การพัฒนาการถ่ายโอนและสอบถามข้อมูลในรูปแบบอาร์ดีเอฟบนกรอบการทำงานฮาดูป

นางสาวจุฑามาศ กะวิเศษ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2558

A DEVELOPMENT OF RDF DATA TRANSFER AND QUERY ON HADOOP FRAMEWORK

Miss Jutamard Kawises

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science Program in Software Engineering

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2015

หัวข้อวิทยานิพนธ์	การพัฒนาการถ่ายโอนและสอบถามข้อมูลในรูปแบบอาร์ดีเอฟบนกรอบการทำงานฮาดูป
โดย	นางสาวจุฑามาศ กะวิเศษ
สาขาวิชา	วิศวกรรมซอฟต์แวร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รองศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวุฒิ

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

.....คณบดีคณะวิศวกรรมศาสตร์

(รองศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ

(รองศาสตราจารย์ ดร.ทวิตีย์ เสนีวงศ์ ณ อยุธยา)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

(รองศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวุฒิ)

.....กรรมการ

(รองศาสตราจารย์ ดร.พรศิริ หมื่นไชยศรี)

.....กรรมการภายนอกมหาวิทยาลัย

(ผู้ช่วยศาสตราจารย์ ดร.มฑุปายาส ทองมาก)

จุฑามาศ กะวิเศษ : การพัฒนาการถ่ายโอนและสอบถามข้อมูลในรูปแบบอาร์ดีเอฟบนกรอบการทำงานฮาดูป (A DEVELOPMENT OF RDF DATA TRANSFER AND QUERY ON HADOOP FRAMEWORK) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: รศ. ดร.วิวัฒน์ วัฒนาวุฒิ, 48 หน้า.

ข้อมูลอาร์ดีเอฟที่ถูกเก็บไว้ในรูปแบบของเอ็กซ์เอ็มแอลหรือระบบฐานข้อมูลเชิงสัมพันธ์โดยในปัจจุบันเป็นที่นิยมนำมาประยุกต์ใช้ในการเก็บข้อมูลต่างๆ ที่มีขนาดใหญ่มากๆ อย่างไรก็ตามเมื่อข้อมูลมีแนวโน้มเพิ่มขึ้น ส่งผลให้เซตของข้อมูลมีขนาดใหญ่ขึ้นตามไปด้วย ดังนั้นทางเลือกในการจัดการข้อมูลและการค้นหาข้อมูลอาร์ดีเอฟ หรือข้อมูลที่มีความเชื่อมโยงกันที่เรียกว่าลิงค์เดต้าคือการใช้อัลกอริทึมของแมปรีดิวซ์ บนกรอบการทำงานของฮาดูป วิทยานิพนธ์นี้จึงนำเสนอการดำเนินการถ่ายโอนข้อมูลและการค้นหาข้อมูลอาร์ดีเอฟจากฮาดูปคลัสเตอร์ เพื่อวัดประสิทธิภาพด้านเวลาในการเข้าถึงข้อมูลและค้นหาข้อมูลบนฮาดูป โดยข้อมูลอาร์ดีเอฟขนาดใหญ่ที่ใช้ในการทดลองจะถูกแปลงให้อยู่ในรูปของเอ็นทริปเปิ้ล และถูกถ่ายโอนเข้าไปยังฮาดูปคลัสเตอร์ซึ่งเป็นแหล่งเก็บข้อมูลของฮาดูปซึ่งอาศัยหลักการของเอชดีเอฟเอส ในการแบ่งข้อมูลขนาดใหญ่เพื่อจัดเก็บเข้าสู่ระบบ การค้นหาข้อมูลอาร์ดีเอฟในระบบโดยใช้สปาร์เคิล ซึ่งจะถูกแปลงให้อยู่ในรูปแบบของการสอบถามแบบเอ็นทริปเปิ้ล ที่เรียกว่า เบสิคกราฟแพทเทิร์น ด้วยจีน่าอัลจีบร้า เพื่อส่งเข้าไปประมวลผลในอัลกอริทึมของแมปรีดิวซ์ เพื่อให้ได้ผลลัพธ์สุดท้ายที่ตรงกับความต้องการของการค้นหาข้อมูล

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต .....

สาขาวิชา วิศวกรรมซอฟต์แวร์ ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

ปีการศึกษา 2558

5570970021 : MAJOR SOFTWARE ENGINEERING
KEYWORDS: BIG DATA / APACHE HADOOP / MAPREDUCE / SPARQL / RDF / N-TRIPLE / LINKED DATA / HDFS

JUTAMARD KAWISES: A DEVELOPMENT OF RDF DATA TRANSFER AND QUERY ON HADOOP FRAMEWORK. ADVISOR: ASSOC. PROF. WIWAT VATANAWOOD, Ph.D.,, 48 pp.

An RDF graph is typically stored in an XML file or a relational database. However, when it becomes a large RDF graph, an alternative way to handle the storing and query RDF graph or linked data is to use the MapReduce algorithm and Hadoop framework. In this thesis, we propose a supporting tool far data transfer and query on big RDF graph. We aim to reduce the access time and query response time by using Hadoop Framework. The RDF/XML or linked data are converted into a huge set of N-triples and they are uploaded onto Hadoop and stored in data nodes of Hadoop Distributed File System (HDFS). The query of RDF graph in SPARQL is analyzed and converted into a specific N-triple format as to search the answer using Jena Algebra. The MapReduce algorithm is developed to relevantly manipulate the RDF graph.

Department: Computer Engineering Student's Signature .....
Field of Study: Software Engineering Advisor's Signature .....
Academic Year: 2015

กิตติกรรมประกาศ

ขอกราบขอบพระคุณ รองศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวุฒิ ที่คอยให้คำแนะนำในการเขียนวิทยานิพนธ์และผลงานทางวิชาการ และกรุณาสละเวลาในการแก้ไขบทความทางวิชาการเพื่อให้สามารถส่งบทความได้ทันเวลา และช่วยผลักดันให้วิทยานิพนธ์เล่มนี้สำเร็จได้

ขอกราบขอบพระคุณ รองศาสตราจารย์ ดร.ทวิตีย์ เสนีวงศ์ ณ อยุธยา รองศาสตราจารย์ ดร.พรศิริ หมื่นไชยศรี และ ผู้ช่วยศาสตราจารย์ ดร.มฑุปายาส ทองมาก ที่กรุณาสละเวลาให้คำแนะนำในการสอบโครงร่างวิทยานิพนธ์ ตลอดไปจนถึงการสอบวิทยานิพนธ์

ขอกราบขอบพระคุณ นางวัชรี กะวิเศษ มารดาของข้าพเจ้าที่เป็นแรงสนับสนุนหลักของในการศึกษาครั้งนี้ และเพื่อนทุกคนที่คอยให้ความช่วยเหลือที่พักพิง สถานที่ทำวิทยานิพนธ์ และแรงสนับสนุนด้านต่างๆ ไปตลอดจนการสอบถามความคืบหน้าวิทยานิพนธ์และให้ความช่วยเหลือในทุกๆ เรื่อง ซึ่งเป็นส่วนหนึ่งของแรงกระตุ้นให้วิทยานิพนธ์นี้สำเร็จได้ และขอขอบคุณ นายชัยณรงค์ อมรเพชรสถาพร ที่คอยช่วยเหลือในการศึกษาครั้งนี้ด้วย

สารบัญ

หน้า

หน้า

บทที่ 1

บทนำ

1.1. ที่มาและความสำคัญของปัญหา

การจัดการข้อมูลบนซีเมนติกเว็บ (Semantic Web) ที่มีความสัมพันธ์กันในรูปแบบลิงค์ของข้อมูลในปัจจุบันมีขนาดใหญ่และมีแนวโน้มเพิ่มมากขึ้น ซึ่งเป็นตัวบ่งชี้สำคัญของปัญหาสำหรับการจัดการข้อมูลจำนวนมาก[1] โดยแต่ละโหนดข้อมูลจะเชื่อมโยงเข้าด้วยกัน การเชื่อมโยงดังกล่าวมีวัตถุประสงค์เพื่อใช้ในการเข้าถึงแหล่งข้อมูลสารสนเทศในวงกว้างขององค์ความรู้ที่สนใจ เทคโนโลยีสำคัญที่สนับสนุนข้อมูลดังกล่าวคือ ยูอาร์ไอ (URIs - Uniform Resource Identifier) ในรูปแบบของลิงค์เดต้า (Linked Data) เกิดการเชื่อมโยงของข้อมูลในระดับต่างๆ มีความซับซ้อนของข้อมูล สามารถทำการค้นหาข้อมูลที่เกี่ยวข้องกันได้ง่ายขึ้น แต่เมื่ออัตราการเพิ่มของข้อมูลที่มีอยู่กลับเพิ่มขึ้นเป็นทวีคูณตามความต้องการของผู้ใช้ ทำให้การเข้าถึงข้อมูลในระดับต่างๆ ที่มีความซับซ้อนเกิดการค้นหาแบบย้อนไปย้อนมา เช่น การค้นหาคำสำคัญหลัก ไปหาคำสำคัญย่อยที่เชื่อมต่อกัน และกลับไปค้นหาคำสำคัญหลักอีกครั้ง ซึ่งเกิดจากข้อมูลที่มีความสัมพันธ์เพิ่มขึ้น ทำให้การค้นหาข้อมูลมีประสิทธิภาพในด้านเวลาลดลง เมื่อเทียบกับอัตราการเติบโตของข้อมูลที่มีอยู่ อย่างไรก็ตามการค้นหาข้อมูลบนซีเมนติกเว็บ ที่มีจำนวนของข้อมูลขนาดใหญ่โดยใช้กฎการอนุมาน ยังคงเป็นปัญหาเมื่อผลลัพธ์ที่ได้จากการค้นหานั้นไม่ตรงกับขอบเขตความต้องการของผู้ใช้ เนื่องจากการสร้างกฎนั้นมีความซับซ้อนและยุ่งยาก [1]

วิทยานิพนธ์นี้ขอนำเสนอการถ่ายโอนข้อมูลและสอบถามข้อมูล เพื่อการเข้าถึงข้อมูลได้อย่างมีประสิทธิภาพในด้านความเร็วเพิ่มขึ้น เนื่องจากฮาดูป (Hadoop) ทำงานบนระบบแบบกระจายซึ่งสามารถทำงานแบบขนานรองรับการประมวลผลข้อมูลขนาดใหญ่[2] โดยการนำข้อมูลจากลิงค์เดต้าในรูปแบบของ อาร์ดีเอฟ เอ็กซ์เอ็มแอล (RDF/XML) เข้าสู่โปรแกรมแปลงข้อมูลที่พัฒนาขึ้นด้วยภาษาจาวาที่เทียบเคียงประสิทธิภาพด้านความถูกต้องของข้อมูลที่นำมาแปลงให้อยู่ในรูปแบบของเอ็นทริปเปิ้ล (N-Triple) เพื่อนำข้อมูลเข้าสู่ฮาดูปโดยใช้ เอชดีเอฟเอช (HDFS - Hadoop Distributed File System) เพื่อเป็นตัวแบ่งข้อมูลที่ต้องการเก็บออกเป็นส่วนย่อยๆ แล้วกระจายส่วนย่อยๆ เหล่านั้นไปยังคลัสเตอร์ (Cluster) ที่มีอยู่ในระบบ[3] และการใช้สปาร์เคิล (SPARQL) เพื่อการสอบถามข้อมูลจากฮาดูป โดยนำหลักการของแมปรีดิวซ์ (MapReduce) ในการค้นหาข้อมูลในฮาดูป ซึ่งจะช่วยเพิ่มประสิทธิภาพการค้นหาข้อมูลในแง่ของความเร็วในการค้นหาข้อมูลในระบบ แล้วนำผลลัพธ์ที่ได้จากการค้นหาข้อมูลไปประมวลผลและได้ผลลัพธ์สุดท้ายเป็นไฟล์เอ็กซ์เอ็มแอล

1.2. วัตถุประสงค์งานวิจัย

พัฒนาการถ่ายโอนข้อมูลจากลิงค์เดต้าในรูปแบบของอาร์ดีเอฟเอ็กซ์เอ็มแอลไปยังเดต้าโหนดของฮาดูป และการสืบค้นข้อมูลจากฮาดูปโดยใช้สปาร์เคิล

1.3. ขอบเขตงานวิจัย

1) ชุดข้อมูลที่นำมาใช้ในการทดสอบเป็นข้อมูลการทดสอบเพียง 1 ชุดข้อมูล เป็นข้อมูลทดลองที่ชื่อว่า The Lehigh University Benchmark (LUBM)[4]
2) ไฟล์ข้อมูลเข้าเป็นไฟล์ในรูปแบบอาร์ดีเอฟเอ็กซ์เอ็มแอล
3) ผลลัพธ์สุดท้ายของการถ่ายโอนและสอบถามข้อมูลเป็นไฟล์
4) การค้นหาข้อมูลบนฮาดูปใช้ชุดคำสั่งสปาร์เคิล
5) รูปแบบสปาร์เคิล ใช้ในการทดสอบการเข้าถึงข้อมูล
6) การตรวจสอบความครบถ้วนของข้อมูลนำเข้าโดยการเปรียบเทียบขนาดของไฟล์ข้อมูลก่อนและหลังนำข้อมูลเข้า
7) ข้อมูลนำเข้าจะเปรียบเทียบความถูกต้องของข้อมูลจากการสอบถามข้อมูลจากลิงค์เดต้าและการสอบถามข้อมูลจากฮาดูปว่าข้อมูลที่ได้มีเนื้อหาข้อมูลที่ตรงกัน
8) ข้อมูลนำเข้าจะไม่มีการประเมินประสิทธิภาพด้านความเร็วของการแปลงและการถ่ายโอนข้อมูล
9) การประเมินและวัดประสิทธิภาพด้านความเร็วการเข้าถึงข้อมูลและสอบถามข้อมูลในฮาดูป โดยเปรียบเทียบเวลาที่ใช้ในการค้นหาข้อมูลกับชุดข้อมูลที่ใช้ทดสอบในงานวิจัย โดยใช้รูปแบบของสปาร์เคิลที่ได้กำหนดไว้เพื่อทำการค้นหาข้อมูลระหว่างลิงค์เดต้าและฮาดูป

1.4. ประโยชน์ที่คาดว่าจะได้รับ

1) สามารถถ่ายโอนข้อมูลจากอาร์ดีเอฟเดต้า (RDF Data) เข้าสู่ฮาดูป
2) สามารถสอบถามข้อมูลจากฮาดูปโดยใช้สปาร์เคิลได้
3) ข้อมูลจากไฟล์ที่นำเข้าสู่ฮาดูปเป็นข้อมูลพร้อมใช้งาน สำหรับการค้นหาข้อมูลในฮาดูป
4) เพิ่มประสิทธิภาพการเข้าถึงข้อมูลจากฮาดูป เมื่อเทียบการการเข้าถึงข้อมูลจากลิงค์เดต้า

1.5. ขั้นตอนการดำเนินงาน

1) ศึกษาหลักการทำงานและโครงสร้างของฮาดูป
2) ศึกษาหลักการทำงานและโครงสร้างของลิงค์เดต้า
3) ศึกษาหลักการทำงานและโครงสร้างของสปาร์เคิล
4) ศึกษางานวิจัยที่เกี่ยวข้องกับฮาดูป ลิงค์เดต้าและอาร์ดีเอฟ

5) พัฒนาการถ่ายโอนข้อมูล เพื่อนำข้อมูลจากลิงค์เดต้าเข้าสู่ฮาดูป
6) ทดสอบการถ่ายโอนข้อมูลและการสอบถามข้อมูลที่พัฒนาขึ้น
7) ตรวจสอบความถูกต้องของข้อมูลที่เข้าสู่ระบบผ่านกระบวนการการถ่ายโอนข้อมูลพัฒนาขึ้น
8) สรุปผลการทดสอบ และประเมินผลการทดสอบระบบ
9) จัดทำรายงานการวิจัย และรูปเล่มวิทยานิพนธ์

1.6. บทความที่ตีพิมพ์จากงานวิจัย

ส่วนหนึ่งของวิทยานิพนธ์นี้ ได้รับการตีพิมพ์ เป็นบทความวิชาการ ดังนี้

1) เรื่อง “A DEVELOPMENT OF RDF DATA TRANSFER AND QUERY ON HADOOP” โดย จุฑามาศ กะวิเศษ และ วิวัฒน์ วัฒนาวุฒิ ในงานประชุมวิชาการ 15th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2016) จัดโดย IEEE Computer Society and International Association for Computer and Information Science (ACIS) เมื่อวันที่ 26 – 29 มิถุนายน 2559 ณ เมืองโอกายามะ ประเทศญี่ปุ่น

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1. กรอบการทำงานฮาดูป (Hadoop Framework)

ฮาดูปเป็นเทคโนโลยีแบบโอเพนซอร์ส (Open source) ที่ออกแบบโดยยาฮู (Yahoo) และถูกนำไปใช้ในกูเกิ้ลไฟล์ซิสเทม (GoogleFS) เพื่อทำงานออกแบบระบบไฟล์ข้อมูลและวิเคราะห์ข้อมูลบนระบบคอมพิวเตอร์แบบกระจายสำหรับข้อมูลขนาดใหญ่ ที่มีความเสถียรสูง และสนับสนุนการทำงานแบบขนาน โดยจะทำหน้าที่เป็น หน่วยความจำแบบกระจาย (Distributed Storage) ซึ่งสามารถรองรับการจัดการกับข้อมูลไม่มีโครงสร้าง หรือข้อมูลที่มีขนาดใหญ่ โดยมีชุดคำสั่งเพื่ออำนวยความสะดวกผ่าน คลัสเตอร์หรือเดต้าโหนด (Datanode) เพื่อการค้นหาและการเข้าถึงข้อมูล
ตัวอย่างของผู้ใช้งานอาร์ปาเช่ฮาดูป (Apache Hadoop) มีดังนี้

(1) เฟสบุ๊ค (Facebook) มี ฮาดูปคลัสเตอร์ จำนวน 2 ชุด ชุดแรกประกอบจากเซิร์ฟเวอร์ (Server) จำนวน 1,100 เครื่อง, ซีพียู (CPU) 8,800 คอร์(Core), หน่วยความจำ 12 PB (12,000TB) ชุดที่สองประกอบจาก เซิร์ฟเวอร์ จำนวน 300 เครื่อง, ซีพียู 2,400 คอร์, หน่วยความจำ 3 PB (3,000TB)

(2) ยาฮูใช้เซิร์ฟเวอร์มากกว่า 40,000 เครื่อง, ซีพียู มากกว่า 100,000 ชุด สำหรับรองรับระบบแอด (Ads) และเว็บเสิร์ช (Web Search) [3]

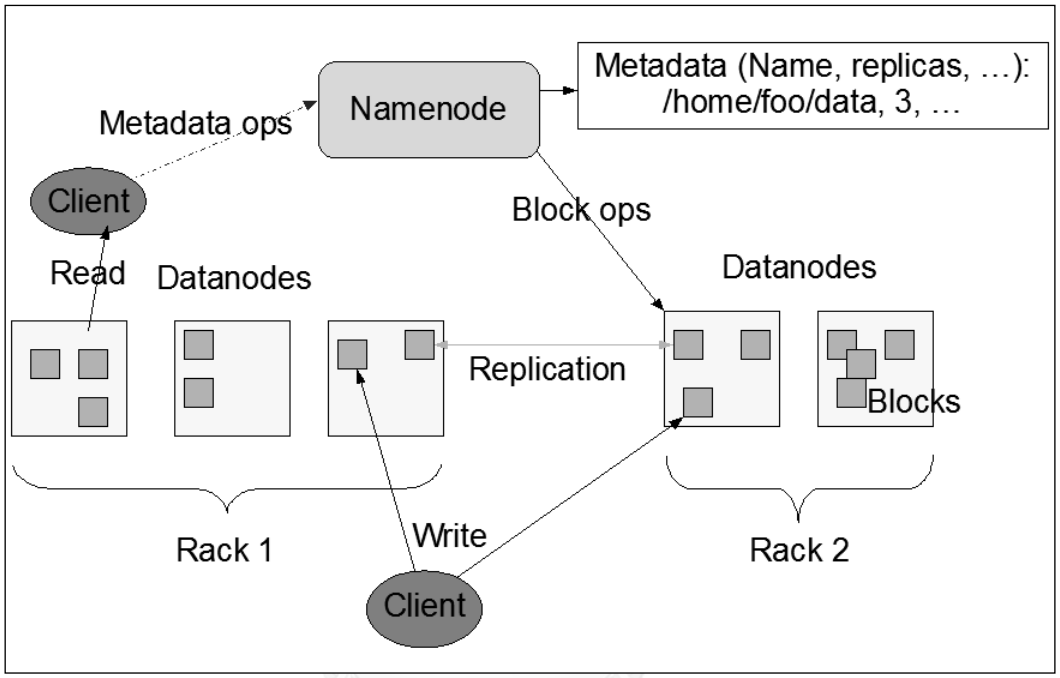
2.1.1. การทำงานของฮาดูป

จากกรอบการทำงานของฮาดูปสามารถแบ่งการทำงานเป็น 2 ส่วน คือ

1) ระบบแฟ้มข้อมูลแบบกระจายของฮาดูป (HDFS - Hadoop Distributed File System) เป็นระบบจัดเก็บข้อมูลบนฮาดูป หลักการทำงานคือ การแตกข้อมูลออกเป็นส่วนย่อยๆ เป็นบล๊อค (Block) ข้อมูลตามขนาดของบล๊อคที่กำหนดไว้ แล้วกระจายข้อมูลในบล๊อคนั้น ไปยังคลัสเตอร์ต่างๆ นอกจากนี้ยังมีการสำรองข้อมูลในระบบให้อัตโนมัติ สถาปัตยกรรมของเอชดีเอฟเอชตามรูปที่ 2.1 แบ่งเซิร์ฟเวอร์ ออกเป็น 2 แบบ คือ เนมโหนด (Namenode) และ เดต้าโหนด (Datanode) ดังนี้ [5]

(1) เนมโหนด หรือ มาสเตอร์โหนด (Namenode or Master Server) ทำหน้าที่เป็นโหนดที่มาสเตอร์เซิร์ฟเวอร์ (Master Server) จะทำการจัดการกับเนมสเปช (Namespace) ของแฟ้มข้อมูลใน เอชดีเอฟเอช เช่น การเข้าถึงข้อมูลของไฟล์ ที่อยู่ของไฟล์ ชื่อของไฟล์ข้อมูล นอกจากนี้ยังทำหน้าที่เป็นคีย์ที่สามารถบอกได้ว่าข้อมูลดังกล่าวเก็บไว้ที่เดต้าโหนดตัวที่เท่าไหร่

(2) เดต้าโหนด หรือ คลัสเตอร์เซิร์ฟเวอร์ หรือ สเลฟเซิร์ฟเวอร์ (Datanode or Cluster Server or Slave Server) เป็นส่วนที่เก็บข้อมูล สามารถมีได้มากกว่า 1 โหนด ใน 1 คลัสเตอร์ ทำหน้าที่จัดการเกี่ยวกับเนื้อที่เก็บข้อมูลของเครื่องที่มีในระบบ และบริการการเข้าถึงข้อมูลตามคำร้องขอ จากไคลแอนท์ (Client) และจัดการทำสำเนาบล๊อคตามคำสั่งของเนมโหนดซึ่งจะเก็บไว้ที่เดต้าโหนดเดียวกัน แต่เกิดเป็นอีกบล๊อค



รูปที่ 2.1 โครงสร้างของ เอชดีเอฟเอช [5]

2) แมปรีดิวซ์ (MapReduce)

เป็นกระบวนการทำงานหลัก ของการเขียนโปรแกรมเพื่อประมวลผลข้อมูล แล้วกระจายคำสั่งไปยัง คลัสเตอร์เซิร์ฟเวอร์ (Cluster Server) ในระบบ แล้วทำการประมวลผลพร้อมๆ กัน ช่วยลดเวลาในการประมวลผล แบ่งเป็น 2 ส่วน ส่วนแรกเรียกว่า แมป ส่วนที่สองเรียกว่า รีดิวซ์ หลักการทำงานของแมปรีดิวซ์ ตามรูปที่ 2.2 แบ่งการทำงานเป็น 2 ส่วน คือ แมป (Map) และ รีดิวซ์ (Reduce) ดังนี้

(1) แมป คือการส่งคำสั่งไปยังเครื่องเดต้าโหนด (Datanode) ต่างๆ ในระบบแบบมัลติเลเวลทรี (multi level Tree) เมื่อเครื่องเดต้าโหนดได้รับคำสั่งการร้องขอการค้นหาข้อมูล จะทำการค้นหาข้อมูลในเดต้าโหนดแต่ละตัวที่มีอยู่ในระบบ โดยอาศัยคีย์ที่ส่งเป็นค่าพารามิเตอร์เข้ามาในโปรแกรมของแมปซึ่งหลักการอ่านข้อมูลแบบ ทีละแถว (Row-by-Row) เมื่อพบข้อมูลจะทำการจับคู่คีย์ที่เป็นพารามิเตอร์และ Value ที่ต้องการ ส่งไปยังส่วนของเดต้าโหนดหรือคลัสเตอร์เซิร์ฟเวอร์สามารถเขียนออกมาในรูปแบบ

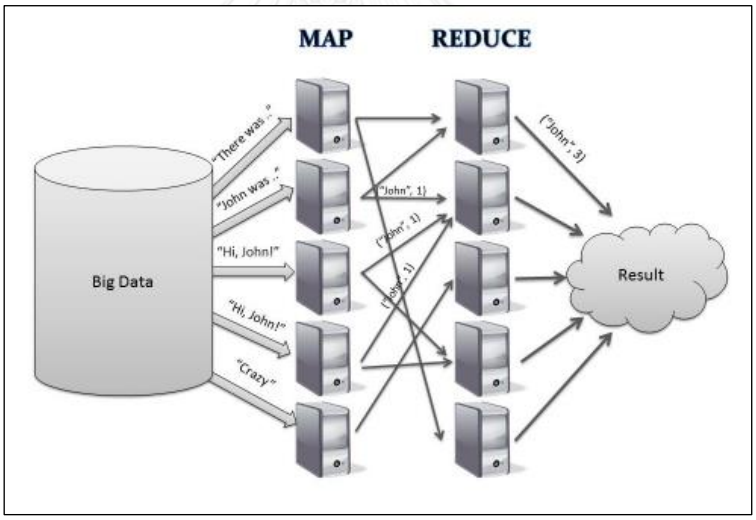
domainMap (k1, v1) ⟶ list (K2, v2)

(2) รีดิวซ์ เป็นการประมวลข้อมูลหรือวิเคราะห์ข้อมูลที่ได้จากการค้นหา โดยจะรับผลลัพธ์ที่ได้จากแมป ซึ่งข้อมูลจะถูกจัดการอย่างอิสระ แล้วนำมาเรียงลำดับหรือ การวิเคราะห์ผลลัพธ์ แล้วนำผลลัพธ์มารวมกันในส่วนรีดิวซ์ เพื่อส่งผลลัพธ์สุดท้ายออกไปโดยที่สามารถกำหนดเครื่องที่ต้องการทำรีดิวซ์เองได้คือ ชุดของค่าที่มีคีย์ตัวเดียวกัน ซึ่งเป็นชุดที่เล็กกว่าชุดของค่าทั้งหมด นั่นคือ

domainReduce (K2, list (v2)) ⟶ list (v3)

2.1.2. ข้อดีของฮาดูป

1) มีความยืดหยุ่นสูง
2) สามารถเพิ่มหรือลดจำนวนโหนดได้ตามต้องการ
3) สามารถเขียนข้อมูลได้อย่างรวดเร็ว
4) สามารถปรับเปลี่ยนตัวแปรให้สัมพันธ์กับระบบได้ง่าย



รูปที่ 2.2 แสดงหลักการทำงานของแมปรีดิวซ์ [6]

2.2. ลิงค์เดต้า (Linked Data)

ข้อมูลขนาดใหญ่ในองค์ความรู้ที่มีความสัมพันธ์และเชื่อมโยงกันด้วยลิงค์เพื่อแสดงให้เห็นถึงความหมายและลักษณะของแต่ละโหนดที่เกี่ยวข้องกัน “หลักการของการสร้าง ลิงค์เดต้า ก็คือ การกำหนดค่ายูอาร์ไอให้กับอ๊อบเจกต์” [7] ซึ่งสนับสนุนการใช้งานข้อมูลซ้ำ ช่วยลดความซ้ำซ้อนของข้อมูล

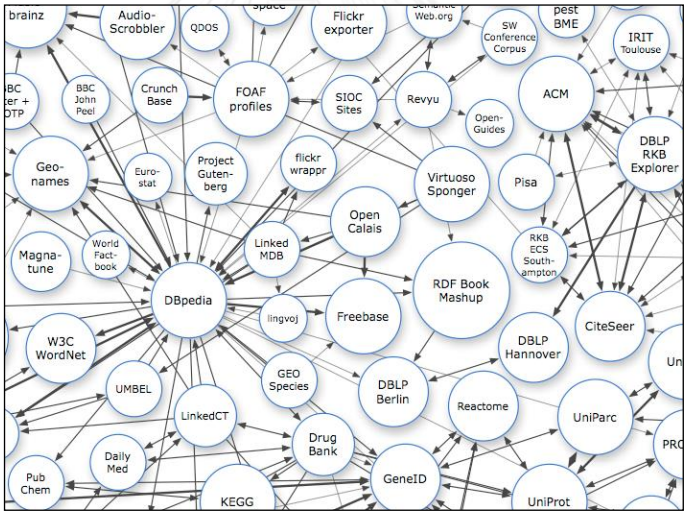
2.2.1. หลักการของลิงค์เดต้า

ใช้ยูอาร์ไอเป็นชื่อของสิ่งของทุกสิ่ง กล่าวได้โดยง่ายคือใช้ยูอาร์ไอ แทนไอดี (ID) ของสิ่งของต่างๆ ใช้ เอชทีทีพี ยูอาร์ไอ (Http URIs) เพื่อให้ค้นหาข้อมูลด้วยโปรโตคอลเอชทีทีพี การค้นหาสิ่งใดสิ่งหนึ่ง จะต้องแสดงข้อมูลโดยการใช้เทคนิคมาตรฐาน ใช้ลิงค์ในการเชื่อมโยงไปยังข้อมูลอื่นๆ เพื่อเพิ่มความสามารถของการค้นหาข้อมูลในวงกว้าง[8]

2.2.2. ข้อดีของ ลิงค์เดต้า

1) ควบคุมการทำงานได้ง่ายเนื่องจากทำงานภายใต้ระบบการกระจาย
2) สามารถเชื่อมโยงและอ้างอิงข้อมูลที่มีอยู่แล้วผ่านทางยูอาร์ไอ
3) มีโครงสร้างแบบหลวมๆ
4) สามารถรวมข้อมูลที่เพิ่มเข้ามากับข้อมูลเดิมได้ง่าย

จากรูปที่ 2.3 แสดง ลิงค์เดต้าบนเว็บ พบว่า ลิงค์เดต้าเป็นหัวใจสำคัญของซีแมนติกเว็ปที่เกือบทุกการใช้งานบนเว็บ มีการเรียกใช้ ลิงค์เดต้า



รูปที่ 2.3 ลิงค์เดต้าบนซีเมนติกเว็บข้อมูลอ้างอิงเมื่อกันยายน ปี 2011 [9]

จากตารางที่ 2.1 แสดงปริมาณของเซตของข้อมูลแบ่งตามประเภทของกลุ่มข้อมูลระหว่างปี 2011 ปี 2014 พบว่าแนวโน้มการเติบโตของข้อมูลในโดเมนต่างๆ ในปี 2014 มีปริมาณข้อมูลเพิ่มขึ้นจากปี 2011 เป็นเท่าตัว

ตารางที่ 2.1 ปริมาณของเซตของข้อมูลแบ่งตามประเภทกลุ่มของข้อมูล[10]

Topic	2014		2011	
	Datasets	%	Datasets	%
Government	183	18.05%	49	42.09 %
Publications	96	9.47%	87	9.33 %
Life sciences	83	8.19%	41	9.60 %
User-generated content	48	4.73%	20	0.42 %
Cross-domain	41	4.04%	41	13.23 %
Media	22	2.17%	25	5.82 %
Geographic	21	2.07%	31	19.43 %
Social web	520	51.28%	N/A	N/A
Total	1014		294	

2.3. อาร์ดีเอฟ (RDF - Resource Description Framework)

เป็นภาษามาตรฐานที่อธิบายโครงสร้างการเก็บข้อมูลที่มีความสัมพันธ์กัน ซึ่งเป็นเทคโนโลยีที่ช่วยในการสนับสนุนซีแมนติกเว็ป สามารถอธิบายลักษณะและความสัมพันธ์ มีความยืดหยุ่นสูงมาก เนื่องจากเป็นการเก็บข้อมูลแบบ ประธาน – คุณสมบัติ – ค่าคุณสมบัติ (Subject – Predicate – Object) เรียกว่า อาร์ดีเอฟทริปเปิ้ล (RDF Triples) ซึ่งสามารถเขียนอธิบายลักษณะของข้อมูลได้ 2 แบบคือ อาร์ดีเอฟกราฟ และ อาร์ดีเอฟ เอ็กซ์เอ็มแอล ดังนี้

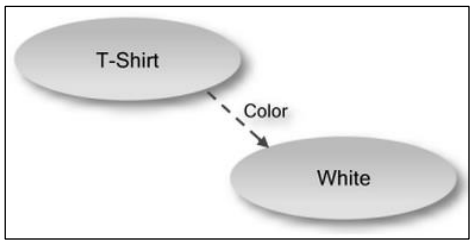
1) อาร์ดีเอฟกราฟ (RDF graph)

จะประกอบด้วย โหนด (Node) ซึ่งแทนด้วยรูปวงรี และมีตัวหนังสือกำกับ จะเชื่อมโยงกันด้วยลิงค์ ซึ่งแทนด้วยเส้นประที่มีหัวลูกศรกำกับ และมีตัวหนังสืออธิบายลักษณะของอ๊อบเจกต์

ตัวอย่างการเก็บข้อมูลแบบอาร์ดีเอฟกราฟ

จากประโยค : “T-Shirt has White Color ”

สามารถเขียนให้อยู่ในรูป RDF triples : (T-Shirt , color , White) และเขียนในรูปของ RDF Graph ได้ดังรูปที่ 2.4



รูปที่ 2.4 ตัวอย่างอาร์ดีเอฟกราฟของอาร์ดีเอฟทริปเปิ้ล

2) อาร์ดีเอฟ เอ็กซ์เอ็มแอล

เป็นการจัดเก็บข้อมูลของอาร์ดีเอฟทริปเปิ้ล ในรูปแบบของเอ็กซ์เอ็มแอล ดังตัวอย่างของอาร์ดีเอฟ เอ็กซ์เอ็มแอล ตามรูปที่ 2.5 โดยมีส่วนประกอบที่สำคัญ ดังนี้

(1) <rdf:RDF> : RDF root node tag

(2) <rdf:Description rdf:about> : เป็นแท็ก (tag) ของส่วนคำสั่ง (Statement) ที่แสดงถึงประธาน ของอาร์ดีเอฟสามารถใส่ได้มากกว่า 1 ส่วนคำสั่ง ภายใต้ประธานเดียวกัน

(3) <Predicates> : เป็นแท็กที่อยู่ภายใต้ <rdf:Description rdf:about> ที่เป็นประธาน ซึ่งสามารถใส่แหล่งกำเนิด (resource) หรือ ตัวกำหนดค่า (properties) ได้ และสามารถอ้างอิงไปยังส่วนคำสั่งอื่นได้ด้วย

```
1  <?xml version="1.0" encoding="UTF-8" ?>
2  <rdf:RDF
3  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
5  xmlns:owl="http://www.w3.org/2002/07/owl#"
6  xmlns:ub="http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#">
7
8  <owl:Ontology rdf:about="">
9  <owl:imports rdf:resource="http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl" />
10 </owl:Ontology>
11
12 <ub:University rdf:about="http://www.University0.edu">
13    <ub:name>University0</ub:name>
14 </ub:University>
15
16 <ub:Department rdf:about="http://www.Department0.University0.edu">
17    <ub:name>Department0</ub:name>
18    <ub:subOrganizationOf>
19       <ub:University rdf:about="http://www.University0.edu" /> </ub:subOrganizationOf>
20 </ub:Department>
21 </rdf:RDF>
```

รูปที่ 2.5 ตัวอย่างข้อมูลในรูปแบบของอาร์ดีเอฟเอ็กซ์เอ็มแอล [11]

2.4. สปาร์เคิล

สปาร์เคิล เป็นภาษามาตรฐานที่กำเนิดจากดับบลิวทรีซี (W3C) ใช้สำหรับสอบถามข้อมูลที่เก็บในรูปแบบของอาร์ดีเอฟมีการเข้าถึงข้อมูลโดยอาศัยโครงสร้างของอาร์ดีเอฟเอ็นทริปเปิ้ล โดยมีผลลัพธ์ของการค้นหาในรูปแบบของเอ็กซ์เอ็มแอล ซึ่งแบ่งเป็น 2 ส่วน คือ ส่วนซีเลค (SELECT) และส่วนแวร์(WHERE) ดังนี้ [12]

1) ส่วนของซีเลคเป็นค่าตัวแปรที่ต้องการค้นหาข้อมูลโดยใช้คำนำหน้า (Prefix) “?” แล้วตามด้วยชื่อตัวแปรที่ต้องการสอบถามข้อมูล

2) ส่วนของแวร์เป็นเงื่อนไขในการค้นหาข้อมูล

รูปแบบของ สปาร์เคิล

```
PREFIX <สำหรับกำหนดการอ้างอิงข้อมูล OWL>
SELECT ?variableNameList
WHERE {
      Basic Graph Pattern
}
```

ตัวอย่าง สปาร์เคิล

```
PREFIX w: <http://xmlns.com/foaf/0.1/>
SELECT ?mbox
WHERE {
  ?mbox w:family_name “Smith”
}
```

2.5. งานวิจัยที่เกี่ยวข้อง

2.5.1. งานวิจัย : การออกแบบและพัฒนาระบบค้นหาข้อมูลจราจรทางคอมพิวเตอร์ด้วยวิธีแมปรีดิวซ์ บนกรอบการทำงานของ ฮาดูป โดย ชูพันธ์ รัตนโภคา ปี 2012 [13]

งานวิจัยนี้นำเสนอ การออกแบบและพัฒนาระบบค้นหาข้อมูลจราจรทางคอมพิวเตอร์ และใช้วิธี แมปรีดิวซ์ ในการค้นหาข้อมูลโดยระบบมีส่วนติดต่อกับผู้ใช้งานผ่านโปรแกรมที่พัฒนาขึ้นด้วยภาษาจาวา และทดลองระบบค้นหาข้อมูลดังกล่าวว่าการใช้แมปรีดิวซ์ ทำให้การค้นหาข้อมูลมีประสิทธิภาพทางด้านความเร็วในการค้นหามากยิ่งขึ้นหรือไม่

การออกแบบระบบค้นหาข้อมูลจราจรทางคอมพิวเตอร์ ได้นำ เอชดีเอฟเอช และ แมปรีดิวซ์ ซึ่งทำงานบนฮาดูป มาช่วยในการจัดเก็บและค้นหาข้อมูล โดยผู้ใช้สามารถใช้งานผ่านทางส่วนติดต่อกับผู้ใช้งาน ซึ่งมีการเรียกใช้งานของ ฮาดูป และเชื่อมต่อกับ เอชดีเอฟเอช บนระบบปฏิบัติการลีนุ๊ก (Linux) โดยมีรูปแบบของแฟ้มข้อมูลที่ใช้ในงานวิจัย คือ แฟ้มข้อมูลจราจรทางคอมพิวเตอร์ ซึ่งเป็นแฟ้มข้อมูลที่บันทึกรายละเอียดการติดต่อสื่อสารระหว่างคอมพิวเตอร์ผ่านระบบเครือข่าย และส่วนที่ติดต่อกับผู้ใช้สามารถค้นหาข้อมูล โดยสามารถใส่เงื่อนไขของข้อมูลที่ต้องการค้นหาหลักๆ คือ

1) วัน – เวลาเริ่มต้น

2) วัน – เวลาสิ้นสุด

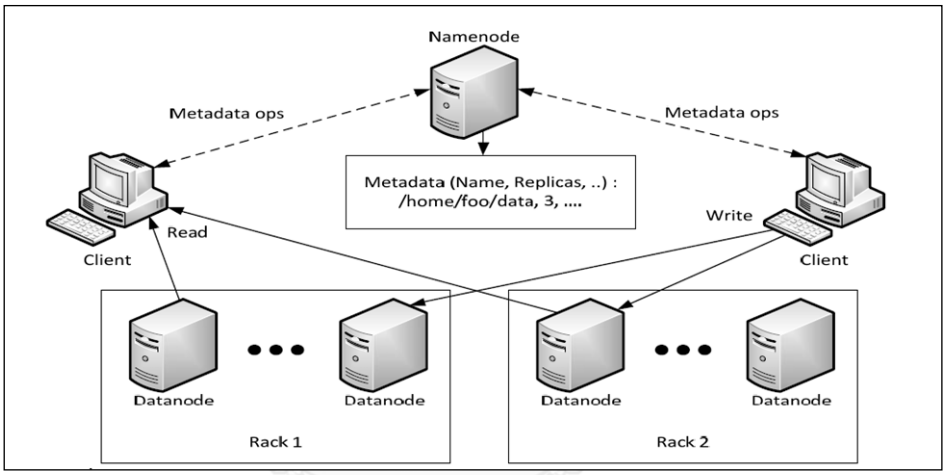
3) แนท ไอพี (NAT IP)

4) ยูอาร์แอล ไอพี (URL IP)

5) เซฟ (Save)

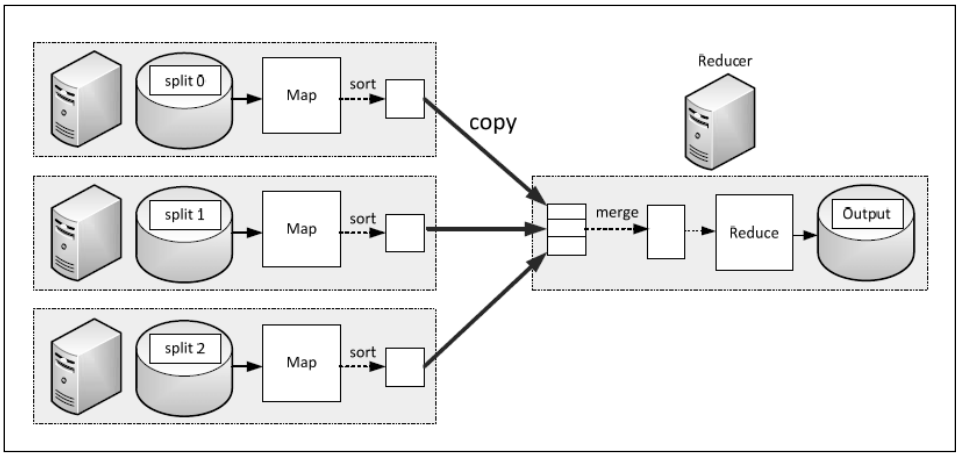
6) ชื่อไฟล์ (Filename)

การนำเข้าข้อมูลสู่เอชดีเอฟเอช จะใช้คำสั่งในรูปแบบของชุดคำสั่ง (Command Line) ที่มาพร้อมกับเอชดีเอฟเอช จากนั้นระบบจะทำการแตกไฟล์ข้อมูลดังกล่าวเป็นบล๊อคแล้วกระจายไปยังเครื่องคอมพิวเตอร์เพื่อเก็บข้อมูลโดยที่โครงสร้างของ เอชดีเอฟเอช ประกอบไปด้วยเนมโหนด ทำหน้าที่จัดการเกี่ยวกับแฟ้มข้อมูล และ เดต้าโหนดทำหน้าที่จัดการกับเนื้อที่ในการเก็บข้อมูลบล๊อคลงในแต่ละเครื่องคอมพิวเตอร์ ดังรูปที่ 2.6



รูปที่ 2.6 โครงสร้างและการทำงานของ เอชดีเอฟเอช[13]

เมื่อแฟ้มข้อมูลได้ถูกนำเข้าระบบและพร้อมใช้งานสำหรับวิธีการค้นหาของแมปรีดิวซ์บนฮาดูปแล้วจะเป็นขั้นตอนการค้นหาโดยใช้แมปรีดิวซ์ตามรูปที่ 2.7 โดยวิธีการทำงานของแมป คือเมื่อเดต้าโหนดได้รับคำร้องขอเพื่อให้ค้นหาข้อมูลจากแต่ละเครื่อง ซึ่งทำการอ่านข้อมูลทีละบรรทัด แล้วนำข้อมูลที่อ่านขึ้นมาเพื่อทำการเปรียบเทียบผลลัพธ์ว่าตรงตามเงื่อนไขที่ผู้ใช้ป้อนเข้ามาหรือไม่ ถ้าตรงโปรแกรมก็จะส่งผลลัพธ์กลับไปรวมกันยังเครื่องของผู้ใช้ แต่เนื่องจากการค้นหาข้อมูลในระบบไม่ได้มีการนำข้อมูลไปประมวลผลต่อ ดังนั้น ผลลัพธ์ที่ได้จากการค้นหาข้อมูลจะเป็นการนำเฉพาะข้อมูลที่ตรงกับเงื่อนไขของผู้ใช้ป้อนเข้ามาแล้วนำไปแสดงผลเท่านั้น ทำให้ไม่มีการใช้รีดิวซ์



รูปที่ 2.7 หลักการทำงานของวิธี แมปรีดิวซ์[13]

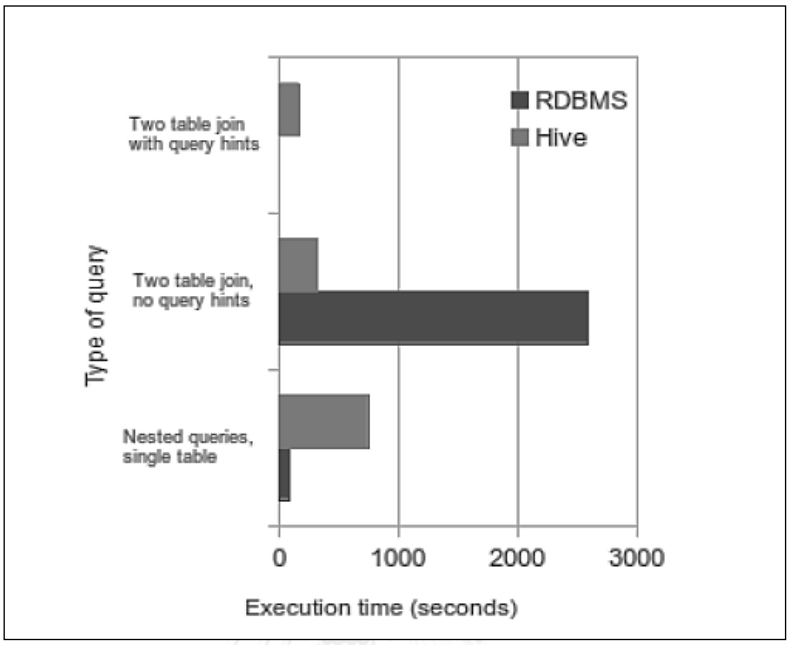
ผลการทดลองความเร็วในการนำแฟ้มข้อมูลเข้าสู่ระบบ เอชดีเอฟเอช ความเร็วของการนำข้อมูลขนาด 5 กิกะไบต์ เข้าสู่ระบบอยู่ที่ 167 เมกกะไบต์ต่อวินาที ในขณะที่ข้อมูลขนาด 50 กิกะไบต์ เข้าสู่ระบบอยู่ที่ 98 เมกกะไบต์ต่อวินาที ด้วยความเร็วที่ตกลงเนื่องจากจำนวนของบล๊อคที่เนมโหนดต้องติดต่อให้เดต้าโหนดจัดสรรให้นั้นมีจำนวนเพิ่มขึ้น และผลการทดสอบความเร็วในการค้นหาข้อมูล พบว่าในการค้นหาข้อมูลเร็วขึ้นเป็นสัดส่วนตามจำนวนเครื่องเดต้าโหนดซึ่งจากการค้นหาข้อมูลขนาด 5 กิกะไบต์ ความเร็วที่ได้จะเพิ่มขึ้นตามจำนวนเท่าของเครื่องเดต้าโหนดที่เพิ่มขึ้น แต่เมื่อข้อมูลที่ต้องการค้นหามีขนาดไม่ใหญ่พอ ความเร็วในการค้นหาจะเริ่มถึงจุดอิ่มตัวที่การเพิ่มจำนวนเครื่อง สรุปได้ง่ายๆว่า การเพิ่มจำนวนเครื่องเดต้าโหนดเข้าไม่ช่วยทำให้การค้นหาเร็วขึ้นเมื่อเทียบกับจำนวนข้อมูลที่ต้องการค้นหา

2.5.2. งานวิจัย : The Impact of Cluster Characterisitics on HiveQL Query Optimization. โดย Joldzic, O. V., & Vukovic, D. R. ปี 2013 [14]

งานวิจัยนี้กล่าวไว้ว่า เมื่อมีข้อมูลจำนวนใหญ่มากโดยใช้การจัดเก็บแบบเดิม ๆ มาจัดการข้อมูลนั้นไม่สมควรอย่างยิ่งเพราะไม่ได้ช่วยแก้ปัญหาการจัดการกับข้อมูลขนาดใหญ่ เพื่อแก้ปัญหาเรื่องนี้จึงนำจัดการข้อมูลแบบไม่มีโครงสร้างเข้ามาช่วย เพื่อที่จะถ่ายโอนข้อมูลจาก ฐานข้อมูลเชิงสัมพันธ์ไปยังฐานข้อมูลที่ไม่มีความสัมพันธ์ โดยนำเสนอเครื่องมือที่มีชื่อว่าสคู๊ป (Sqoop) ซึ่งจะช่วยในการเพิ่มประสิทธิภาพการดึงข้อมูลจากฐานข้อมูลโดยไม่สนใจว่าข้อมูลดังกล่าวจะอยู่ที่ตำแหน่งใด งานวิจัยนี้จึงนำเสนอลักษณะของคลัสเตอร์ที่ส่งผลกระทบต่อการเพิ่มประสิทธิภาพของไฮท์คิวแอล (HiveQL)

คุณสมบัติสำคัญของเอชดีเอฟเอชเบส (HDFS-based) คือความยืดหยุ่น และประสิทธิภาพ โดยต้องคำนึงถึงปัจจัยหลายอย่าง เช่น การกำหนดค่าพารามิเตอร์ที่ส่งผลต่อแอพพลิเคชัน (Application) ที่เรียกใช้คลัสเตอร์ในสภาพแวดล้อมเดียวกัน หรือการปรับปรุงประสิทธิภาพของระบบโดยการ

ปรับปรุงชุดคำสั่งในการสอบถามข้อมูลสำหรับการเรียกใช้ฐานข้อมูล แต่ไม่ได้มีจุดมุ่งหมายในการลดประสิทธิภาพในการทำงานของคลังข้อมูลเพียงแต่ลดการใช้งานบางอย่างของระบบ



รูปที่ 2.8 ชาร์ตเปรียบเทียบเวลาในการเข้าถึงข้อมูลสำหรับชุดคำสั่งของอาร์ดีบีเอ็มเอชและไฮท์[14]

จากรูปที่ 2.8 จะแสดงถึงตัวอย่างการเปรียบเทียบระหว่าง 2 ชุดคำสั่งการสอบถาม (Query) บน ไฮท์คลัสเตอร์ (Hive Cluster) และ อาร์ดีบีเอ็มเอช (RDBMS) โดยที่จะเลือกเป็นแบบการค้นหาทุกตัว (Full-text search)

ชุดสอบถามข้อมูลที่ 1 (Nested queries single table) เป็นการสอบถามข้อมูล 1 ตารางโดยไม่มีการทำดัชนี มีจำนวนแถวในตาราง (record) 108 แถว และ ประกอบด้วย 2 ชุดการสอบถามข้อมูลย่อย (Sub Query) สรุปผลดังกล่าวได้ว่าไฮท์ มีความต้องการใช้ทรัพยากรด้านหน่วยความจำทั้งขาเข้าและขาออกสูงดังนั้น จึงใช้เวลามากกว่าอาร์ดีบีเอ็มเอช

ชุดสอบถามข้อมูลที่ 2 (2 table join, no query hint) ประกอบไปด้วยอินเนอร์จอย (inner join) ระหว่างตาราง เดียวกันสำหรับชุดสอบถามข้อมูลในชุดแรกแรก และตารางที่ 2 มีขนาดเล็กมาก เนื่องจากไม่มีการทำดัชนีข้อมูล และในกรณีนี้เองไฮท์ ก็สามารถจัดการผ่านคลัสเตอร์หรือโหนดข้อมูลได้เป็นอย่างดีซึ่งให้ผลดีกว่าการสอบถามข้อมูลแบบแรก เนื่องจากมีการใช้งานแมปจอย (MapJoin)

ชุดสอบถามข้อมูลที่ 3 (2 table join, with query hint) ในกรณีนี้แตกต่างจากการสอบถามข้อมูลที่ 2 คือ มีตัวช่วยในการสอบถามข้อมูล (hint query) แต่อย่างไรก็ตามผลที่ได้ออกมาคือ ไฮท์ยังสามารถประมวลผลได้ แต่ว่าอาร์ดีบีเอ็มเอชเกิดปัญหาคอขวดของข้อมูลและผลลัพธ์ ทำให้ไม่สนับสนุนการเกิดปัญหาดังกล่าว จึงไม่มีผลแสดงในแผนภาพดังกล่าว

ดังนั้นประสิทธิภาพการทำงานของการสอบถามข้อมูลของไฮท์ จะพบว่าการปรับปรุงการสอบถามข้อมูลเป็นปัจจัยหนึ่งในการใช้ทรัพยากรของไฮท์ด้วย นอกจากการปรับปรุงการสอบถามข้อมูลข้างต้น อีกปัจจัยหนึ่งที่สามารถเพิ่มประสิทธิภาพได้ คือ การเรียงลำดับขนาดของตารางและใช้ตัวช่วยในการสอบถามข้อมูลสำหรับตารางที่มีขนาดใหญ่

2.5.3. งานวิจัย : Storage and Retrieval of Large RDF Graph Using Hadoop and MapReduce โดย Husain, M. F., Doshi, P., & Khan, L. ปี 2009 [2]

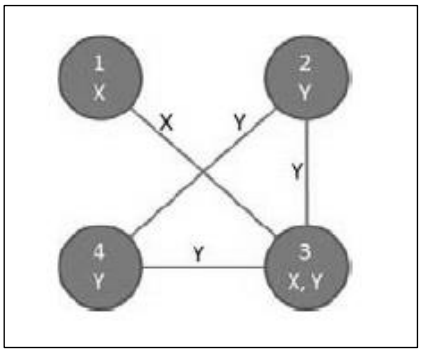
งานวิจัยนี้นำเสนอการจัดเก็บและการค้นหาอาร์ดีเอฟกราฟที่มีขนาดใหญ่โดยใช้ฮาดูป เพื่อแสดงให้เห็นถึงประสิทธิภาพการเข้าถึงข้อมูลด้านความเร็วที่เพิ่มขึ้น โดยแบ่งงานในส่วนของการเตรียมข้อมูลออกเป็น 3 ขั้นตอนหลักๆ คือ

1) การสร้าง อาร์ดีเอฟ เอ็กซ์เอ็มแอล ของข้อมูลที่ใช้ในงานวิจัย โดยข้อมูลที่ใช้คือ The Lehigh University Benchmark (LUBM)
2) แปลงข้อมูลให้อยู่ในรูปแบบของเอ็นทริปเปิ้ลโดยใช้กรอบการทำงานของจีน่า (Jena Framework) ในการแปลงข้อมูล
3) ทำการแบ่งข้อมูลโดยแยกตามประเภท คือ กริยา (Predicate Type) และ กรรม (Object Type) แล้วเขียนไฟล์เพื่อเตรียมนำไฟล์ผลลัพธ์ที่ได้เข้าสู่ เอชดีเอฟเอชงานในส่วนของการการค้นหาข้อมูลจะทำการรับ สปาร์เคิล เพื่อทำการค้นหาข้อมูลใน ฮาดูป

```
SELECT ?X WHERE {                                                    1
?X rdf:type ub:Chair .                                               2
?Y rdf:type ub:Department .                                          3
?X ub:worksFor ?Y .                                                  4
?Y ub:subOrganizationOf <http://www.University0.edu> }               5
```

รูปที่ 2.9 ตัวอย่างการสอบถามข้อมูลจาก LUBM[2]

จากรูปที่ 2.9 สามารถแสดงกราฟสำหรับการสอบถามข้อมูลได้ดังรูปที่ 2.10



รูปที่ 2.10 กราฟสำหรับชุดการสอบถามข้อมูล LUBM[2]

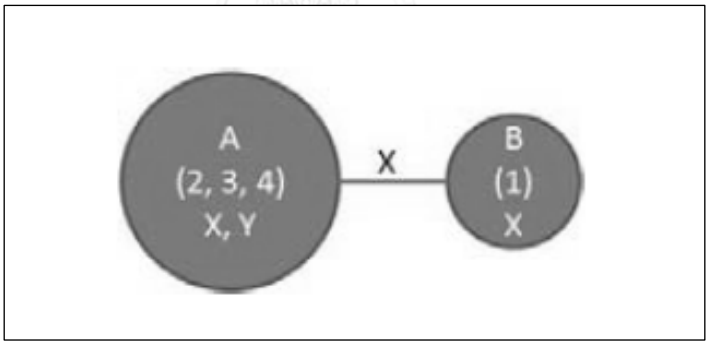
Algorithm 1. DETERMINEJOBS(*Query q*)

Require: A Query object returned by RewriteQuery algorithm.
Ensure: The number of jobs and their details needed to answer the query.

```
1:  jobs ← φ; graph ← makeGraphFromQuery(q); joins_left ← calculateJoins(graph)
2:  while joins_left ≠ 0 do
3:      variables ← getVariables(graph); job ← createNewJob()
4:      for i ← 1 to |variables| do
5:          v ← variables[i]; v.nodes ← getMaximumVisitableNodes(v, graph)
6:          v.joins ← getJoins(v.nodes, graph)
7:      end for
8:      sortVariablesByNumberOfJoins(variables)
9:      for i ← 0 to |variables| do
10:         if |v.joins| ≠ 0 then
11:             job.addVariable(v); jobs_left ← jobs_left − |v.joins|
12:             for j ← i + 1 to |variables| do
13:                 adjustNodesAndJoins(variables[j], v.nodes)
14:             end for
15:             mergeNodes(graph, v.nodes)
16:         end if
17:     end for
18:     jobs ← jobs ∪ job
19: end while
20: return jobs
```

รูปที่ 2.11 หลักการดีเทอร์ไมท์จ๊อป (DetermineJobs Algorithm)[2]

และเมื่อสปาร์เคิล ผ่านกระบวนการทำงานของโปรแกรม ดังรูปที่ 2.11 ทำให้กราฟของการสอบถามข้อมูลที่ได้แสดงดังรูปที่ 2.12



รูปที่ 2.12 กราฟสำหรับหลักการดีเทอร์ไมท์จ็อป[2]

ผลการทดลองจะแสดงให้เห็นเพียง 6 ชุดคำสั่งการสอบถามข้อมูล ซึ่งสามารถสรุปได้ว่าจำนวนของ ทริปเปิ้ลเพิ่มขึ้นส่งผลให้เวลาในการสอบถามข้อมูลเพิ่มขึ้นด้วย

2.5.4. งานวิจัย: Visualization of Resource Description Framework Ontology Using Hadoop โดย Park, S. ปี 2013 [15]

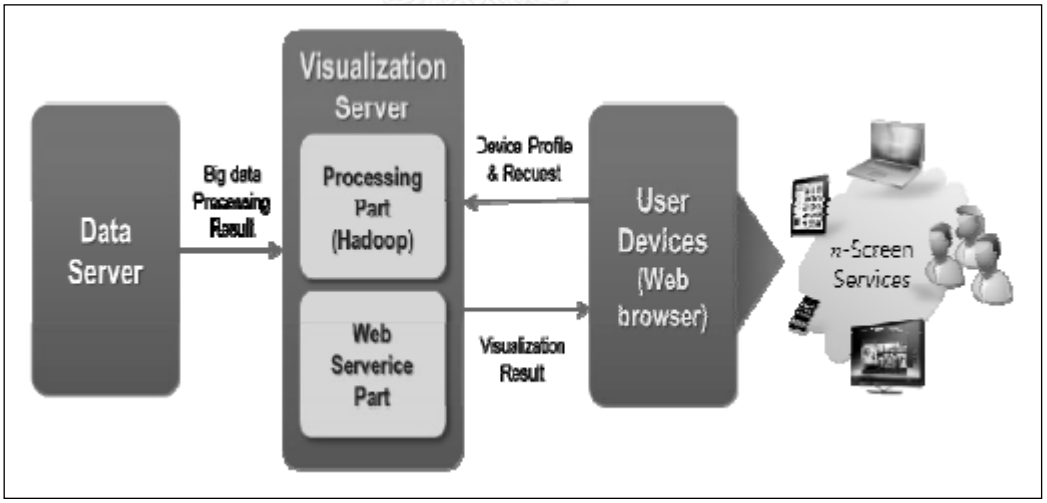
งานวิจัยนี้นำเสนอปัญหาจากการจัดการการแสดงข้อมูลที่มีขนาดใหญ่ด้วยอาร์ดีเอฟแล้วนำ ฮาดูป มาเปรียบเทียบเพื่อแก้ปัญหาดังกล่าว จากรูปที่ 2.13 ระบบจะประกอบด้วยเดต้าเซิร์ฟเวอร์ (Data Server), เวอร์ชัวไลท์เซชันเซิร์ฟเวอร์ (Visualization Server) และ ยูเซอร์ไดร์ฟเวอร์ (User Device) โดยมีหลักการทำงานดังนี้

1) เดต้าเซิร์ฟเวอร์จะทำการประมวลผลข้อมูลก่อน
2) เวอร์ชัวไลท์เซชันเซิร์ฟเวอร์จะประมวลผลต่อจากเดต้าเซิร์ฟเวอร์
3) ถ่ายโอนข้อมูลไปยังเว็บโปรเซสชิง (Web processing)
4) เดต้าเซิร์ฟเวอร์และเวอร์ชัวไลท์เซชันเซิร์ฟเวอร์เรียกใช้ฮาดูป
5) ผู้ใช้งานสามารถกำหนดผลของการแสดงข้อมูลของเวอร์ชัวไลท์เซชันเซิร์ฟเวอร์ ผ่านเว็บเบราเซอร์ (Web browser)

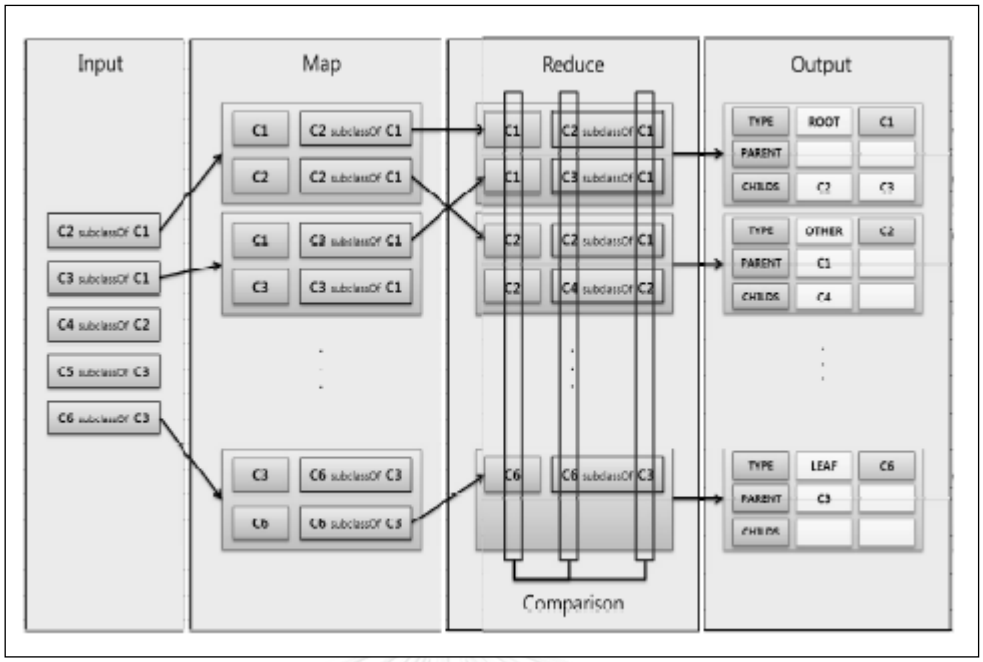
ปัญหาที่เกิดจากเมื่อข้อมูลที่มากขึ้น ทำให้เวอร์ชัวไลท์เซชันเซิร์ฟเวอร์มีความเร็วลดลง ดังนั้นจึงได้นำอาร์ดีเอฟเวอร์ชัวไลท์เซชันเซิร์ฟเวอร์ (RDFS Visualization) เข้ามาช่วยในการประมวลผล โดยการนำข้อมูลที่มีอยู่ในเดต้าเซิร์ฟเวอร์แล้วนำมาทำการแมปค่าของคีย์และค่าของข้อมูล ให้ได้เป็น Map<Key, Value> ใหม่ เพื่อลดความซ้ำซ้อนของ <Key, Value> จากรูปที่ 2.14 เมื่อใช้กระบวนการประมวลผลได้ ผลลัพธ์สุดท้ายของข้อมูลสามารถสรุปได้เป็นรูปแบบของข้อมูลได้เป็น

[ClassName][Type][Parent Class][Child Classes]

ผลการทดสอบระบบด้วยชุดของคลาส (Class) ที่เป็นลำดับชั้นของคลาส (Hierarchies) สามารถสรุปได้ว่าฮาดูป สามารถประมวลผลงานที่มีจำนวนมากและมีขนาดใหญ่มีประสิทธิภาพดีกว่าการประมวลผลแบบปกติ



รูปที่ 2.13 ภาพรวมของระบบ[15]



รูปที่ 2.14 ภาพรวมการประมวลผลของอาร์ดีเอฟเอชในฮาดูป[15]

2.5.5. งานวิจัย: Executing SPARQL Queries over the Web of Linked Data. โดย Olaf Hartig , Christian Bizer, Johann-Christoph Freytag. ปี 2009. [16]

การศึกษางานวิจัยที่ใช้สปาคิวแอล เพื่อการเข้าถึงข้อมูลของซีแมนติกเว็บสำหรับลิงค์เดต้า (Semantic web of Linked Data) โดยใช้ข้อมูลของห้องสมุดในการทดลอง แนวคิดสำคัญของงานวิจัย การสอบถามข้อมูลแบบ traverse สำหรับอาร์ดีเอฟซึ่งบางครั้งสามารถที่จะเข้าถึงโหนด ของเริ่มต้นของตัวมันเองได้ด้วย ดังนั้นพวกเขาจึงคิดว่าการนำไปใช้กับอิทเทอร์เรเตอร์เบสไปป์ไลน์ (iterator-based pipeline) และยูอาร์ไอ (URI) ที่จะสามารถเพิ่มประสิทธิภาพของการสอบถามข้อมูลได้เป็นอย่างดี ซึ่งพวกเขาหวังว่าข้อมูลที่นำมาใช้ มีจำนวนของลิงค์ค่อนข้างสูงนั้นทำให้ผลลัพธ์มีความสมบูรณ์มากขึ้น เหตุผลที่นอกเหนือจากการใช้ข้อมูลดังกล่าว สามารถการันตีได้ว่าการเริ่มต้นของสอบถามข้อมูลจะไม่เริ่มจากการค้นหาโหนดหรือชุดของข้อมูลที่ว่างเปล่าแน่นอน แต่อย่างไรก็ตามการสอบถามข้อมูลผ่านซีแมนติกเว็บด้วยหลักการเอชทีทีพีทำให้เกิดอาการคอขวด (bottleneck) เนื่องจากการล่าช้าของการตอบสนองของโปรโตคอล (Protocol) ซึ่งทำให้เกิดการล้มเหลวของการสอบถามข้อมูลสูง

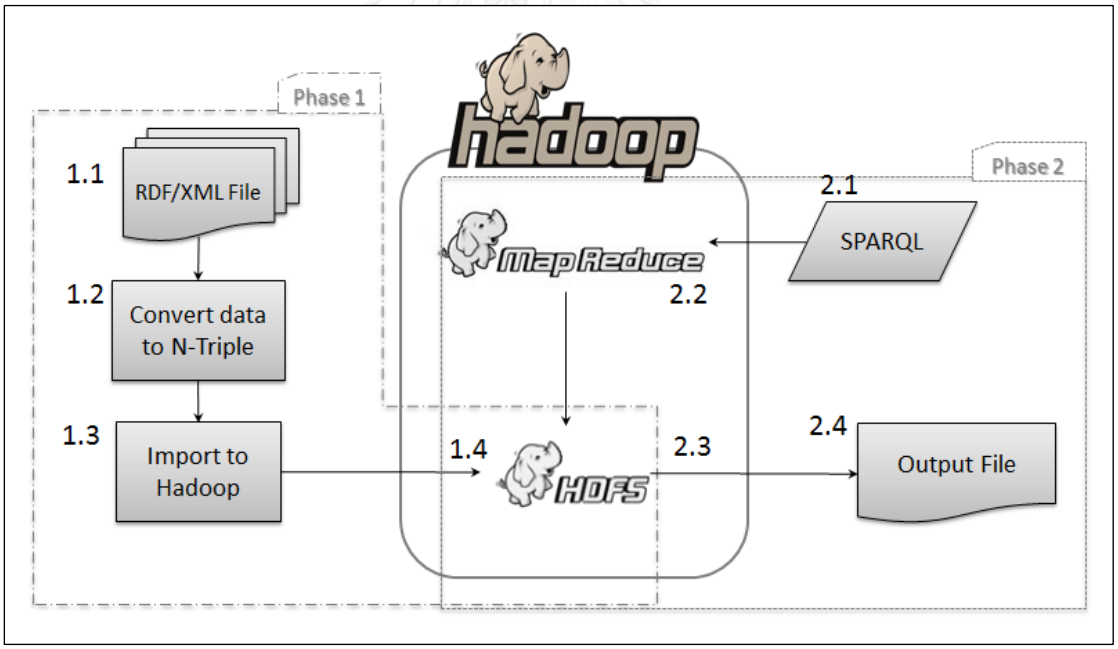
บทที่ 3

การออกแบบและการพัฒนาการถ่ายโอนและสอบถามข้อมูล

ในบทนี้จะครอบคลุมเนื้อหากระบวนการ หลักการพัฒนาการถ่ายโอนและสอบถามข้อมูลที่มีขนาดใหญ่จากลิงค์เดต้าในรูปแบบของอาร์ดีเอฟเอ็กซ์เอ็มแอล เข้าสู่กรอบการทำงานของฮาดูป และวัดประสิทธิภาพของการสอบถามข้อมูลที่มีอยู่ในระบบ

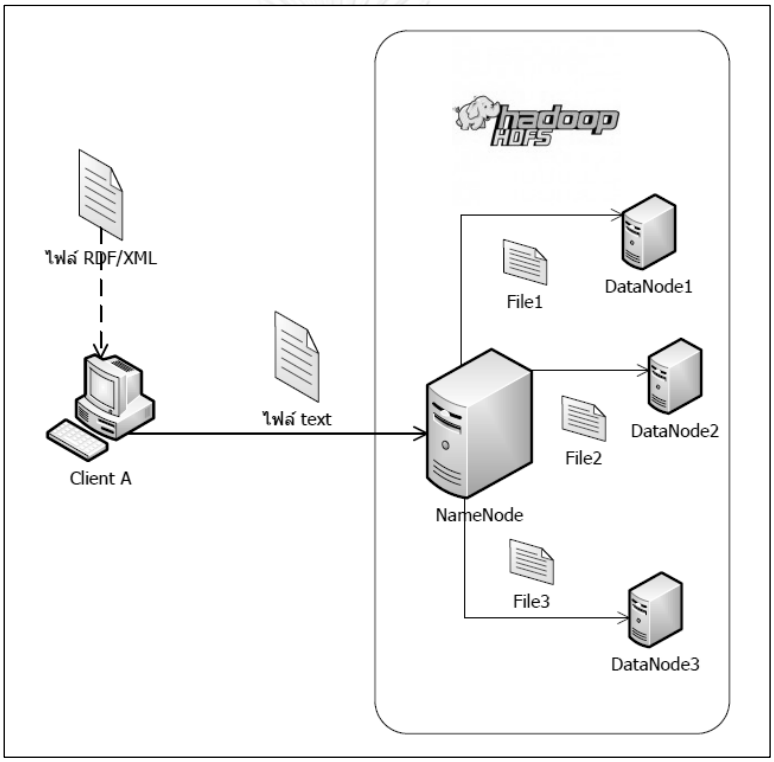
3.1 โครงสร้างการถ่ายโอนและสอบถามข้อมูล

จากแนวคิดสำหรับกระบวนการทำงานของการถ่ายโอนและสอบถามข้อมูล สามารถสรุปภาพรวมของการทำงานการถ่ายโอนข้อมูลจากลิงค์เดต้าในรูปแบบของอาร์ดีเอฟเอ็กซ์เอ็มแอลไปยังฮาดูปตามรูปที่ 3.1



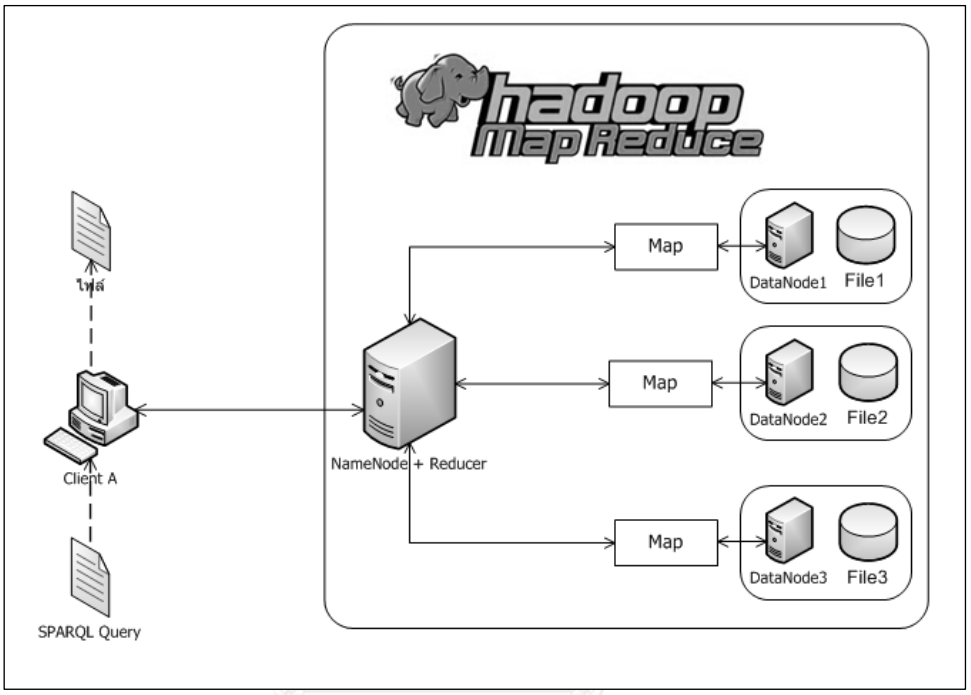
รูปที่ 3.1 ภาพรวมการทำงานของการถ่ายโอนข้อมูลและสอบถามข้อมูล

จากรูปที่ 3.1 สามารถอธิบายรายละเอียดภาพรวมของการถ่ายโอนและสอบถามข้อมูลเป็น 2 ส่วน ซึ่งในส่วนแรกเป็นการถ่ายโอนข้อมูล โดยการนำข้อมูลอาร์ดีเอฟในรูปแบบเอ็กซ์เอ็มแอล จาก The Lehigh University Benchmark (LUBM). นำข้อมูลที่ได้ไปตรวจสอบความถูกต้องของโครงสร้างตามรูปแบบของเอ็กซ์เอ็มแอล (Well-formed XML) เมื่อผ่านการตรวจสอบแล้ว นำข้อมูลดังกล่าวแปลงให้อยู่ในรูปของเอ็นทริปเปิ้ล ที่ประกอบด้วย ประธาน – คุณสมบัติ – ค่าคุณสมบัติ ตามโครงสร้างของเอ็นทริปเปิ้ล ด้วยจีน่าโมเดล เมื่อได้ข้อมูลที่อยู่ในรูปของเอ็นทริปเปิ้ลเรียบร้อยแล้ว ขั้นตอนต่อไปเป็นการนำข้อมูลเข้าสู่เอชดีเอฟเอสในฮาดูป โดยใช้คำสั่งของฮาดูปเฟรมเวิร์คที่มีให้ เมื่อนำข้อมูลเข้าสู่ฮาดูป ในส่วนของเอชดีเอฟเอสจะทำการแตกไฟล์ข้อมูลเป็นส่วนย่อยๆ เพื่อส่งไปเก็บยังคลัสเตอร์ที่มีอยู่ในระบบ เมื่อทำการจัดเก็บเรียบร้อยแล้วเป็นอันเสร็จสิ้นงานในส่วนที่ 1 สามารถอธิบายได้ดังรูปที่ 3.2



รูปที่ 3.2 โครงสร้างและการทำงานของส่วนที่ 1 นำไฟล์ข้อมูลอาร์ดีเอฟเอ็กซ์เอ็มแอลเข้าสู่ฮาดูป

ในส่วนที่ 2 หลังจากที่ทำการจัดเก็บข้อมูลในคลัสเตอร์เรียบร้อย ในส่วนนี้จะเป็นการสอบถามข้อมูลที่มีอยู่ในระบบ โดยการอาศัยหลักการของแมปรีดิวซ์ เพื่อรับสปาร์เคิลซึ่งเป็นภาษาในการสอบถามข้อมูลสำหรับเอ็นทริปเปิ้ลและแสดงค่าผลลัพธ์ที่ได้จากการสอบถามเป็นไฟล์ สามารถอธิบายตามรูปที่ 3.3

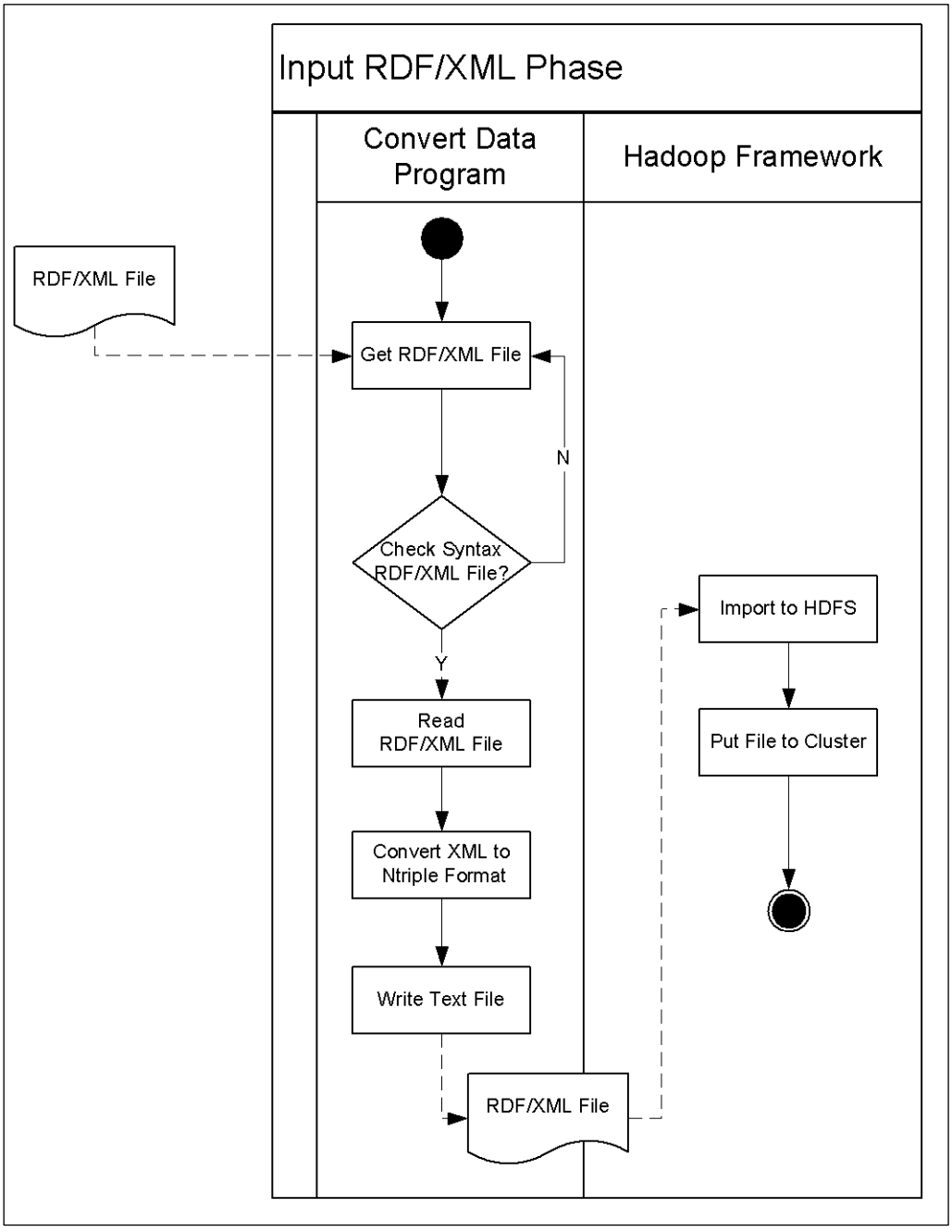


รูปที่ 3.3 โครงสร้างและการทำงานของส่วนที่ 2 สอบถามข้อมูลจากฮาดูปโดยใช้สปาร์เคิล

3.2 หลักการทำงานของการถ่ายโอนและสอบถามข้อมูล

3.2.1 ส่วนที่ 1 การแปลงข้อมูลและนำข้อมูลเข้าสู่ฮาดูป

ในส่วนของหลักการทำงานในส่วนที่ 1 สามารถอธิบายเป็นลำดับการทำงานตามรูปที่ 3.4



รูปที่ 3.4 ลำดับการทำงานของการถ่ายโอนและสอบถามข้อมูลในส่วนที่ 1 การแปลงข้อมูลและนำข้อมูลเข้าสู่ฮาดูป

จากรูปที่ 3.4 เมื่อนำชุดข้อมูลสำหรับทำการทดลองที่มีชื่อว่า The Lehigh University Benchmark (LUBM). ในรูปแบบของอาร์ดีเอฟเอ็กซ์เอ็มแอล โดยจากแหล่งชุดข้อมูลดังกล่าว มีการแตกไฟล์เป็นส่วนๆ ที่ครบถ้วนถูกต้องตามรูปแบบเอ็กซ์เอ็มแอลที่สมบูรณ์ ดังตัวอย่างข้อมูลอาร์ดีเอฟเอ็กซ์เอ็มแอล ตามรูปที่ 3.5 ผ่านการตรวจสอบความถูกต้องข้อมูลพื้นฐานในรูปแบบเอ็กซ์เอ็มแอลทั่วไป

```
<?xml version="1.0" encoding="UTF-8" ?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:ub="http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#">

<owl:Ontology rdf:about="">
<owl:imports rdf:resource="http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl" />
</owl:Ontology>

<ub:University rdf:about="http://www.University0.edu">
  <ub:name>University0</ub:name>
</ub:University>

<ub:Department rdf:about="http://www.Department0.University0.edu">
  <ub:name>Department0</ub:name>
  <ub:subOrganizationOf>
        <ub:University rdf:about="http://www.University0.edu" />
  </ub:subOrganizationOf>
</ub:Department>
</rdf:RDF>
```

รูปที่ 3.5 ตัวอย่างอาร์ดีเอฟเอ็กซ์เอ็มแอลจาก The Lehigh University Benchmark (LUBM).

จากนั้นนำข้อมูลเข้าสู่กระบวนการตรวจสอบความถูกต้องของข้อมูลตามหลักการของรูปแบบข้อมูลแบบเอ็กซ์เอ็มแอล เมื่อรูปแบบข้อมูลถูกต้อง กระบวนการต่อมาคือการแปลงข้อมูลในรูปแบบของอาร์ดีเอฟเอ็กซ์เอ็มแอลให้อยู่ในรูปของเอ็นทริปเปิ้ลโดยใช้จีน่าอัลจีบรา ซึ่งเป็นเครื่องมือที่ช่วยในการแปลงข้อมูลอาร์ดีเอฟเอ็กซ์เอ็มแอล ให้อยู่ในรูปของเอ็นทริปเปิ้ล ซึ่งมีรูปแบบของข้อมูลดังนี้

ประธาน – คุณสมบัติ – ค่าคุณสมบัติ (Subject – Predicate – Object)

โดยข้อมูลที่ผ่านการแปลงข้อมูลแล้วจะยังคงความสัมพันธ์เดิมของอาร์ดีเอฟเอ็กซ์เอ็มแอลไว้ ตัวอย่างข้อมูลหลังจากการแปลงให้อยู่ในรูปของเอ็นทริปเปิ้ล ตามรูปที่ 3.6

```
<http://www.Department0.University0.edu> <http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#subOrganizationOf> <http://www.University0.edu> .
<http://www.Department0.University0.edu> <http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#name> "Department0" .
<http://www.Department0.University0.edu> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#Department>
<http://www.University0.edu> <http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#name> "University0" .
<http://www.University0.edu> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#University> .
```

รูปที่ 3.6 ตัวอย่างข้อมูลหลังจากผ่านการแปลงให้อยู่ในรูปแบบเอ็นทริปเปิ้ล

เมื่อได้ข้อมูลที่ผ่านการแปลงเรียบร้อยแล้ว ขั้นตอนการนำเข้าข้อมูลเข้าสู่ฮาดูปคลัสเตอร์ โดยใช้ชุดคำสั่งของเอชดีเอฟเอช เพื่อนำเข้าข้อมูลและให้เอชดีเอฟเอชประมวลผล จัดสรรหน่วยความจำในการจัดเก็บข้อมูล ด้วยคำสั่งของเอชดีเอฟเอชตามรูปที่ 3.7 และตรวจสอบขนาดไฟล์ก่อนนำเข้าสู่ฮาดูปคลัสเตอร์ แสดงตามรูปที่ 3.8

```
[hadooper@sandbox ~]$ hadoop fs -put NTriple_1.75GB_20160326.nt /user/input/NTriple.nt
```

รูปที่ 3.7 คำสั่งการนำไฟล์ข้อมูลในรูปแบบเอ็นทริปเปิ้ลเข้าสู่ฮาดูปคลัสเตอร์

```
[hadooper@sandbox data]$ pwd
/home/hadooper/rdfdata/data
[hadooper@sandbox data]$ ls -l
total 7610576
-rw-rw-r-- 1 hadooper hadooper 4043723650 2016-03-26 10:38 NTriple.nt
-rw-rw-r-- 1 hadooper hadooper 1874741629 2016-07-18 09:39 sum_owl
-rw-rw-r-- 1 hadooper hadooper  857346586 2016-07-20 19:21 sum_owl1
-rw-rw-r-- 1 hadooper hadooper 1017391801 2016-07-20 20:24 sum_owl2
```

รูปที่ 3.8 แสดงขนาดของไฟล์ก่อนนำเข้าสู่เอชดีเอฟเอส

ตรวจสอบไฟล์ที่นำเข้าสู่ฮาดูปคลัสเตอร์และแสดงผลชื่อไฟล์ที่นำเข้าสู่ฮาดูปตาม รูปที่ 3.9

```
[hadooper@sandbox ~]$ hadoop fs -ls /user/input
Found 1 items
-rw-r--r--   3 hadooper hdfs 4043723650 2016-03-29 14:52 /user/input/NTriple.nt
[hadooper@sandbox ~]$
```

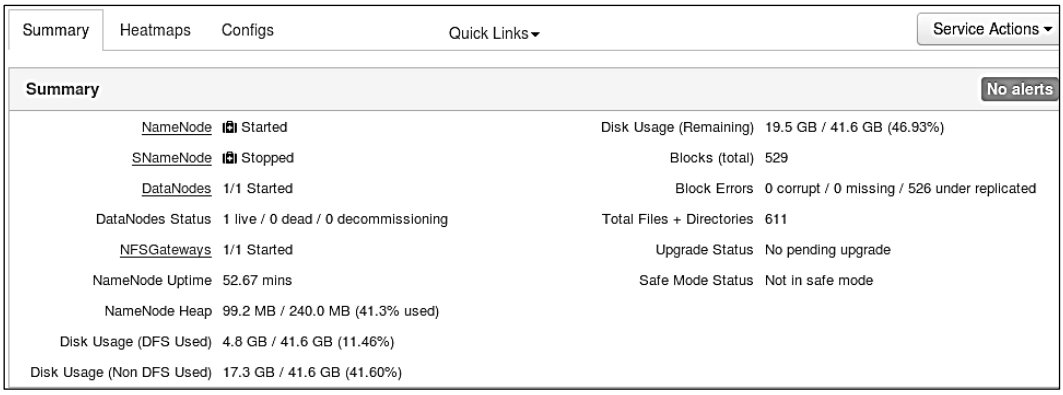
รูปที่ 3.9 คำสั่งในการตรวจสอบไฟล์ที่นำเข้าฮาดูป

และตรวจสอบขนาดเมื่อนำไฟล์เข้าสู่ฮาดูปคลัสเตอร์เรียบร้อยจะได้ไฟล์ข้อมูลที่มีขนาดเท่าเดิมตามรูปที่ 3.10

```
[hadooper@sandbox data]$ hadoop fs -ls /user/input/NTriple.nt
-rw-r--r--   1 hadooper hdfs 4043723650 2016-07-19 21:28 /user/input/NTriple.nt
[hadooper@sandbox data]$
```

รูปที่ 3.10 แสดงขนาดของไฟล์หลังนำเข้าสู่เอชดีเอฟเอส

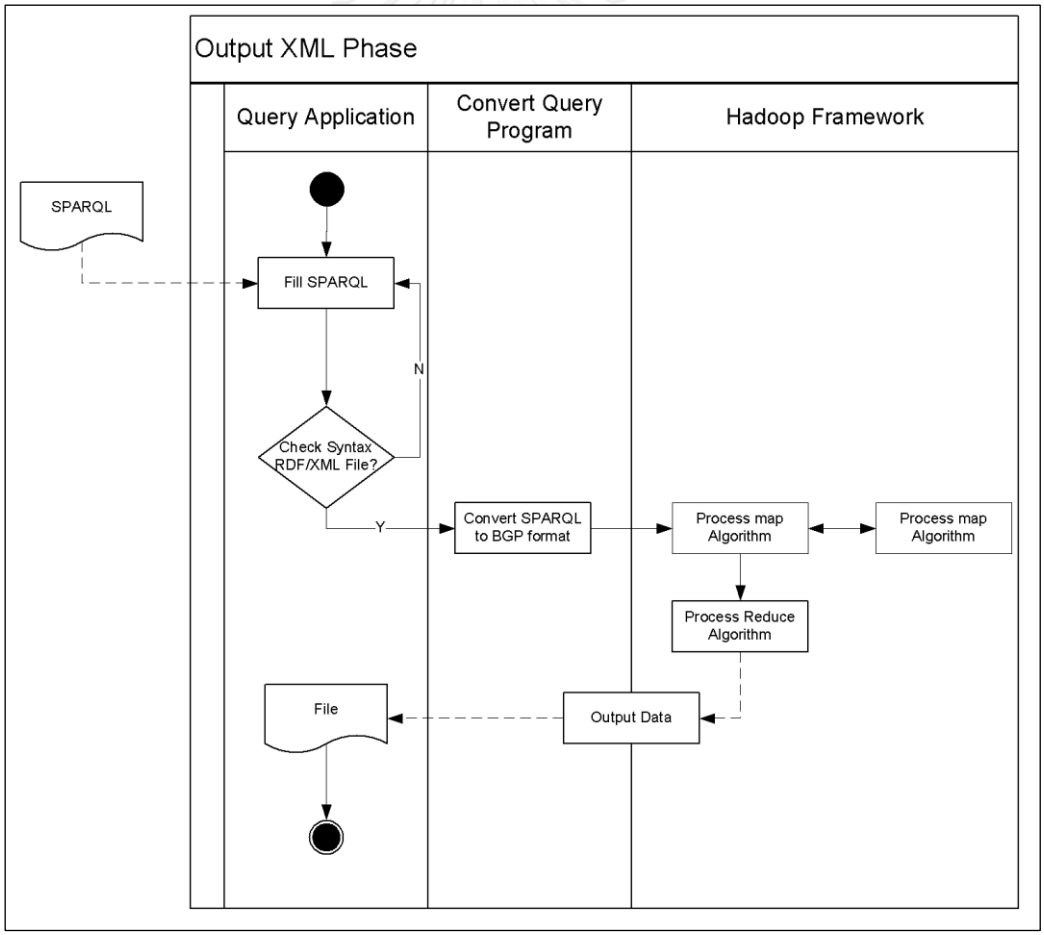
เมื่อทำการนำข้อมูลเข้าสู่ฮาดูปคลัสเตอร์เรียบร้อยแล้ว ข้อมูลทั่วไปของฮาดูปคลัสเตอร์จะแสดงรายละเอียดของข้อมูลที่ถูกจัดสรรโดยเอชดีเอฟเอช ตามรูปที่ 3.11



รูปที่ 3.11 รายละเอียดข้อมูลของข้อมูลในฮาดูปคลัสเตอร์

3.2.2 ส่วนที่ 2 การสอบถามข้อมูลโดยใช้สปาร์เคิล

จากรูปที่ 3.12 แสดงโครงสร้างการทำงานส่วนที่ 2 ของการสอบถามข้อมูลในฮาดูปด้วยหลักการของแมปรีดิวซ์โดยใช้สปาร์เคิล



รูปที่ 3.12 ลำดับการทำงานของการถ่ายโอนและสอบถามข้อมูลในส่วนที่ 2 การสอบถามข้อมูลโดยใช้สปาร์เคิล

ก่อนการส่งคำสั่งเข้าไปสอบถามข้อมูลในฮาดูปโดยใช้หลักการของแมปรีดิวซ์นั้น จะทำการแปลงชุดคำสั่ง ตามรูปที่ 3.13 โดยทำการรับสปาร์เคิลด้วยการอ่านชุดคำสั่งจากไฟล์ ตัวอย่างชุดคำสั่งสปาร์เคิล เมื่อรับชุดคำสั่งเข้าสู่ระบบแล้ว จะทำการตรวจสอบความถูกต้องของรูปแบบข้อมูลว่ามีรูปแบบที่ถูกต้องหรือไม่

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX ub: <http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#>
3 SELECT ?X
4 WHERE
5 {?X rdf:type ub:GraduateStudent .
6  ?X ub:takesCourse "http://www.Department0.University0.edu/GraduateCourse0"
7 }
```

รูปที่ 3.13 ตัวอย่างชุดคำสั่งสปาร์เคิล

ในกรณีที่ชุดคำสั่งมีรูปแบบที่ถูกต้อง จะทำการแปลงชุดคำสั่งสปาร์เคิลให้อยู่ในรูปของเอ็นทริปเปิ้ลที่เรียกว่า เบสิคกราฟแพทเทิร์น (BGP – Basic Graph Pattern) ซึ่งแบ่งโครงสร้างเป็น 3 ส่วน ตามรูปแบบของเอ็นทริปเปิ้ล คือ ประธาน – คุณสมบัติ – ค่าคุณสมบัติ ตามรูปที่ 3.14 เพื่อเตรียมชุดคำสั่งส่งเข้าไปสอบถามข้อมูลในฮาดูป

```
1 (project (?X)
2   (BGP
3     (triple ?X <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#GraduateStudent>)
4     (triple ?X <http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#takesCourse> "http://www.Department0.University0.edu/GraduateCourse0")
5   )
6 )
```

รูปที่ 3.14 ตัวอย่างชุดคำสั่งการสอบถามข้อมูลในรูปแบบบีจีพี

หลังจากแปลงชุดคำสั่งเรียบร้อยที่ถูกแปลงด้วยจาวาแล้ว จะส่งคำสั่งที่แปลงเรียบร้อยแล้วเข้าไปยัง ไดเร็กทอรี่ของฮาดูปที่ถูกกำหนดไว้ จากนั้นจะเริ่มทำการรันโปรแกรมในส่วนของแมปรีดิวซ์เพื่อส่งคำสั่งในการสอบถามข้อมูลที่มีอยู่ในระบบ

หลักการทำงานของแมปรีดิวซ์

(1) หลักการของแมป

ระบบจะทำการอ่านค่าด้วยหลักการทีละแถวของข้อมูลทั้งหมดที่มีอยู่ในระบบนั่นคือ ข้อมูลขาเข้า ตามโดเมน domainMap (k1, v1) เพื่อทำการกำหนดและสร้างคีย์และค่าของข้อมูลตามโดเมนของแมป

$$\text{domainMap } (k1, v1) \longrightarrow \text{list } (K2, v2)$$

จากรูปที่ 3.16 ก่อนที่โปรแกรมจะทำการอ่านค่าข้อมูลจากเอชดีเอฟเอสนั้น โปรแกรมจะทำการอ่านสปาร์เคิล แล้วทำการแปลงสปาร์เคิลให้อยู่ในรูปของบีจีพี โดยใช้จีน่าโอพี ในการแปลงสปาร์เคิลให้อยู่ในรูปบีจีพี เพื่อทำการแยกแพทเทิร์นของชุดคำสั่ง ทำการแยกแพทเทิร์นของชุดคำสั่งการสอบถามที่ใช้สอบถามข้อมูลว่าตรงกับแพทเทิร์นใด ซึ่งจะแทนค่าตัวแปรที่ต้องการค้นหาด้วย ?ชื่อตัวแปร เช่น ?subject และสำหรับค่าที่นำมาเปรียบเทียบเพื่อหาคำตอบจะแทนด้วย ชื่อตำแหน่งข้อมูลแล้วกำกับด้วยคำว่า Data เช่น predicateData โดยจะแบ่งเป็น 3 แพทเทิร์นดังนี้

แพทเทิร์นที่ (1) ?subject predicateData objectData

แพทเทิร์นที่ (2) ?subject predicateData ?object

แพทเทิร์นที่ (3) subjectData predicateData ?object

แพทเทิร์นที่ (1) เป็นแพทเทิร์นที่ต้องการหาประธาน นั่นคือ ?subject โดยค่าที่ต้องการค้นหาคือค่าที่อยู่ในตำแหน่งของประธาน และค่าที่นำมาเปรียบเทียบนั่นคือค่าของ คุณสมบัติ ที่แทนด้วย predecateData และอีกหนึ่งค่าที่นำมาเปรียบเทียบคู่กับ predecateData คือ ค่าของคุณสมบัติ ที่ถูกแทนค่าด้วย objectData

แพทเทิร์นที่ (2) เป็นแพทเทิร์นที่ต้องการหาประธานและค่าของคุณสมบัติ นั่นคือ ?subject และ ?object โดยค่าที่ต้องการค้นหาคือค่าที่อยู่ในตำแหน่งของประธาน และค่าคุณสมบัติ ซึ่งค่าที่ต้องนำมาเปรียบเทียบนั่นคือ คุณสมบัติ ที่แทนด้วย predecateData

แพทเทิร์นที่ (3) เป็นแพทเทิร์นที่ต้องการหาค่าคุณสมบัติ นั่นคือ ?object โดยค่าที่ต้องการค้นหาคือค่าที่อยู่ในตำแหน่งของค่าคุณสมบัติ และค่าที่นำมาเปรียบเทียบนั่นคือค่าของ คุณสมบัติ ที่แทนด้วย predecateData และอีกหนึ่งค่าที่นำมาเปรียบเทียบคู่กับ predecateData คือ ประธานที่ถูกแทนค่าด้วย subjectData

ด้วยแพทเทิร์นที่ถูกกำหนดขึ้นมานี้เพื่อรองรับชุดคำสั่งในการสอบถามข้อมูลตัวอย่างที่นำมาใช้ในการทดสอบซึ่งในชุดคำสั่งในการสอบถามข้อมูลนั้นพบว่า คุณสมบัติ ในชุดคำสั่งในการสอบถามนั้นปรากฏเสมอในทุกชุดคำสั่ง ดังนั้นแพทเทร์นของชุดคำสั่งจึงถูกกำหนดขึ้นตามชุดคำสั่งที่มี

จากแพทเทิร์นที่ (1) – (3) แพทเทิร์นที่ถูกกำหนดขึ้นมานั้นจะถูกนำเข้ามาจับคู่กับชุดคำสั่งการสอบถามข้อมูลที่ได้จากการแยกชุดคำสั่งการสอบถามด้วยการอ่านค่าในแถวที่มีค่าทริปเปิ้ลแล้วแยกด้วย ค่าว่าง แล้วนำมาจับคู่กับรูปแบบที่ถูกกำหนดไว้ โดยกำหนดตัวแปรตัวแรกเป็นประธาน คุณสมบัติ และค่าคุณสมบัติ ตามลำดับ จากแพทเทิร์นชุดคำสั่งสามารถเขียนเป็นซูโดโค้ด (pseudo code) ตามรูปที่ 3.15 ตัวอย่างเช่น จากรูปที่ 3.14 สามารถแยกชุดคำสั่งการสอบถามข้อมูลได้เป็น 2 ชุด 1 แพทเทิร์นนั่นคือแพทเทิร์นที่ (1) เนื่องจากค่าตัวแปรที่ต้องการหาคือ ?x ซึ่งเป็นตำแหน่งของค่าตัวแปรสำหรับ ?subject ในแพทเทิร์นที่ (1) และค่าที่เหลือเป็น ค่าของ predicateData และ objectData ตามลำดับ ซึ่งสามารถแสดงตารางการจับคู่ตามตารางที่ 3.1

```
1  Function checkFormatQuery(String sql){
2      for i = 0 to i < sql line
3          if match pattern 1
4              return sqlType[subject]
5          else if match  pattern 2
6              return sqlType[subObj]
7          else if match  pattern 3
8              return sqlType[object]
9          End if
10     End for
11 }
```

รูปที่ 3.15 ซูโดโค้ดการจำแนกรูปแบบของชุดคำสั่งในการสอบถามข้อมูล

ตารางที่ 3.1 ตารางแสดงการจับคู่ค่าของรูปแบบกับค่าของชุดคำสั่งในการสอบถามข้อมูล

ตัวแปร	**ค่าของชุดคำสั่งการสอบถามข้อมูลใน triple 1**
?subject	?X
predicateData	<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
objectData	<http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#GraduateStudent>
ตัวแปร	**ค่าของชุดคำสั่งการสอบถามข้อมูลใน triple 2**
?subject	?X
predicateData	<http://www.lehigh.edu/~zhp2/2004/0401/univbench.owl#takesCourse>
objectData	"http://www.Department0.University0.edu/GraduateCourse0"

เมื่อทำการแยกแพทเทิร์นของชุดคำสั่งเรียบร้อยแล้ว โปรแกรมจะทำการอ่านข้อมูลจากเอชดีเอฟเอสที่ถูกเก็บไว้ในระบบเป็นบรรทัด และโปรแกรมจะทำการแยกค่าของ ประธาน คุณสมบัติ และค่าคุณสมบัติ ด้วยค่า “ ” (space) เพื่อนำค่าของข้อมูลที่อ่านได้จากการแยกข้อมูลในแต่ละบรรทัดกำหนดค่าให้กับตัวแปร ประธาน คุณสมบัติ และค่าคุณสมบัติตามลำดับ

ซึ่งค่าของคีย์และแวลูที่ออกจากแมปมี 3 แบบดังตารางที่ 3.2

ตารางที่ 3.2 แสดงรูปแบบของคีย์และแวลูที่ออกจากแมป

แบบที่	outputKey	outputValue
1	?subject|subject	Predicate+object
2	?subject+?object|subject+object	Predicate
3	?object|object	Predicate+subject

ในค่าของคีย์ที่ถูกส่งออกจากแมปนั้นจะส่งค่าชื่อของตัวแปรที่ต้องการและค่าของตัวแปรที่หาได้ ส่วนแวลูที่ส่งออกมานั้นจะเป็นค่าในส่วนที่นอกเหนือจากค่าของคีย์ที่ออกจากแมป ทำการส่งค่าคีย์ list (K2, v2) ส่งต่อให้กับรีดิวซ์ต่อไป สามารถเขียนซูโดโค้ดตามรูปที่ 3.16

```
function map
    Map(inputKey, inputValue, outputContext)
        int i = 0;

        var sql = readSparql
        var newSql = convert sparql to BGP
        HashMap sqlMap = checkFormatQuery(newSql)

        while(value is not null)
            var[] tmpValue = split value by space //tmpValue is var[3]
            if(i%3 = 0)
                subject = tmpValue[0]
            else if (i%3 = 1)
                predicate = tmpValue[1]
            else if (i%3 = 2)
                object = tmpValue[2]
            end if
            i++

            if(subject is not null && predicate is not null && object is not null)

                for(sqlMap)
                    if(key of sqlMap eq pattern 1)
                        if(value of sqlMap eq predicate && value of sqlMap eq object)
                            outputValue = ?subject | subject
                            outputKey = predicate + object
                        end if
                    else if(key of sqlMap eq pattern 2)
                        if(value of sqlMap eq predicate)
                            outputValue = ?subject + ?object | subject + objecct
                            outputKye = predicate
                        end if
                    else if(key of sqlMap eq pattern 3)
                        if(value of sqlMap eq predicate && value of sqlMap subject)
                            outputValue = ?object | object
                            outputKey = subject + predicate
                        end if
                    end if

                    write outputContext outputKey, outputValue
                end for
            end if

        End while
    End Map
End function

Function checkFormatQuery(String sql){
    for i = 0 to i < sql line
        if match pattern 1
            return sqlType[subject]
        else if match  pattern 2
            return sqlType[subObj]
        else if match  pattern 3
            return sqlType[object]
        End if
    End for
}
```


รูปที่ 3.16 ซูโดโค้ดการทำงานของแมป

(2) หลักการของรีดิวซ์

เมื่อส่วนของแมปทำการอ่านข้อมูลเรียบร้อยแล้วกระบวนการรีดิวซ์จะทำการประมวลผลจากคีย์ที่ถูกส่งออกจากแมป โดยทำการวนอ่านค่าของแวลูที่ถูกส่งออกมา และตรวจสอบค่าของคีย์ในกรณีที่มีคีย์ที่เป็นตัวแปรเดียวกัน จะทำการประมวลผลในกรณีที่ชุดคำสั่งต้องการฟิวเตอร์ค่าของตัวแปร

เมื่อทำการฟิวเตอร์เรียบร้อยแล้ว โปรแกรมจะทำการเขียนค่าแวลูที่ได้ลงไฟล์เพื่อเป็นผลลัพธ์ตามโดนแมนหลักการของรีดิวซ์

$$\text{domainReduce (K2, list (v2))} \longrightarrow \text{list (v3)}$$

ซึ่งสามารถเขียนซูโดโค้ดตามรูปที่ 3.17

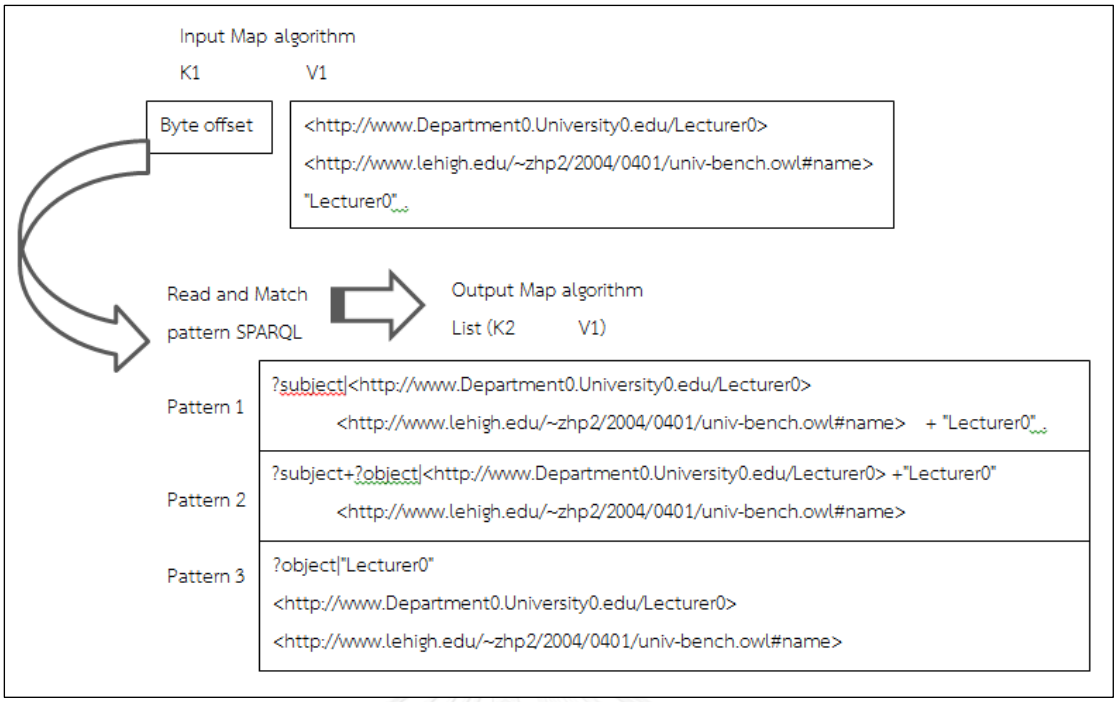
```
1  function reduce(inputKey, inputValues, context)
2      Map resultMap
3      while(value is not null)
4          var[] newValue = value split by | //newValue is var[2]
5          if(resultMap not contant newValue[1])
6              newValue[1] put in resultMap
7          end if
8      End while
9
10     for(resultMap)
11         outputValue = value of resultMap
12     end for
13
14     write context Text, ouputValue
15 End function
```

รูปที่ 3.17 ซูโดโค้ดการทำงานของรีดิวซ์

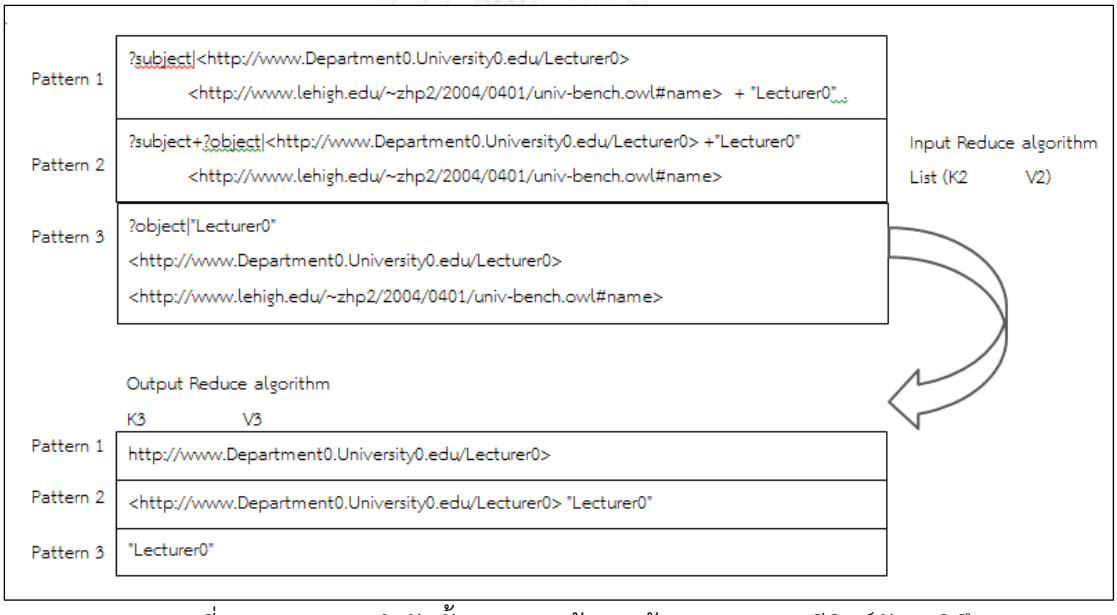
เมื่อได้ชุดของข้อมูลผลลัพธ์ที่ผ่านการรีดิวซ์เรียบร้อยแล้ว จะนำข้อมูลผลลัพธ์ออกมาเป็นไฟล์ แล้วตรวจสอบความถูกต้องของผลลัพธ์ในการสอบถามข้อมูล โดยใช้ สปาร์เคิลจียูไอที่ถูกสร้างขึ้นโดยแหล่งกำเนิดข้อมูลตามแหล่งข้อมูล[17]

3.3 สรุปภาพการประมวลผลด้วยแมปรีดิวซ์

จากรูปที่ 3.18 และ รูปที่ 3.19 แสดงความสัมพันธ์ระหว่างข้อมูลเข้าและออกของแมปและรีดิวซ์ ซึ่งเริ่มต้นด้วยข้อมูลเข้าของแมปอัลกอริทึม อ่านข้อมูลที่อยู่ในคลัสเตอร์ได้เป็น ข้อมูลคีย์และแวลูขาเข้าของแมป (K1, V1) แล้วจัดรูปแบบข้อมูลให้เกิดเป็นคีย์และแวลู เพื่อให้เกิดเป็นข้อมูลขาออกคีย์และแวลู และข้อมูลขาเข้าของรีดิวซ์ (List (K2, V2)) เพื่อทำการประมวลให้เกิดผลลัพธ์สุดท้ายที่ถูกต้อง (K3, V3)



รูปที่ 3.18 แสดงลำดับขั้นตอนของข้อมูลเข้าและออกของแมปอัลกอริทึม



รูปที่ 3.19 แสดงลำดับขั้นตอนของข้อมูลเข้าและออกของรีดิวซ์อัลกอริทึม

3.4 การประเมินและวัดประสิทธิภาพของการถ่ายโอนและสอบถามข้อมูล

การประเมินและวัดประสิทธิภาพด้านความเร็วของการถ่ายโอนข้อมูลและสอบถามข้อมูลในฮาดูป โดยเปรียบเทียบเวลาที่ใช้ในการค้นหาข้อมูลกับชุดข้อมูลที่ใช้ทดสอบในงานวิจัย โดยใช้รูปแบบของ สปาร์เคิล ที่ได้กำหนดไว้ในเพื่อทำการค้นหาข้อมูลระหว่างลิงค์เดต้าและฮาดูป โดยระบบจะมีการเขียนไฟล์เพื่อเก็บล็อก (Log) สำหรับการวัดผลของการถ่ายโอนข้อมูลและ ระยะเวลาในการสอบถามข้อมูล ดังนี้

(1) ไฟล์ logMain.log

ไฟล์ล็อกนี้จะแสดงเวลาการทำงานของแต่ละเมท็อด (Method) ในแต่ละคลาส ตามรูปที่ 3.20

```
[2016-03-30 02:48:11,508][INFO ][ConvertTDRToXMLBatch][main] - ----------------------------------------
[2016-03-30 02:48:11,508][INFO ][ConvertTDRToXMLBatch][main] - |    Start.. ConvertTDRToXMLBatch       |
[2016-03-30 02:48:11,509][INFO ][ConvertTDRToXMLBatch][main] - ----------------------------------------
[2016-03-30 02:48:11,540][INFO ][ConvertTDRToXMLBatch][main] - listFile size:3284
[2016-03-30 02:48:11,540][INFO ][ConvertTDRToXMLBatch][main] - File RDF: D:\HadoopProject\rdfdata\data\OWL
[2016-03-30 02:48:12,251][INFO ][ProcessRDFServiceImpl][main] - Start convertRDFData
[2016-03-30 02:48:12,252][INFO ][ProcessRDFServiceImpl][main] - validateRDFInputFile[D:\HadoopProject\rdfdata\data\OWL\University0_0.owl]
[2016-03-30 02:48:12,330][INFO ][ProcessRDFServiceImpl][main] - Check WellFormed of RDF
[2016-03-30 02:48:12,331][INFO ][ProcessRDFServiceImpl][main] - convertXMLtoNTriple
[2016-03-30 02:48:12,622][WARN ][RDFDefaultErrorHandler][main] - unknown-source: {W136} Relative URIs are not permitted in RDF: specifically
[2016-03-30 02:48:12,629][ERROR][RDFDefaultErrorHandler][main] - file:///D:/HadoopProject/ConvertXMLToTriple/RDF/XML(line 9 column 28): {E215
[2016-03-30 02:48:12,630][WARN ][RDFDefaultErrorHandler][main] - file:///D:/HadoopProject/ConvertXMLToTriple/RDF/XML(line 9 column 28): {W136
[2016-03-30 02:48:13,585][INFO ][ProcessRDFServiceImpl][main] - Model size 8520
[2016-03-30 02:48:13,585][INFO ][ProcessRDFServiceImpl][main] - Write File Success..D:\HadoopProject\rdfdata\data\OWL\University0_0.owl
[2016-03-30 02:48:13,586][INFO ][performanceAppLog][main] - [30/03/2559 02:48:12 - 30/03/2559 02:48:13 Time : 1073][ProcessRDFServiceImpl.cor
```

รูปที่ 3.20 ตัวอย่างไฟล์ logMain.log

บทที่ 4

ผลการทดสอบการถ่ายโอนข้อมูลและสอบถามข้อมูล

4.1 สภาพแวดล้อมของระบบ

จากตารางที่ 4.1 แสดงค่าและสภาพแวดล้อมที่ถูกกำหนดขึ้นเพื่อทดสอบและการถ่ายโอนข้อมูลและสอบถามข้อมูลในกรอบการทำงานของฮาดูปโดยอาศัยหลักการของแมปรีดิวซ์

ตารางที่ 4.1 การตั้งค่าและสภาพแวดล้อมของระบบที่ใช้ทดสอบการถ่ายโอนและสอบถามข้อมูลด้วยแมปรีดิวซ์

การตั้งค่าสภาพแวดล้อมสำหรับระบบ	พารามิเตอร์
Platform	CentOS release 6.7
Ram	8 GB
Hard disk	50 GB
Hadoop	2.7.1
Data node	1
Java	1.7.0_95

จากตารางที่ 4.2 แสดงค่าและสภาพแวดล้อมที่ถูกกำหนดขึ้นเพื่อทดสอบการสอบถามข้อมูลด้วยจีน่าฟูเซกิ (Jena Fuseki) เพื่อใช้ทดสอบในการเปรียบเทียบระยะเวลาการเข้าถึงข้อมูลระหว่างการเข้าถึงข้อมูลด้วยแมปรีดิวซ์และการเข้าถึงข้อมูลด้วยจีน่าฟูเซกิ

ตารางที่ 4.2 การตั้งค่าและสภาพแวดล้อมของระบบที่ใช้ทดสอบการถ่ายโอนและสอบถามข้อมูลด้วยจีน่าฟูเซกิ

การตั้งค่าสภาพแวดล้อมสำหรับระบบ	พารามิเตอร์
Platform	CentOS release 6.7
Ram	8 GB
Hard disk	50 GB
Fuseki	1.0.0
Java	1.7.0_95

4.2 ชุดข้อมูลการทดสอบการถ่ายโอนข้อมูลและสอบถามข้อมูล

ชุดข้อมูลที่นำมาใช้ในการทดสอบการถ่ายโอนข้อมูลและสอบถามข้อมูล คือ The Lehigh University Benchmark (LUBM) เป็นชุดข้อมูลที่ถูกสร้างขึ้นเพื่ออำนวยความสะดวกในการประเมินผลของซีแมนติกเว็บที่เป็นมาตรฐานและเป็นระบบ โดยประกอบด้วยข้อมูลของมหาวิทยาลัย ซึ่งไฟล์จะถูกแบ่งย่อยเป็นไฟล์เล็กๆ ที่สมบูรณ์ตามรูปแบบสัญลักษณ์ของเอ็กซ์เอ็มแอลถูกแบ่งเป็นไฟล์จำนวน 3,284 ไฟล์ ซึ่งมีขนาดรวมทั้งสิ้น 1.74 กิกะไบต์ (GB) ใช้เวลาในการแปลงให้อยู่ในรูปของเอ็นทริปเปิ้ลทั้งหมด 37.40 นาที ได้ขนาดไฟล์ทั้งหมด 3.76 กิกะไบต์ หลังจากการแปลงข้อมูลตัวอย่างข้อมูล The Lehigh University Benchmark (LUBM) ตามรูปที่ 4.1

```
<?xml version="1.0" encoding="UTF-8" ?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:ub="http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#">


<owl:Ontology rdf:about="">
<owl:imports rdf:resource="http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl" />
</owl:Ontology>

<ub:University rdf:about="http://www.University0.edu">
   <ub:name>University0</ub:name>
</ub:University>

<ub:Department rdf:about="http://www.Department0.University0.edu">
   <ub:name>Department0</ub:name>
   <ub:subOrganizationOf>

        <ub:University rdf:about="http://www.University0.edu" />
   </ub:subOrganizationOf>
</ub:Department>
</rdf:RDF>
```

รูปที่ 4.1 ตัวอย่างข้อมูล The Lehigh University Benchmark (LUBM)

4.3 ชุดคำสั่งในการสอบถามข้อมูล

จากชุดคำสั่งในการสอบถามข้อมูลที่นำมาใช้ในการทดสอบการถ่ายโอนและสอบถามข้อมูล นำมาจากแหล่งของข้อมูลที่นำมาทดสอบ ซึ่งชุดคำสั่งดังกล่าวถูกกำหนดไว้เพื่อทดสอบสำหรับข้อมูล The Lehigh University Benchmark (LUBM) มีด้วยกันทั้งหมด 14 ชุดคำถาม พร้อมกับผลลัพธ์จากชุดสอบถาม ซึ่งในวิทยานิพนธ์นี้ได้ทำการเลือกชุดคำสั่งในการสอบถามมาทั้งหมด 7 ชุดคำสั่ง ซึ่งจะอ้างอิงชุดคำสั่งตามแหล่งข้อมูลตัวอย่าง และทำการจัดกลุ่มชุดคำสั่งใหม่โดยคัดเลือกชุดคำสั่งในการสอบถามข้อมูลที่แตกต่างกันออกไป ซึ่งสามารถจัดกลุ่มให้กับชุดสอบถามข้อมูลดังกล่าวได้ ดังนี้

4.3.1 กลุ่มของชุดสอบถามข้อมูลที่ 1

กลุ่มชุดคำสั่งในการสอบถามนี้ จัดเป็นชุดคำสั่งที่คัดเลือกคำสั่งแบบทั่วไป นั่นคือเป็นชุดคำสั่งในการสอบถามที่ไม่ซับซ้อน สอบถามเพียง 1 ตัวแปร และลำดับชั้นในการสอบถามมีเพียง 1 ชั้น ประกอบไปด้วย ชุดคำสั่งที่ 6 และ ชุดคำสั่งที่ 14

1) ชุดการสอบถามข้อมูลที่ 6

สำหรับชุดการสอบถามที่ 6 เป็นเพียงการสอบถามเพียงหนึ่งระดับเท่านั้น และได้กำหนดค่าของตัวแปรไว้อย่างชัดเจน แต่ถือได้ว่าเป็นการสอบถามที่ไม่ซับซ้อน

```
1  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2  PREFIX ub: <http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#>
3  SELECT ?X
4  WHERE
5  {?X rdf:type ub:Student}
```

รูปที่ 4.2 ชุดการสอบถามข้อมูลที่ 6

2) ชุดการสอบถามชุดที่ 6 ในรูปแบบของบีจีพี

```
1  (project (?X)
2    (BGP (triple ?X
3                 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
4                 <http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#Student>)))
```

รูปที่ 4.3 ชุดการสอบถามชุดที่ 6 ในรูปแบบของบีจีพี

3) ผลการทดสอบในการสอบถามโดยใช้ชุดการสอบถามชุดที่ 6

จำนวนข้อมูลที่ค้นพบตามความต้องการของชุดสอบถามข้อมูลจำนวน 7790 แถว ใช้เวลาในการสอบถามข้อมูล 90.50 นาที

4) ชุดการสอบถามข้อมูลที่ 14

ถือว่าเป็นชุดสอบถามข้อมูลที่ง่ายที่สุดในชุดสอบถามทั้งหมด ซึ่งมีการเลือกข้อมูลที่ต่ำไม่มีความซับซ้อนของข้อมูลเป็นลำดับชั้นหรือการเปรียบเทียบใดๆ

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX ub: <http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#>
SELECT ?X
WHERE
{?X rdf:type ub:UndergraduateStudent}
```

รูปที่ 4.4 ชุดการสอบถามข้อมูลที่ 14

5) ชุดการสอบถามชุดที่ 14 ในรูปแบบของบีจีพี

```
(project (?X)
  (BGP (triple ?X
               <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
               <http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#UndergraduateStudent>)))
```

รูปที่ 4.5 ชุดการสอบถามชุดที่ 14 ในรูปแบบของบีจีพี

6) ผลการทดสอบในการสอบถามโดยใช้ชุดการสอบถามชุดที่ 14

จำนวนข้อมูลที่ค้นพบตามความต้องการของชุดสอบถามข้อมูลจำนวน 5916 แถว ใช้เวลาในการสอบถามข้อมูล 95.12 นาที

4.3.2 กลุ่มของชุดสอบถามข้อมูลที่ 2

กลุ่มชุดคำสั่งในการสอบถามนี้ จัดเป็นชุดคำสั่งที่คัดเลือกที่มีความซับซ้อนขึ้นมาอีก 1 ระดับ แต่ยังคงสอบถามค่าตัวแปรเพียง 1 ตัวแปร โดยจัดว่าเป็นการสอบถามข้อมูลที่มีเงื่อนไขเพิ่มขึ้น โดยคัดเลือกชุดคำสั่งที่ 1 ชุดคำสั่งที่ 3 และชุดคำสั่งที่ 5

1) ชุดการสอบถามข้อมูลที่ 1

สำหรับชุดการสอบถามชุดที่ 1 เป็นการสอบถามที่มีการสอบถามข้อมูลในวงกว้างเนื่องจากเป็นการสอบถามที่สอบถามเพียง 1 ตัวแปร โดยไม่มีการเลือกแบบลำดับชั้น ไม่มีความซับซ้อนของเงื่อนไข

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX ub: <http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#>
SELECT ?X
WHERE
{?X rdf:type ub:GraduateStudent .
 ?X ub:takesCourse http://www.Department0.University0.edu/GraduateCourse0}
```

รูปที่ 4.6 ชุดการสอบถามข้อมูลที่ 1

2) ชุดการสอบถามชุดที่ 1 ในรูปแบบของบีจีพี

```
(project (?X)
  (BGP
    (triple ?X
    <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
    <http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#GraduateStudent>)
    (triple ?X
    <http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#takesCourse>
    "http://www.Department0.University0.edu/GraduateCourse0")
  ))
```

รูปที่ 4.7 ชุดการสอบถามชุดที่ 1 ในรูปแบบของบีจีพี

3) ผลการทดสอบในการสอบถามโดยใช้ชุดการสอบถามชุดที่ 1

จำนวนข้อมูลที่ค้นพบตามความต้องการของชุดสอบถามข้อมูลจำนวน 4 แถว ใช้เวลาในการสอบถามข้อมูล 48.21 นาที

4) ชุดการสอบถามข้อมูลที่ 3

เป็นชุดการสอบถามที่มีลักษณะเหมือนชุดการสอบถามที่ 1 แต่ลำดับเงื่อนไขจะขยายขอบเขตของคำตอบที่กว้างขึ้น

```
1  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2  PREFIX ub: <http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#>
3  SELECT ?X
4  WHERE
5  { ?X rdf:type ub:Publication .
6    ?X ub:publicationAuthor
7          http://www.Department0.University0.edu/AssistantProfessor0}
```

รูปที่ 4.8 ชุดการสอบถามข้อมูลที่ 3

5) ชุดการสอบถามชุดที่ 3 ในรูปแบบของบีจีพี

```
1  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2  PREFIX ub: <http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#>
3  SELECT ?X
4  WHERE
5  { ?X rdf:type ub:Publication .
6    ?X ub:publicationAuthor
7          http://www.Department0.University0.edu/AssistantProfessor0}
```

รูปที่ 4.9 ชุดการสอบถามชุดที่ 3 ในรูปแบบของบีจีพี

6) ผลการทดสอบในการสอบถามโดยใช้ชุดการสอบถามชุดที่ 3

จำนวนข้อมูลที่ค้นพบตามความต้องการของชุดสอบถามข้อมูลจำนวน 6 แถว ใช้เวลาในการสอบถามข้อมูล 59.08 นาที

7) ชุดการสอบถามข้อมูลที่ 5

ชุดการสอบถามนี้ถือว่าเป็นความสัมพันธ์ระหว่าง 2 เงื่อนไข ซึ่งผลลัพธ์ของเงื่อนไขแรก จะต้องสอดคล้องกับเงื่อนไขที่ 2 ของชุดคำสั่ง

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX ub: <http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#>
SELECT ?X
WHERE
{ ?X rdf:type ub:Person .
  ?X ub:memberOf <http://www.Department0.University0.edu>}
```

รูปที่ 4.10ชุดการสอบถามข้อมูลที่ 5

8) ชุดการสอบถามชุดที่ 5 ในรูปแบบของบีจีพี

```
(project (?X)
  (BGP
    (triple ?X
                <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
                <http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#Person>)
    (triple ?X
                <http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#memberOf>
                <http://www.Department0.University0.edu>)
  ))
```

รูปที่ 4.11 ชุดการสอบถามชุดที่ 5 ในรูปแบบของบีจีพี

9) ผลการทดสอบในการสอบถามโดยใช้ชุดการสอบถามชุดที่ 5

จำนวนข้อมูลที่ค้นพบตามความต้องการของชุดสอบถามข้อมูลจำนวน 719 แถว ใช้เวลาในการสอบถามข้อมูล 63.32 นาที

4.3.3 กลุ่มของชุดสอบถามข้อมูลที่ 3

กลุ่มชุดคำสั่งในการสอบถามนี้ จัดเป็นชุดคำสั่งที่คัดเลือกที่มีความซับซ้อนซึ่งสร้างลำดับความสัมพันธ์สำหรับ 2 เงื่อนไขที่ใช้ในการสอบถามข้อมูล แสดงให้เห็นผลลัพธ์ที่มีความสัมพันธ์กัน โดยเลือก ชุดคำสั่งที่ 6 เป็นตัวแทนของกลุ่มชุดคำสั่งสอบถามนี้

1) ชุดการสอบถามข้อมูลที่ 6

ชุดสอบถามที่ใช้ทฤษฎีของการผกผัน ซึ่งประกอบด้วยลักษณะย่อยอยู่ภายใต้เงื่อนไขแรก ที่จะต้องสอดคล้องกับเงื่อนไขที่ 2

```
1  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2  PREFIX ub: <http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#>
3  SELECT ?X
4  WHERE
5  { ?X rdf:type ub:Person .
6    <http://www.University0.edu> ub:hasAlumnus ?X}
```

รูปที่ 4.12 ชุดการสอบถามข้อมูลที่ 6

2) ชุดการสอบถามชุดที่ 6 ในรูปแบบของบีจีพี

```
1  (project (?X)
2    (BGP
3      (triple ?X
4                  <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
5                  <http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#Person>)
6      (triple <http://www.University0.edu>
7                  <http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#hasAlumnus>
8                  ?X)
9    ))
```

รูปที่ 4.13 ชุดการสอบถามชุดที่ 6 ในรูปแบบของบีจีพี

3) ผลการทดสอบในการสอบถามโดยใช้ชุดการสอบถามชุดที่ 6

จำนวนข้อมูลที่ค้นพบตามความต้องการของชุดสอบถามข้อมูลจำนวน 1 แถว ใช้เวลาในการสอบถามข้อมูล 49.22 นาที

4.3.4 กลุ่มของชุดสอบถามข้อมูลที่ 4

กลุ่มชุดคำสั่งในการสอบถามนี้ จัดเป็นชุดคำสั่งที่คัดเลือกที่มีความซับซ้อนขึ้นมาเพื่อสอบถามค่าของ 2 ตัวแปร โดยอาศัยความสัมพันธ์ระหว่าง 2 ตัวแปร ที่มีลำดับความสัมพันธ์สำหรับ 2 เงื่อนไขที่ใช้ในการสอบถามข้อมูลซึ่งมีค่าของตัวเชื่อมระหว่าง 2 ชุดคำสั่ง โดยเลือกชุดคำสั่งที่ 7 เป็นตัวแทนของกลุ่มชุดคำสั่งสอบถามนี้

1) ชุดการสอบถามข้อมูลที่ 7

ชุดสอบถามนี้จะเพิ่มคุณสมบัติของการเลือกข้อมูลที่สูงขึ้น และเพิ่มเงื่อนไขในการสอบถามข้อมูลเป็น 2 ตัวแปร

```
1  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2  PREFIX ub: <http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#>
3  SELECT ?X, ?Y
4  WHERE
5  {?X rdf:type ub:Student .
6    ?Y rdf:type ub:Course .
7    ?X ub:takesCourse ?Y .
8    <http://www.Department0.University0.edu/AssociateProfessor0>
9          ub:teacherOf ?Y}
```

รูปที่ 4.14ชุดการสอบถามข้อมูลที่ 7

2) ชุดการสอบถามชุดที่ 7 ในรูปแบบของบีจีพี

```
(project (?X ?Y)
  (BGP
    (triple ?X
                <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
                <http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#Student>)
    (triple ?Y
                <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
                <http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#Course>)
    (triple ?X
                <http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#takesCourse>
                ?Y)
    (triple <http://www.Department0.University0.edu/AssociateProfessor0>
                <http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl#teacherOf>
                ?Y)
  ))
```

รูปที่ 4.15 ชุดการสอบถามชุดที่ 7 ในรูปแบบของบีจีพี

3) ผลการทดสอบในการสอบถามโดยใช้ชุดการสอบถามชุดที่ 7

จำนวนข้อมูลที่ค้นพบตามความต้องการของชุดสอบถามข้อมูลจำนวน 67 แถว ใช้เวลาในการสอบถามข้อมูล 108.55 นาที

4.4 สรุปผลการทดสอบในการสอบถามข้อมูล

จากผลการทดลองการสอบถามข้อมูลผลการทดลอง สามารถสรุปเป็นตารางในการทดลองได้ดังตารางที่ 4.3 โดยอ้างอิงตามกลุ่มข้อมูลในวิทยานิพนธ์ได้จัดขึ้น และด้วยคำตอบที่ได้จากการสอบถามข้อมูลด้วยแมปรีดิวซ์ และคำตอบที่ได้จากแหล่งข้อมูลนั้น เมื่อนำมาเปรียบเทียบและตรวจสอบจะได้ผลลัพธ์ที่ตรงกัน โดยผลลัพธ์จากแมปรีดิวซ์ที่ได้จะถูกเขียนเป็นไฟล์ โดยสามารถดาวน์โหลดไฟล์ผลลัพธ์ดังกล่าว จากหน้าจอที่แสดงข้อมูลและสถานะของฮาดูปทั้งระบบ

ตารางที่ 4.3 สรุปผลการทดลอง

กลุ่มชุดสอบถามที่	ชุดสอบถามข้อมูล	จำนวนข้อมูลที่พบ (แถว)	แมปรีดิวซ์ เวลาที่ใช้(นาที)	ฟูเซกิ เวลาที่ใช้(นาที)
G1	Q6	7,790	90.50	58.46
	Q14	5,916	95.12	69.26
G2	Q1	4	48.21	58.24
	Q3	6	58.08	77.08
	Q5	719	63.32	112.13
G3	Q6	1	49.22	103.21
G4	Q7	67	108.55	203.07

จากกลุ่มชุดสอบถามข้อมูลที่ 1 ประกอบด้วย ชุดคำสั่งที่ 6 และชุดคำสั่งที่ 14 ซึ่งไม่มีเงื่อนไขในการค้นหาข้อมูลใดๆ เรียกได้ว่าไม่ได้มีการฟิวเตอร์ออกในหลายขั้นตอน พบว่าเวลาที่ฟูเซกิใช้ในการสอบถามข้อมูลใช้เวลาน้อยกว่าการสอบถามข้อมูลจากแมปรีดิวซ์ ซึ่งสาเหตุที่แมปรีดิ๊วใช้เวลามากกว่านั้นเกิดจากการประมวลผลของแมปใช้เวลานาน นอกจากนี้หลังจากที่จบกระบวนการทำงานของแมปเรียบร้อยแล้วยังต้องทำการตรวจสอบข้อมูลและฟิวเตอร์ตามเงื่อนไขของรีดิวซ์ที่ถูกเขียนไว้ ทำให้ใช้เวลาค่อนข้างนาน

กลุ่มชุดสอบถามข้อมูลที่ 2 ประกอบไปด้วย ชุดคำสั่งที่ 1 ชุดคำสั่งที่ 3 และชุดคำสั่งที่ 5 ซึ่งเป็นชุดสอบถามที่มีลักษณะที่คล้ายกันคือมีเงื่อนไขสำหรับตัวแปรมากกว่า 1 เงื่อนไข พบว่าเวลาที่ใช้ในการสอบถามข้อมูลจากแมปรีดิวซ์ใช้เวลาน้อยกว่าเวลาที่ใช้ในฟูเซกิ

กลุ่มชุดสอบถามข้อมูลที่ 3 โดยชุดสอบถามที่ 6 พบว่าฟูเซกิใช้เวลานานกว่าปกติซึ่งใช้เวลาเป็น 2 เท่าของแมปรีดิวซ์ เนื่องจากเกิดความสัมพันธ์กันด้วยตัวมันเอง อาจทำให้ฟูเซกิใช้เวลานานขึ้น

นอกจากนี้กลุ่มชุดสอบถามที่ 4 การสอบถามข้อมูลจากฟูเซกิใช้เวลาเป็นสองเท่า ของการสอบถามข้อมูลจากแมปรีดิวซ์ อาจเกิดจากชุดการสอบถามที่ 7 มีการสอบถามข้อมูลที่ค่อนข้างซับซ้อนทำให้เมื่อสอบถามข้อมูลมาได้แล้ว ต้องการเวลาในการประมวลผลหรือฟิวเตอร์ข้อมูลที่มากกว่าปกติ

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1. บทสรุปการถ่ายโอนและสอบถามข้อมูล

การพัฒนาการถ่ายโอนข้อมูล ซึ่งทดสอบด้วยข้อมูล The Lehigh University Benchmark (LUBM) ซึ่งไฟล์จะถูกแบ่งย่อยเป็นไฟล์เล็กๆ ที่สมบูรณ์ตามรูปแบบสัญลักษณ์ของเอ็กซ์เอ็มแอล จำนวน 3,284 ไฟล์ ซึ่งมีขนาดรวมทั้งสิ้น 1.74 กิกะไบต์ (GB) ใช้เวลาในการแปลงให้อยู่ในรูปของเอ็นทริปเปิ้ลทั้งหมด 37.40 นาที ได้ขนาดไฟล์ทั้งหมด 3.76 กิกะไบต์ แล้วนำเข้าข้อมูลดังกล่าวเข้าสู่ฮาดูปเฟรมเวิร์ค ด้วยชุดคำสั่งเฉพาะของฮาดูป โดยข้อมูลจะถูกจัดเก็บไว้ที่เอชดีเอฟเอชเพื่อถูกดำเนินการต่อไป

หลังจากนำข้อมูลเข้าสู่เอชดีเอฟเอชเรียบร้อยแล้วกระบวนการการสอบถามข้อมูลจะทำการอ่านข้อมูลโดยอาศัยหลักการของแมปและรีดิวซ์ในการสอบถาม โดยเริ่มต้นด้วยกระบวนการแมปทำการอ่านค่าของสปาร์เคิลเพื่อนำมาแปลงให้อยู่ในรูปบีจีพี และตรวจสอบเพื่อแยกชุดสอบถามให้ตรงตามรูปแบบที่ถูกกำหนดไว้ จากนั้นทำการอ่านข้อมูลที่มีถูกเก็บอยู่ในฮาดูป แล้วจัดรูปแบบข้อมูลให้ตรงกับรูปแบบของคีย์และแวลูในหลักการของแมป โดยสามารถแบ่งตามลักษณะที่สอดคล้องกับรูปแบบของชุดคำสั่งในการสอบถาม เพื่อทำการเปรียบเทียบข้อมูลเพื่อให้ได้ชุดของผลลัพธ์ที่จะส่งออกไปให้กระบวนการของรีดิวซ์ประมวลผล หลังจากที่ได้ผลลัพธ์เรียบร้อยแล้วกระบวนการรีดิวซ์จะทำการตรวจสอบและทำการเชคข้อมูลเพื่อให้ตรงตามเงื่อนไข ทำการกรองผลลัพธ์ตัวที่ซ้ำซ้อนออกจากผลลัพธ์ เพื่อให้ได้ข้อมูลที่ตรงตามความต้องการ ซึ่งการถ่ายโอนและสอบถามข้อมูลจะแสดงค่าของผลลัพธ์และเวลาที่ใช้ในการสอบถามข้อมูลดังกล่าวเพื่อวัดประสิทธิภาพของการสอบถามข้อมูลดังกล่าวด้วยการเปรียบเทียบข้อมูลโดยใช้เครื่องมือของจีน่า ที่เรียกว่า จีน่าฟูเซกิ ซึ่งเป็นเครื่องมือที่ช่วยในการสอบถามข้อมูลที่รองรับการสอบถามข้อมูลแบบอาร์ดีเอฟเอ็กซ์เอ็มแอล และในรูปแบบอื่นๆ เช่น เอ็นทริปเปิ้ล เป็นต้น นอกจากนี้ยังสามารถใช้ตรวจสอบความถูกต้องของข้อมูลและสปาร์เคิลได้ด้วย

โดยผลการทดลองของการถ่ายโอนและสอบถามข้อมูลพบว่า จากกลุ่มของชุดคำสั่งที่ 1 พบว่าเวลาที่ใช้ของแมปรีดิวซ์ ใช้เวลาเยอะกว่าการใช้งานของฟูเซกิ เนื่องจากการทำงานของแมปรีดิวซ์จะต้องทำการอ่านไฟล์ทั้งหมดเพื่อทำการสร้างคีย์และแวลู แล้วส่งค่าให้รีดิวซ์ทำการประมวลผล ทำให้เวลาที่ใช้นานกว่าการทดสอบด้วยฟูเซกิ ซึ่งสามารถอ่านข้อมูลและประมวลผลได้เลยเมื่อจบการ

สอบถามโดยไม่ต้องประมวลผลเพื่อเงื่อนไขอื่นๆ ต่ออีกสามารถกล่าวได้ว่าในกรณีที่ไม่มีเงื่อนไขที่ซับซ้อนมากกว่า 1 ระดับ ฟูเซกิจะสามารถสอบถามข้อมูลโดยใช้เวลาน้อยกว่าแมปรีดิวซ์ ซึ่งแตกต่างจากกลุ่มชุดคำสั่งที่ 2 ที่เพิ่มเงื่อนไขในการสอบถามขึ้นมาอีก 1 ระดับ พบว่า เวลาที่ใช้ในการสอบถามของแมปรีดิวซ์ทำงานได้มีประสิทธิภาพได้ดีกว่าฟูเซกิโดยเมื่อมีผลลัพธ์ที่มากขึ้นฟูเซกิก็จะใช้เวลาเพิ่มขึ้นไปด้วย และจากกลุ่มชุดคำสั่งที่ 3 และ กลุ่มชุดคำสั่งที่ 4 พบว่าเวลาที่ใช้ของแมปรีดิวซ์มีประสิทธิภาพมากกว่าการใช้ฟูเซกิ เกิดเนื่องจากชุดคำสั่งที่มีการสอบถามข้อมูลที่ค่อนข้างซับซ้อนและมีความสัมพันธ์กันระหว่างค่าของตัวแปรที่ต้องการสอบถาม ทำให้การประมวลของฟูเซกิใช้เวลาเกือบเป็นสองเท่าของการสอบถามด้วยวิธีแมปรีดิวซ์

5.2. ข้อเสนอแนะ

จากทรัพยากรที่มีอยู่จำกัดทำให้การทดลองทดสอบโปรแกรมด้วยการสร้างโหนดของฮาดูปเพียงหนึ่งโหนดเท่านั้น ดังนั้นเวลาที่ใช้ในการสอบถามข้อมูลดังกล่าวในบางชุดสอบถามจะใช้เวลามากเกินไปเมื่อเทียบกับชุดการสอบถามข้อมูลชนิดเดียวกันหรือชุดการสอบถามข้อมูลที่มีรูปแบบเดียวกัน ซึ่งการทดสอบการถ่ายโอนและสอบถามข้อมูลสำหรับ 1 โหนดเนื่องจากงานวิจัยใช้ฮาดูปจากแซนบล๊อค วีเอ็มแวร์ (SandBox VMWare) ทำให้ไม่สามารถกำหนดค่าเพื่อเพิ่มจำนวนโหนดได้ ซึ่งยังเป็นข้อบกพร่องของการทดสอบการถ่ายโอนและสอบถามข้อมูลนี้ เพื่อทำให้การทดสอบการถ่ายโอนและสอบถามข้อมูลนี้ได้ประสิทธิภาพมากขึ้นสามารถเพิ่มจำนวนโหนด หรือปรับเปลี่ยนจำนวนโหนดที่มีในระบบให้เพียงพอต่อความต้องการของการใช้ทรัพยากรของระบบเพื่อทดสอบผลลัพธ์ที่มีประสิทธิภาพมากขึ้น

รายการอ้างอิง

1. ชำนาญค้า, ต., ระบบจัดการฐานกฎและระบบจัดการโปรแกรมแนะนำข้อมูล. 2553.
2. Husain, M.F., Doshi, P., & Khan, L. , *Storage and Retrieval of Large RDF Graph Using Hadoop and MapReduce.* 2009: p. 680-686.
3. ทรูเวฟ. จัดการข้อมูลขนาดใหญ่ด้วย *Apache Hadoop* สำหรับองค์กร. 2554; Available from: http://www.throughwave.co.th/2011/11/10/enterprise-apache-hadoop/.
4. *The Lehigh University Benchmark (LUBM).* Available from: http://swat.cse.lehigh.edu/projects/lubm/.
5. *HDFS Users Guide*. Available from: http://archive.cloudera.com/cdh/3/hadoop/hdfs_user_guide.html.
6. *MapReduce* Available from: http://www.cyberthai.com/.
7. *Linked Data - Connect Distributed Data across the Web*. Available from: http://linkeddata.org/guides-and-tutorials.
8. Munindar, P.S., & Shengru, T. , *Exploiting Linked Data to Build Web Applications. IEEE Internet Computing.* 2009.
9. *The Linking Open Data cloud diagram*. Available from: http://lod-cloud.net/.
10. *State of the LOD Cloud 2014*. 2014; Available from: http://lod-cloud.net/.
11. *Tutorial 2: Introducing RDF/XML*. Available from: http://www.linkeddatatools.com/introducing-rdf-part-2.
12. *SPARQL 1.1 Query Language*. Available from: http://www.w3.org/TR/sparql11-query/.
13. Rattanapoka, C., *The Design and Implementation of Computer Traffic Log Searcher System using Hadoop Map/Reduce Framework.,* . The Journal of Industrial Technology, 2012. **8**.

14. Joldzic, O.V., & Vukovic, D. R, *The impact of cluster characteristics on HiveQL query optimization.* 21st Telecommunications Forum Telfor (TELFOR), 2013: p. 837-840.

15. Park, S., *Visualization of Resource Description Framework Ontology Using Hadoop,.* 2013: p. 228-231.

16. Olaf Hartig , C.B., Johann-Christoph Freytag, *Executing SPARQL Queries over the Web of Linked Data.* 2009.

17. *Answer The Lehigh University Benchmark (LUBM).* Available from: http://swat.cse.lehigh.edu/projects/lubm/answers.htm.

ภาคผนวก

ประวัติผู้เขียนวิทยานิพนธ์

นางสาวจุฑามาศ กะวิเศษ สำเร็จการศึกษาระดับปริญญาตรีวิทยาศาสตร์บัณฑิต คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2552 และเข้าศึกษาต่อ ปริญญาโท หลักสูตรวิทยาศาสตร์มหาบัณฑิต สาขาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2555 ประสบการณ์การทำงานเริ่มต้นในบริษัทแอลทีอินโฟเทค ในตำแหน่งจูเนียร์โปรแกรมเมอร์ และในปัจจุบันทำงานบริษัททีเอ็นเอฟไอโซลูชั่น ในตำแหน่งซอฟต์แวร์เอ็นจิเนียร์ ซึ่งเป็นบริษัทเกี่ยวกับการจัดการและพัฒนาระบบสินเชื่อของธนาคาร