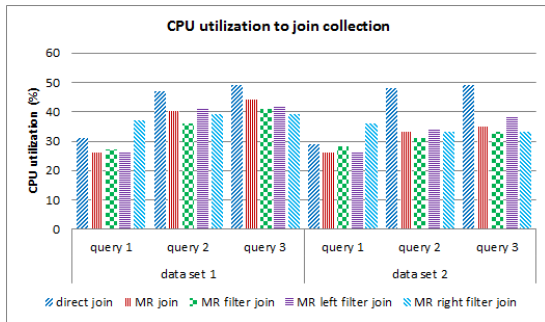
จากรูปที่ 5 และรูปที่ 6 จะเห็นได้ว่าหากข้อมูลจากเงื่อนไขในการเชื่อมคอลเลกชันสามารถกรองข้อมูลได้มากจะทำให้การเชื่อมคอลเลกชันทำได้ค่อนข้างเร็วแต่หากเงื่อนไขในการเชื่อมคอลเลกชันกรองข้อมูลได้น้อยวิธี MR filter join จะช่วยให้การเชื่อมคอลเลกชันทำได้ค่อนข้างเร็ว



รูปที่ 7 กราฟแสดงเวลาในการเชื่อมคอลเลกชันของข้อมูลชุดที่ 1 และชุดที่ 2

จากรูปที่ 7 เป็นการเปรียบเทียบเวลาที่ใช้ในการเชื่อมคอลเลกชันของข้อมูลชุดที่ 1 และชุดที่ 2 ของวิธีการเชื่อมคอลเลกชันแต่ละแบบในแต่ละคำสั่งสอบถาม จะเห็นได้ว่าข้อมูลชุดที่ 2 ซึ่งเงื่อนไขสามารถกรองข้อมูลได้มากกว่าชุดที่ 1 เมื่อนำมาใช้กับวิธี MR filter join จะทำให้การเชื่อมคอลเลกชันได้เร็วกว่าวิธีอื่น

การเชื่อมคอลเลกชันด้วยวิธี direct join นอกจากจะใช้เวลาในการเชื่อมคอลเลกชันนานแล้ว ยังใช้พื้นที่มากกว่าวิธีอื่นอีกด้วย เนื่องจากช่วงของการวนลูปนำค่าคีย์นอกแต่ละคีย์ไปค้นหาข้อมูลที่อยู่ในคอลเลกชันที่สองซึ่งจะต้องมีการติดต่อกับฐานข้อมูลตลอด จนกว่าคีย์ที่ใช้ในการค้นหาข้อมูลจะหมดลง



รูปที่ 8 กราฟแสดงปริมาณการใช้งานซีพียูในการเชื่อมคอลเลกชัน

จากรูปที่ 8 เป็นการเปรียบเทียบปริมาณการใช้จ่ายเพื่อเชื่อมคอลเลกชันแต่ละแบบตามแต่ละคำสั่งสอบถามจะเห็นว่าวิธี direct join จะใช้ซีพียูมากกว่าวิธีอื่น

อภิปรายและสรุปผลการวิจัย

ในบทความงานวิจัยนี้ได้นำเสนอแนวคิดและวิธีการเชื่อมคอลเลกชันของฐานข้อมูลมอโกดีบีซึ่งไม่รองรับการเชื่อมคอลเลกชันก็เพื่อเป็นการนำข้อดีของฐานข้อมูลโนเอสคิวแอลในเรื่องการอ่านและเขียนข้อมูลที่เร็วมาพร้อมกับแนวคิดของฐานข้อมูลเชิงสัมพันธ์ที่ลดความซ้ำซ้อนของข้อมูลโดยการแยกตารางซึ่งในฐานข้อมูลมอโกดีบีคือการแยกคอลเลกชันและประยุกต์ใช้วิธีการประมวลผลข้อมูลด้วยวิธีแมพรีดิวซ์มาช่วยในการเชื่อมคอลเลกชันของฐานข้อมูลมอโกดีบีโดยเพิ่มขั้นตอนการกรองข้อมูลของคอลเลกชัน แล้วนำมาสร้างเป็นคอลเลกชันชั่วคราวเพื่อใช้ในการเชื่อมคอลเลกชัน

จากการทดลองจะเห็นได้ว่าการสร้างคอลเลกชันชั่วคราวขึ้นมาก่อนที่จะเชื่อมกันนั้นจะทำให้การเชื่อมคอลเลกชันทำได้เร็วขึ้น ในวิธี MR filter join จะเป็นการสร้างคอลเลกชันชั่วคราวขึ้นมาทั้งสองคอลเลกชัน วิธี MR left filter join จะสร้างคอลเลกชันชั่วคราวของพารามิเตอร์ sCondition ลำดับแรก วิธี MR right filter join จะสร้างคอลเลกชันชั่วคราวของพารามิเตอร์ sCondition ลำดับสอง แม้ว่าการทดลองจะเห็นว่าวิธี MR filter join ใช้เวลาน้อยสุด ก็ไม่ได้หมายความว่า

ดีกว่าวิธี 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 ใช้เวลาในการเชื่อมคอลเลกชันใกล้เคียงกัน

ในส่วนของวิธี MR join จะไม่มีการสร้างคอลเลกชันชั่วคราวขึ้นมาก่อนแต่จะใช้แมพรีดิวซ์อย่างเดียว ดังนั้นถ้าหากเงื่อนไขในการเชื่อมคอลเลกชันกรองข้อมูลได้น้อยก็จะใช้เวลาที่ค่อนข้างนานเพราะหลักแมพรีดิวซ์จะอ่านข้อมูลทั้งหมดในคอลเลกชันสำหรับวิธี direct join จะเป็นการเขียนโปรแกรมเองซึ่งก็หลักการกรองข้อมูลจากคอลเลกชันที่หนึ่งก่อนหลังจากนั้นจะนำคีย์ออก แต่ละคีย์วนกลับไปสอบถามในคอลเลกชันที่สอง ซึ่งวิธีนี้ใช้เวลาค่อนข้างนานและเปลืองทรัพยากรซีพียู แต่ถ้าเงื่อนไขกรองข้อมูลได้น้อยการใช้ซีพียูของแต่ละแบบก็ไม่ได้แตกต่างกันมากจนเป็นนัยสำคัญ

แนวคิดและวิธีที่นำเสนอในงานวิจัยนี้เป็นแนวทางหนึ่งที่จะเชื่อมคอลเลกชันกรณีการออกแบบข้อมูลที่มีความสัมพันธ์กันแต่อยู่คนละคอลเลกชันให้สามารถเชื่อมกันได้ สำหรับการพัฒนางานวิจัยนี้ในอนาคตอาจพัฒนาให้โปรแกรมมีการตัดสินใจได้ว่ากรองข้อมูลเพื่อสร้างคอลเลกชันชั่วคราว หรือไม่ โดยถ้าเงื่อนไขกรองข้อมูลได้น้อยก็ไม่จำเป็นต้องสร้างคอลเลกชันชั่วคราวให้เสียเวลา

กิตติกรรมประกาศ

ข้าพเจ้าในฐานะผู้วิจัยขอขอบคุณ คุณศิริธนา นิลน้อย ที่ให้คำแนะนำและเป็นกำลังใจในการทำวิจัยครั้งนี้ ผู้วิจัยขอกราบขอบพระคุณอย่างสูงต่อคุณบิดา มารดา ที่ได้มอบชีวิต และอบรมเลี้ยงดู สนับสนุน ช่วยเหลือผู้วิจัยในทุกๆด้านโดยตลอด สุดท้ายขอขอบคุณครูบาอาจารย์ทุกท่านที่ให้ความรู้ ญาติมิตร เพื่อนทุกท่านที่ให้การสนับสนุนช่วยเหลือ ความคิดอันได้เกิดจากงานวิจัยในครั้งนี้ ผู้วิจัยขออุทิศให้แก่บุคคลผู้มีพระคุณดังกล่าวทุกท่าน

เอกสารอ้างอิง

- Pramod J. Sadalage and Martin Fowler. 2013. NoSQL distilled : a brief guide to the emerging world of polyglot, 2nded. United States of America: Pearson Education Inc.
- Rubayeet Islam. 2011. PHP and MongoDB Web Development Beginner's Guide. Birmingham: Packt Publishing.
- Jing Han. 2011. Survey on NoSQL Database. Pervasive Computing and Applications (ICPCA), 2011 6th International Conference: 363-366.
- MongoDB Inc. Customer Success Stories. Retrieved January 27, 2014, from <http://www.mongodb.com/customers>.
- Gautam Shroff. 2010. Enterprise Cloud Computing Technology Architecture Application. United Kindom: Cambride University Press.
- Andre Calil, Ronaldo dos Santos Mello. 2012. SimpleSQL: A Relational Layer for SimpleDB. ADBIS 2012, LNCS 7503, 2012: 99-110.