A new approach of healthcare data analytics using Hadoop Map/Reduce framework

Natthapong Noosing[1] , Wansa Paoin[2], Worapol Pongpech[1]

[1]Big Data Engineering, Faculty of Engineering, Dhurakij Pundit University, Thailand

[2]Faculty of Medicine, Thammasat University, Thailand

Abstract

The purpose of this experimental research aims to develop a new approach to analyze healthcare data by using MapReduce technique. The healthcare data is called "43 Standard Structure Files" which have been retrieved from the database of Thailand's Ministry of Public Health and gathered by each health service affiliated to the ministry. Three files out of the 43 have been used for this research which consists of 1) the diagnosis of OPD, 2) the diagnosis of IPD, and 3) the patients' personal information. The results show that the average time spent on processing MapReduce on 243,610,347 records of the diagnosis of OPD data is 1.58 hours whereas the average time spent on 201,733,782 records from the diagnosis of IPD file is 1.68 hours. Comparatively, the average time spent on 201,733,782 records of the patients' personal information data is 1.83 hours. This can be concluded that MapReduce technique is an appropriate approach for analysis of healthcare data from health reports retained by Ministry of Public Health.

Keywords: Big data, map reduce, data analytics.

Received 20 March 2017; Accepted 31 May 2017

Correspondence: Natthapong Noosing, Big Data Engineering, Faculty of Engineering, Dhurakij Pundit University, Thailand 110/1-4 Prachachuen Road, Tungsonghong, Laksi, Bangkok, Thailand, 10210 (Tel.: +66-8696-88553; E-mail address: 585162020014@dpu.ac.th).

การพัฒนาแนวทางใหม่ในการวิเคราะห์ข้อมูลรายงานการป่วยของกระทรวงสาธารณสุข โดยวิธี Map Reduce

นัฐพงษ์ หนูสิงห์[1], วรรษา เปาอินทร์[2], วรพล พงษ์เพ็ชร[1]

[1]สาขาวิศวกรรมข้อมูลขนาดใหญ่ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยธุรกิจบัณฑิตย์

[2]คณะแพทยศาสตร์ มหาวิทยาลัยธรรมศาสตร์

บทคัดย่อ

การวิจัยนี้เป็นแบบการทดลอง มีวัตถุประสงค์เพื่อพัฒนาแนวทางใหม่ในการวิเคราะห์ข้อมูลรายงานการป่วยของกระทรวงสาธารณสุข โดยใช้ Map Reduce กับข้อมูลจากโครงสร้างฐานข้อมูลด้านการแพทย์และสุขภาพในรูปแบบ 43 แฟ้ม ประกอบด้วย 1) แฟ้มข้อมูลวินิจฉัยโรคของผู้ป่วยนอกและผู้มารับบริการ (DIAGNOSIS_OPD) 2) แฟ้มข้อมูลวินิจฉัยโรคของผู้ป่วยใน (DIAGNOSIS_IPD) และ 3) แฟ้มข้อมูลทั่วไปของประชาชนในเขตรับผิดชอบและผู้ที่มาใช้บริการ (PERSON) รวบรวมจากหน่วยงานบริการในสังกัดกระทรวงสาธารณสุขทั่วประเทศ ผลการทดลองพบว่า การประมวลผลกลุ่มรายงานการป่วยผู้ป่วยนอกจำนวน 243,610,347 records ใช้เวลาเฉลี่ยประมาณ 1.58 ชั่วโมง ในขณะที่กลุ่มรายงานการป่วยผู้ป่วยในจำนวน 201,733,782 records ใช้เวลาประมาณ 1.68 ชั่วโมง และกลุ่มรายงานจำนวนและอัตราผู้ป่วยในจากสาเหตุภายนอกจำนวน 201,733,782 records ใช้เวลาประมาณ 1.83 ชั่วโมง ซึ่งสรุปได้ว่าสามารถใช้วิธี Map Reduce กับชุดข้อมูลขนาดใหญ่ของกระทรวงสาธารณสุข และสามารถพัฒนาเป็นแนวทางการวิเคราะห์ข้อมูลรายงานการป่วยได้อย่างมีประสิทธิภาพ

คำสำคัญ: ข้อมูลขนาดใหญ่, แมพ-รีดิวส์, การวิเคราะห์ข้อมูล

วันที่รับต้นฉบับ 20 มีนาคม 2560; วันที่ตอบรับ 31 พฤษภาคม 2560

บทนำ

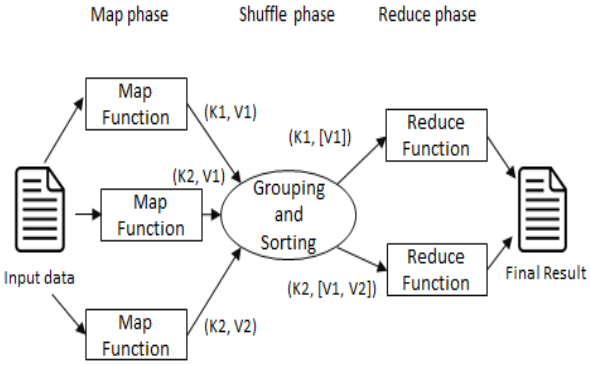
ในปี พ.ศ. 2558 กระทรวงสาธารณสุขได้มีการพัฒนาแนวทางในการจัดเก็บและรวบรวมข้อมูลด้านการแพทย์และสุขภาพให้เป็นไปในแนวทางเดียวกัน โดยให้โรงพยาบาลและสถานบริการปฐมภูมิทั่วประเทศดำเนินการจัดเก็บและรวบรวมข้อมูลตามโครงสร้างฐานข้อมูลด้านการแพทย์และสุขภาพในรูปแบบ 43 แฟ้มมาตรฐาน[1] ส่งผลกระทบให้ข้อมูลที่เก็บรวบรวมและส่งมายัง Health Data Center ซึ่งเป็นหน่วยงานกลางของกระทรวงสาธารณสุข มีขนาดใหญ่และเก็บอยู่ในรูปแบบของไฟล์ ทำให้การนำข้อมูลไปวิเคราะห์เพื่อจัดทำรายงานการป่วย[2] ด้วยวิธีการเดิมซึ่งใช้ฐานข้อมูลเชิงสัมพันธ์ (Relational Database) ต้องใช้เวลานานในการประมวลผล ก่อให้เกิดปัญหาความล่าช้าในการใช้รายงานการป่วยของกระทรวงสาธารณสุข ซึ่งจากการศึกษาของ Poonam S. Patil[3] พบว่าฐานข้อมูลเชิงสัมพันธ์ส่วนใหญ่ไม่เหมาะสมที่จะนำมาใช้กับข้อมูลที่มีขนาดใหญ่ นอกจากนี้บริษัทระดับโลกอย่าง Google ซึ่งเป็นผู้ให้บริการสืบค้นข้อมูลในอินเทอร์เน็ตก็เคยประสบปัญหาในการบริหารจัดการข้อมูลขนาดใหญ่ จนได้มีการศึกษาและเผยแพร่งานวิจัย Map Reduce[4] เพื่อใช้ในการประมวลผลข้อมูลบน Cluster Computer ของ Google ทำให้การสืบค้นข้อมูลของ Google ทำได้อย่างรวดเร็วและมีประสิทธิภาพมากขึ้น จากนั้น Doug Cutting ได้นำแนวความคิด Map Reduce ของ Google มาพัฒนาต่อจนเกิดเป็นเทคโนโลยี Hadoop ซึ่งเป็นหนึ่งในเทคโนโลยีการจัดการข้อมูลขนาดใหญ่ที่ได้รับความนิยมอย่างมาก ซึ่ง Hadoop Map Reduce[5] เป็นเทคโนโลยีที่ส่งคำสั่งไปยังคอมพิวเตอร์แต่ละเครื่อง เพื่อทำการประมวลผลข้อมูลโดยไม่ต้องมีการย้ายข้อมูลระหว่างประมวลผล ซึ่งได้มีการนำ Apache Hadoop ไปใช้แก้ปัญหากับการบริหารจัดเก็บและประมวลผลข้อมูลขนาดใหญ่และประยุกต์ใช้ในงานต่างๆ เช่น ด้านข้อมูลจราจรทางคอมพิวเตอร์ (Log)[6] โดยใช้ Hadoop Distributed File System (HDFS) ในการเก็บรายละเอียดการติดต่อสื่อสารผ่านระบบเครือข่ายในรูปแบบของไฟล์ และใช้ Hadoop Map Reduce ในการสืบค้นข้อมูล แต่ยังไม่มีการพัฒนาแนวคิดเทคโนโลยี Hadoop Map Reduce มาใช้ในการบริหารจัดการข้อมูลตามโครงสร้างฐานข้อมูลด้านการ

ผู้นิพนธ์ประสานงาน: นัฐพงษ์ หนูสิงห์, คณะวิศวกรรมศาสตร์, มหาวิทยาลัยธุรกิจบัณฑิตย์ 110/1-4 ถนนประชาชื่น แขวงทุ่งสองห้อง เขตหลักสี่ กรุงเทพฯ 10210 (โทร. 0-2954-7300; e-mail address: 585162020014@dpu.ac.th)

แพทย์และสุขภาพในรูปแบบ 43 แฟ้ม เพื่อวิเคราะห์ประมวลผลเป็นสรุปรายงานการป่วย ซึ่งข้อมูลที่มีขนาดใหญ่และถูกเก็บในรูปแบบของไฟล์เหล่านี้จำเป็นต้องมีการบริหารจัดการ โดยงานวิจัยชิ้นนี้ได้นำแนวคิดเทคโนโลยี Hadoop Map Reduce มาใช้พัฒนาแนวทางใหม่ในการวิเคราะห์ข้อมูลรายงานการป่วยของกระทรวงสาธารณสุข เพื่อแก้ปัญหาการบริการจัดเก็บและประมวลผลข้อมูลขนาดใหญ่ให้เป็นไปอย่างมีประสิทธิภาพ

งานวิจัยที่เกี่ยวข้อง

Hadoop เกิดขึ้นในปี พ.ศ. 2545 โดย Doug Cutting และ Mike Cafarella ได้เริ่มต้นโครงการที่ชื่อว่า Apache Nutch จุดประสงค์เพื่อต้องการจัดเก็บและทำ Index ของ Web Content สำหรับระบบสืบค้นข้อมูลผ่านเครือข่าย Internet อย่างไรก็ตามระบบจัดเก็บข้อมูล ณ ขณะนั้นส่วนใหญ่เป็นฐานข้อมูลเชิงสัมพันธ์ซึ่งที่ไม่เหมาะสมที่จะนำมาใช้ทำระบบสืบค้นข้อมูลผ่านเครือข่าย Internet ดังนั้นในช่วงแรก Apache Nutch จึงยังไม่รองรับข้อมูล Web Content ที่มีขนาดใหญ่ระดับพันล้านเพจได้[7] ต่อมาในปี พ.ศ. 2546 Google ได้ทำการเผยแพร่งานวิจัยที่ช่วยแก้ปัญหาการจัดเก็บข้อมูลขนาดใหญ่ที่ชื่อว่า Google Distributed Filesystem (GDFS)[8] และตามด้วยงานวิจัย Map Reduce ซึ่งมาใช้ในการประมวลผลข้อมูลบน Cluster Computer ทำให้การสืบค้นข้อมูลของ Google มีประสิทธิภาพและรวดเร็วมากกว่าเดิม หลังจากนั้นจึงเริ่มมีการพัฒนาโปรแกรมที่เรียกว่า Hadoop ซึ่งต่อยอดแนวคิดมาจาก GoogleDFS และ Map Reduce ของ Google โดย Hadoop เป็นโอเพนซอร์ส ประกอบด้วยส่วนการทำงานที่สำคัญ คือ Hadoop Distributed File System (HDFS)[9] ที่เป็นระบบแฟ้มข้อมูลแบบกระจายสำหรับจัดเก็บข้อมูลขนาดใหญ่ และรองรับรูปแบบการเขียนโปรแกรม Map Reduce ซึ่งประกอบด้วย 2 ฟังก์ชั่นการทำงาน ได้แก่ ฟังก์ชั่น Map และฟังก์ชั่น Reduce โดยที่ฟังก์ชั่น Map ทำหน้าที่รับข้อมูลแต่ละเรคคอร์ดจากชุดข้อมูลทั้งหมดมาประมวลผลเพื่อให้ได้รูปแบบข้อมูลเป็น key-value ซึ่งปกติ Hadoop จะสร้าง Map ฟังก์ชั่น ขึ้นมาเป็นจำนวนเท่ากับเรคคอร์ดที่ประมวลผล เช่น ถ้าใน 1 ไฟล์มีทั้งหมด 1,000 เรคคอร์ด Hadoop ก็จะสร้างฟังก์ชั่น Map ขึ้นมาทั้งหมด 1,000 ฟังก์ชั่นเช่นกัน โดยแต่ละฟังก์ชั่นจะทำงานอิสระต่อกัน ทำให้สามารถประมวลผลแบบขนานกันได้และคอยจัดการว่าถ้ามีฟังก์ชั่น Map ไหนเกิดข้อผิดพลาดทำงานไม่สำเร็จ ก็จะสั่งให้ฟังก์ชั่น Map นั้นทำงานใหม่อีกครั้ง ด้วยเหตุนี้ทำให้กรณีที่เครื่องคอมพิวเตอร์เครื่องใดเครื่องหนึ่งเกิดข้อผิดพลาด ก็จะไม่ส่งผลกระทบต่อผลลัพธ์ที่ได้จากการคำนวณ และจากนั้น Hadoop MapReduce จะรวบรวมผลลัพธ์ที่ได้จากฟังก์ชั่น Map มาจัดกลุ่มตาม key พร้อมกับเรียงลำดับของข้อมูล โดยเรียกส่วนการทำงานนี้ว่า Shuffle phase แล้วจึงส่งชุดข้อมูลแต่ละ key ไปให้ฟังก์ชั่น Reduce ประมวลผลต่อเพื่อให้ได้ผลลัพธ์ออกมา



ภาพที่ 1 แผนผังแสดงการประมวลผลข้อมูลด้วยวิธี Map Reduce

นอกจากนี้ Hadoop ยังถูกนำไปประยุกต์ใช้ในงานต่างๆ เช่น การประยุกต์ใช้ Hadoop Map Reduce ในด้านข้อมูลจราจรทางคอมพิวเตอร์ (Log) ภายใต้ระบบ HDFS ซึ่งทำหน้าที่เก็บรายละเอียดการติดต่อสื่อสารผ่านระบบเครือข่ายในรูปแบบของไฟล์ นอกจากนี้ยังนำ Hadoop Map Reduce ไปใช้สำหรับการสืบค้นข้อมูลของผู้กระทำความผิดตามพระราชบัญญัติว่าด้วยการกระทำความผิดเกี่ยวกับคอมพิวเตอร์ พ.ศ. 2550 ได้อย่างรวดเร็วมากยิ่งขึ้น ทั้งนี้ข้อมูลที่นำมาประมวลผลส่วนใหญ่จะอยู่ในรูปแบบที่มีความสัมพันธ์กัน ซึ่งจำเป็นต้องมีการรวบรวมข้อมูลทั้งหมดก่อนนำไปใช้วิเคราะห์ประมวลผล ดังเช่น การศึกษาของ Spyros Blanas, Jignesh M. Patel ที่ทำการศึกษาเปรียบเทียบวิธีการประมวลผลของชุดข้อมูลที่มีความสัมพันธ์กัน โดยวิธีการใช้ Map Reduce และวิธีการ Join Algorithm[10] นำมาทดสอบกับข้อมูล 2 ชุด ซึ่งประกอบด้วย 1) ข้อมูลจราจรเหตุการณ์ต่างๆ (Log) ที่เกิดจากการใช้งานเวปไซต์ และ 2) ข้อมูลผู้ใช้งาน (User Information) พบว่าวิธีการใช้ Join Algorithm มีความเหมาะสมกับข้อมูลในรูปแบบต่างๆ ได้แก่ Repartition Join, Broadcast Join, Semi Join, และ Per-split Semi-Join

ดังนั้นจากการศึกษาที่ผ่านมา จึงทำให้เกิดแนวคิดของการพัฒนาวิธีการวิเคราะห์ประมวลผลรายงานการป่วยแบบใหม่ โดยการใช้ HDFS ของ Hadoop ในการบริหารจัดเก็บข้อมูลแบบกระจายควบคู่กับการใช้ Hadoop Map Reduce ในการประมวลผลข้อมูล เพื่อให้ได้ผลลัพธ์ตามรูปแบบรายงานของกระทรวงสาธารณสุข ทำให้การวิเคราะห์ประมวลผลรายงานการป่วยสามารถทำได้อย่างมีประสิทธิภาพมากยิ่งขึ้น

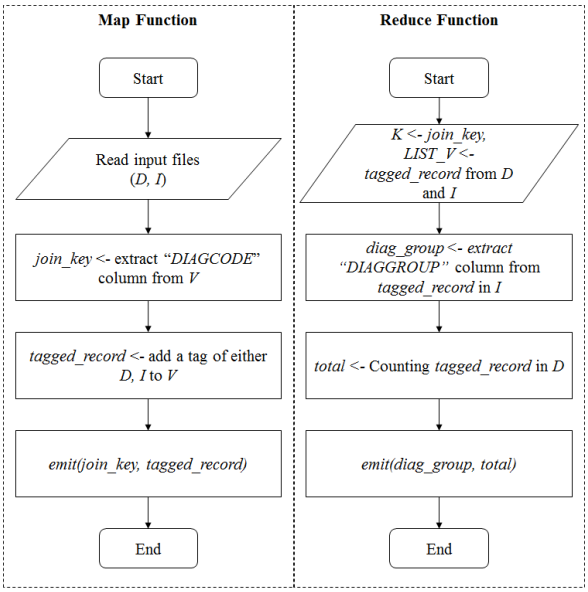
การพัฒนาแนวทางการวิเคราะห์ข้อมูลรายงานการป่วย

แนวทางการวิเคราะห์ข้อมูลรายงานการป่วยได้นำเทคโนโลยี Apache Hadoop มาใช้โดยได้นำข้อมูลจำนวน 3 แฟ้มจากข้อมูลสุขภาพตามมาตรฐานโครงสร้างทั้งหมดจำนวน 43 แฟ้ม มาจัดเก็บรวบรวมเข้าสู่ระบบ Hadoop Distributed File System ซึ่งแบ่งข้อมูลแต่ละประเภทแฟ้มออกเป็น Directory เช่น ใน Directory “diagnosis_opd” ก็จะเก็บเฉพาะแฟ้มข้อมูลวินิจฉัยโรคของผู้ป่วยนอกและผู้มารับบริการ (DIAGNOSIS_OPD) ของทุกจังหวัด เป็นต้น และใช้ Hadoop Map Reduce ในการประมวลผลข้อมูล เพื่อให้ได้สรุปรายงานการป่วยตามรูปแบบของกระทรวงสาธารณสุข ซึ่งมีการ นำเสนอข้อมูลดังกล่าวแบ่งเป็น 3 ส่วน ได้แก่ ส่วนที่ 1 รายงานการป่วยผู้ป่วยนอก ส่วนที่ 2 รายงานการป่วยของผู้ป่วยใน และส่วนที่ 3 รายงานจำนวนและอัตราผู้ป่วยในจากสาเหตุภายนอก

รายงานการป่วยผู้ป่วยนอก

การนำเสนอข้อมูลรายงานการป่วยของผู้ป่วยนอก เป็นการแยกจำนวนผู้ป่วยตามกลุ่มโรคจำนวน 21 กลุ่ม โดยข้อมูลที่นำมาประมวลผล ประกอบด้วยข้อมูลจำนวน 2 ชุด ได้แก่ 1) แฟ้มข้อมูลวินิจฉัยโรคของผู้ป่วยนอกและผู้มารับบริการ (DIAGNOSIS_OPD) ตามโครงสร้างมาตรฐาน 43 แฟ้ม และ 2) ข้อมูล 21 กลุ่มโรคของผู้ป่วยนอกสาเหตุตาม รง.504 (ICD10TM) ซึ่งแบ่งขั้นตอนการทำงานของ Map Reduce จำนวน 2 งาน (Job) ตามลำดับ

- งานแรก (Job1) ใช้ Repartition Join Algorithm ในการรวมแฟ้มข้อมูลวินิจฉัยโรคของผู้ป่วยนอกและผู้มารับบริการ (DIAGNOSIS_OPD) และชุดข้อมูลกลุ่มโรคของผู้ป่วยนอกสาเหตุตาม รง.504 (ICD10TM) โดยใช้รหัสสถานบริการ (HCODE) และทะเบียนบุคคล (PID) เป็นข้อมูลสำคัญในการเชื่อมความสัมพันธ์ระหว่างข้อมูลทั้ง 2 ชุด และเมื่อกำหนดให้ D แทน ชุดข้อมูล DIAGNOSIS_OPD กำหนดให้ I แทนชุดข้อมูล ICD10TM และกำหนดให้ V แทนข้อมูลจำนวน 1 Record ในชุดข้อมูล D หรือ I สามารถอธิบายการทำงานของ Map Reduce งานแรก (Job1) ได้ตาม**ภาพที่ 2**
- งานที่สอง (Job2) เป็นการนับข้อมูลที่ได้จากงานแรก (Job1) ว่ามีจำนวนผู้ป่วยแต่ละกลุ่มโรคจำนวนเท่าไร และทำการเรียงลำดับจำนวนผู้ป่วยแบ่งตามกลุ่มโรคจากจำนวนมากไปหาน้อย

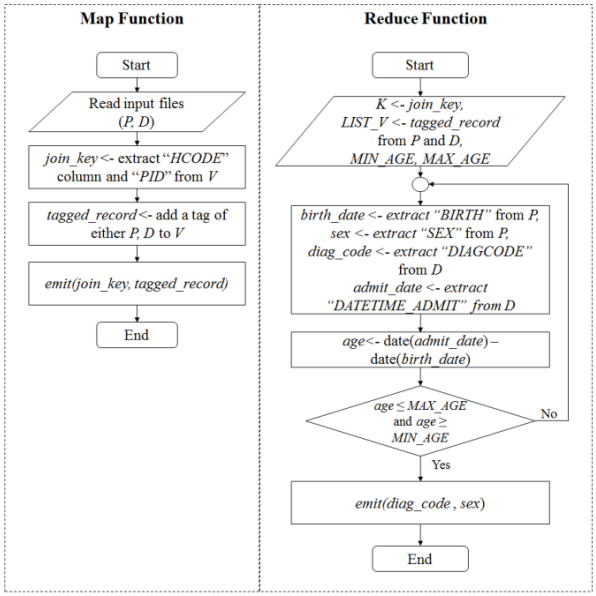


ภาพที่ 2 แผนผังงาน Map Reduce งานแรก (Job1) การประมวลผลรายงานการป่วยผู้ป่วยนอก

รายงานการป่วยของผู้ป่วยใน

นำเสนอข้อมูลผู้ป่วยที่เข้ามารับการรักษาในโรงพยาบาลและมีสิทธิการรักษาในระบบประกันสุขภาพของประเทศไทย โดยจำแนกตามสาเหตุการป่วยแต่ละช่วงอายุ และจัดลำดับตามจำนวนการเข้ารับการรักษา ข้อมูลที่นำมาประมวลผลประกอบด้วยข้อมูลจำนวน 3 ชุดจากจำนวนทั้งหมด 43 ชุด โดยข้อมูล 2 ชุดแรก เป็นข้อมูลที่ได้จากโครงสร้างมาตรฐาน 43 แฟ้ม ได้แก่ 1) แฟ้มข้อมูลทั่วไปของประชาชนในเขตรับผิดชอบและผู้ที่มาใช้บริการ (PERSON) และ 2) แฟ้มข้อมูลวินิจฉัยโรคของผู้ป่วยใน (DIAGNOSIS_IPD) ส่วนข้อมูลชุดที่สอง เป็นข้อมูลกลุ่มโรคจำนวน 298 กลุ่มโรคของผู้ป่วยใน (ICD10TM) ซึ่งการประมวลผลข้อมูลจะใช้ Map Reduce จำนวน 3 งาน โดยทำงานตามลำดับ ดังนี้

- งานแรก (Job1) เป็นการรวบรวมข้อมูลจำนวน 2 ชุดเข้าด้วยกันตามวิธีการ Repartition Join โดยใช้ข้อมูลรหัสสถานบริการ (HCODE) และทะเบียนบุคคล (PID) ในการเชื่อมความสัมพันธ์ระหว่างข้อมูลทั้ง 2 ชุด และกรองข้อมูล (Filter) ที่ไม่ตรงช่วงอายุผู้ป่วยออกไป และเมื่อกำหนดให้ D แทน ชุดข้อมูล DIAGNOSIS_IPD กำหนดให้ P แทนชุดข้อมูล PERSON และกำหนดให้ V แทนข้อมูลจำนวน 1 Record ในชุดข้อมูล D หรือ P ซึ่งสามารถอธิบายการทำงานของ Map Reduce งานแรก (Job1) ได้ดังภาพ



ภาพที่ 3 แผนผังงาน Map Reduce งานแรก (Job1) การประมวลผลรายงานการป่วยผู้ป่วยใน

- งานที่สอง (Job2) ใช้ Repartition Join Algorithm รวมผลลัพธ์จากงานแรกและข้อมูลสาเหตุการป่วยแยกตามกลุ่มโรคของผู้ป่วยในจำนวน 298 กลุ่มโรค (ICD10TM) โดยใช้ข้อมูลรหัสโรคที่วินิจฉัย (DIAGCODE) เป็นข้อมูลสำคัญในการเชื่อมความสัมพันธ์ระหว่างข้อมูลทั้ง 2 ชุดเข้าด้วยกัน
- งานที่สาม (Job3) เป็นการจัดกลุ่มผลลัพธ์จากงานที่สอง (Job2) ตามสาเหตุของโรคแยกเป็นกลุ่มโรคจำนวน 298 กลุ่มโรค และเรียงลำดับจากจำนวนของผู้ป่วยที่เข้ารับการรักษาจากมากไปหาน้อย

รายงานจำนวนและอัตราผู้ป่วยในจากสาเหตุภายนอก

รายละเอียดการป่วยของผู้ป่วยในจำแนกตามกลุ่มสาเหตุภายนอกหลักๆทั้งหมดจำนวน 35 สาเหตุ โดยจำแนกตามเพศ และมีการจัดลำดับตามสาเหตุภายนอกจำนวน 10 ลำดับ ข้อมูลที่นำมาประมวลผลประกอบไปด้วยข้อมูลจำนวน 3 ชุด ได้แก่ 1) แฟ้มข้อมูลทั่วไปของประชาชนในเขตรับผิดชอบและผู้ที่มาใช้บริการ (PERSON) 2) แฟ้มข้อมูลวินิจฉัยโรคของผู้ป่วยใน (DIAGNOSIS_IPD) และ 3) ข้อมูลบัญชีกลุ่มโรคย่อยซึ่งเกิดจากสาเหตุภายนอกตามบทที่ 20 (รหัส V01-Y98) จากบัญชีจำแนกทางสถิติระหว่างประเทศของโรคและปัญหาสุขภาพที่เกี่ยวข้อง (International Statistical Classification of Diseases and Related Health Problems; ICD10TM) โดยการประมวลผลรายงานใช้ Hadoop Map Reduce จำนวน 3 งาน (Job) ตามลำดับ ดังนี้

- งานแรก (Job1) ใช้ Repartition Join Algorithm ในการรวมแฟ้มข้อมูลทั่วไปของประชาชนในเขตรับผิดชอบและผู้ที่มาใช้บริการ (PERSON) และข้อมูลวินิจฉัยผู้ป่วยใน (DIAGNOSIS_IPD) โดยใช้รหัสสถานบริการ (HCODE) และทะเบียนบุคคล (PID) เป็นข้อมูลสำคัญในการเชื่อมความสัมพันธ์ระหว่างข้อมูลทั้งจำนวน 2 ชุด
- งานที่สอง (Job2) ใช้ Repartition Join Algorithm รวมข้อมูลผลลัพธ์จากงานแรก (Job1) กับข้อมูลกลุ่มโรคย่อยซึ่งเกิดจากสาเหตุภายนอกจำนวน 35 กลุ่มสาเหตุ (รหัส V01-Y98) โดยใช้รหัสโรคที่วินิจฉัย (DIAGCODE) เป็นข้อมูลสำคัญในการเชื่อมความสัมพันธ์
- งานที่สาม (Job3) เป็นการนับข้อมูลผลลัพธ์ที่ได้จากงานที่สอง (Job2) ตามกลุ่มโรคซึ่งเกิดจากสาเหตุภายนอก และเรียงลำดับจากจำนวนผู้ป่วยที่เข้ารับการรักษาจากมากไปหาน้อย

ผลการวิจัย

การศึกษาครั้งนี้ได้ทำการติดตั้ง Apache Hadoop บน Computer Server แบ่งออกเป็น ทำหน้าที่กระจายงาน (Master) จำนวน 1 เครื่อง และทำหน้าที่เป็นเครื่องประมวลผลทำงาน (Slave) จำนวน 3 เครื่อง โดยทั้งหมดทำงานบนระบบปฏิบัติการ Linux

การศึกษาใช้ Map Reduce ประมวลผลรายงานผู้ป่วยนอก การศึกษาครั้งนี้เป็นการนำข้อมูลจริงที่ได้จากแฟ้มข้อมูลการวินิจฉัยโรคของผู้ป่วยนอกและผู้มารับบริการ (DIAGNOSIS_OPD) ตามโครงสร้างฐานข้อมูลด้านการแพทย์และสุขภาพในรูปแบบ 43 แฟ้มมาตรฐาน ซึ่งรวบรวมข้อมูลจากทั่วประเทศมาประมวลผล โดยการใช้ Map Reduce เพื่อจัดทำเป็นรายงานการป่วยของผู้ป่วยนอก ซึ่งประกอบด้วย 2 รายงาน ได้แก่ 1) รายงาน 10 ลำดับแรกของอัตราผู้ป่วยนอกจำแนกตามกลุ่มสาเหตุการป่วย (Report2) และ 2) รายงานจำนวนผู้ป่วยนอกทั้งหมดทั่วประเทศ (Report3) เพื่อนำ Hadoop Map Reduce มาพัฒนาเป็นแนวทางใหม่ในการวิเคราะห์ข้อมูลรายงานการป่วยดังกล่าว

จากตารางที่ 1 พบว่าการใช้ Hadoop Map Reduce ในการประมวลผลข้อมูลด้านการแพทย์และสุขภาพในรูปแบบ 43 แฟ้มมาตรฐาน จากแฟ้มข้อมูลวินิจฉัยโรคของผู้ป่วยนอกและผู้มารับบริการ (DIAGNOSIS_OPD) จำนวนทั้งหมด 243,610,347 records โดยใช้เวลาในการประมวลผลรายงาน 10 ลำดับแรกอัตราผู้ป่วยนอกตามกลุ่มสาเหตุการป่วย (Report2) รวมทั้งหมด (Total time spent) 5,895,056 มิลลิวินาทีหรือ

ประมาณ 1.64 ชั่วโมง ขณะที่รายงานจำนวนผู้ป่วยนอกทั้งหมดทั่วประเทศ (Report3) ใช้เวลาใกล้เคียงกันอยู่ที่ 5,491,574 มิลลิวินาที หรือประมาณ 1.53 ชั่วโมง

ดังนั้นจากการศึกษาครั้งนี้สามารถสรุปได้ว่า การนำ Map Reduce มาใช้ในการประมวลผลข้อมูลด้านการแพทย์และสุขภาพในรูปแบบ 43 แฟ้มมาตรฐานที่มีข้อมูลขนาดใหญ่สามารถนำมาจัดทำเป็นรายงานการป่วยของผู้ป่วยนอกได้อย่างมีประสิทธิภาพ

การศึกษาใช้ Map Reduce ประมวลผลรายงานผู้ป่วยใน

การศึกษานี้เป็นการนำข้อมูลจริงจากแฟ้มข้อมูลวินิจฉัยโรคของผู้ป่วยใน (DIAGNOSIS_IPD) และแฟ้มข้อมูลทั่วไปของประชาชนในเขตรับผิดชอบและผู้ที่มาใช้บริการ (PERSON) ตามโครงสร้างฐานข้อมูลด้านการแพทย์และสุขภาพในรูปแบบ 43 แฟ้มมาตรฐาน ซึ่งรวบรวมจากทั่วประเทศมาประมวลผลเพื่อจัดทำเป็นรายงานการป่วยของผู้ป่วยใน ประกอบด้วย 5 รายงาน ได้แก่ 1) รายงานอัตราการป่วยของผู้ป่วยในกลุ่มอายุต่ำกว่า 1 ปี รวมทุกการวินิจฉัยโรคจำแนกตามเพศ (Report7) 2) รายงานอัตราการป่วยของผู้ป่วยในกลุ่มอายุแรกเกิด - 4 ปี รวมทุกการวินิจฉัยโรคจำแนกตามเพศ (Report8) 3) รายงานอัตราการป่วยของผู้ป่วยในกลุ่มอายุ 5 - 14 ปี รวมทุกการวินิจฉัยโรคจำแนกตามเพศ (Report9) 4) รายงานอัตราการป่วยของผู้ป่วยในกลุ่มอายุ 15 - 24 ปี รวมทุกการวินิจฉัยโรคจำแนกตามเพศ (Report10) และ 5) รายงานอัตราการป่วยของผู้ป่วยในกลุ่มอายุ 15-59 ปี รวมทุกการวินิจฉัยโรคจำแนกตามเพศ ทั้งนี้เพื่อนำ Hadoop Map Reduce มาพัฒนาเป็นแนวทางใหม่ในการวิเคราะห์ข้อมูลรายงานการป่วยดังกล่าว

จากตารางที่ 2 พบว่าการใช้ Hadoop Map Reduce ประมวลผลข้อมูล จากแฟ้มข้อมูลวินิจฉัยโรคของผู้ป่วยใน (DIAGNOSIS_IPD) และแฟ้มข้อมูลทั่วไปของประชาชนในเขตรับผิดชอบและผู้ที่มาใช้บริการ (PERSON) ตามโครงสร้างฐานข้อมูลด้านการแพทย์และสุขภาพในรูปแบบ 43 แฟ้มมาตรฐาน จำนวนทั้งสิ้น 201,733,782 records ใช้เวลาประมวลผลรายงานอัตราการป่วยของผู้ป่วยในกลุ่มอายุต่ำกว่า 1 ปี รวมทุกการวินิจฉัยโรคจำแนกตามเพศ (Report7) ทั้งหมด 6,486,066 มิลลิวินาที หรือประมาณ 1.8 ชั่วโมง ใช้เวลาในการประมวลผลรายงานอัตราการป่วยของผู้ป่วยในกลุ่มอายุแรกเกิด - 4 ปี รวมทุกการวินิจฉัยโรคจำแนกตามเพศ (Report8) รวมทั้งหมด 6,069,371 มิลลิวินาที หรือประมาณ 1.68 ชั่วโมง ใช้เวลาประมวลผลรายงานอัตราการป่วยของผู้ป่วยในกลุ่มอายุ 5 - 14 ปี รวมทุกการวินิจฉัยโรคจำแนกตามเพศ (Report9) รวมทั้งหมด 5,767,446 มิลลิวินาที หรือประมาณ 1.6 ชั่วโมง ในส่วนของรายงานอัตราการป่วยของผู้ป่วยในกลุ่มอายุ 15 - 24 ปี รวมทุกการวินิจฉัยโรคจำแนกตามเพศ (Report10) ใช้เวลาในการประมวลผลรวมทั้งหมด 5,848,874 มิลลิวินาที หรือประมาณ 1.62 ชั่วโมง และรายงานอัตราการป่วยของผู้ป่วยในกลุ่มอายุ 15-59 ปี รวมทุกการวินิจฉัยโรคจำแนกตามเพศ (Report11) ใช้เวลาประมวลผลรวมทั้งหมด 6,043,604 หรือประมาณ 1.68 ชั่วโมง

จากผลการทดลองพบว่าสามารถนำ Map Reduce มาประมวลผลแฟ้มข้อมูลวินิจฉัยโรคของผู้ป่วยใน (DIAGNOSIS_IPD) และแฟ้มข้อมูลทั่วไปของประชาชนในเขตรับผิดชอบและผู้ที่มาใช้

ตารางที่ 1 ผลลัพธ์การ ประมวลผลรายงานการป่วยผู้ป่วยนอก โดยวิธี Map Reduce

Report	Job1				Job2				Total time spent (ms)
	Map Input Records	*Reduce Output Records*	*CPU time spent (ms)*	*Time spent (ms)*	*Map Input Records*	*Reduce Output Records*	*CPU time spent (ms)*	*Time spent (ms)*	
Report 2	243,610,347	4,866	1,789,550	5,895,056	4,866	10	1,480	4,074	5,899,130
Report 3	243,610,347	4,866	1,808,320	5,487,222	4,866	21	1,780	4,352	5,491,574

ตารางที่ 2 ผลลัพธ์การประมวลผลรายงานการป่วยผู้ป่วยใน โดยวิธี Map Reduce

Report	Job1				Job2				Job3				Total time spent (ms)
	Map Input Records	*Reduce Output Records*	*CPU time spent (ms)*	*Time spent (ms)*	*Map Input Records*	*Reduce Output Records*	*CPU time spent (ms)*	*Time spent (ms)*	*Map Input Records*	*Reduce Output Records*	*CPU time spent (ms)*	*Time spent (ms)*	
Report 7	201,733,782	1,544,563	2,365,410	6,457,393	1,546,603	2,266	162	24,532	2,266	299	1,330	4,141	6,486,066
Report 8	201,733,782	1,427,454	2,408,110	6,053,588	1,429,494	2,235	7,720	11,864	2,235	299	1,330	3,919	6,069,371
Report 9	201,733,782	503,400	2,390,260	5,750,847	505,440	2,235	4,740	9,314	2,253	299	1,360	7,285	5,767,446
Report 10	201,733,782	1,050,169	2,377,650	5,834,410	1,052,209	2,343	6,650	10,324	2,343	299	1,350	4,140	5,848,874
Report 11	201,733,782	5,873,346	2,362,900	5,996,126	5,875,386	2,719	22,780	43,346	2,719	299	1,410	4,132	6,043,604

บริการ (PERSON) ตามโครงสร้างฐานข้อมูลด้านการแพทย์และสุขภาพในรูปแบบ 43 แฟ้มมาตรฐานที่ถูกรวบรวมมาจากทั่วประเทศซึ่งเป็นข้อมูลขนาดใหญ่จำนวน 201,733,782 records สำหรับจัดทำรายงานการป่วยของผู้ป่วยใน ทั้ง 5 รายงานดังกล่าวได้

การศึกษาใช้ Map Reduce ประมวลผลรายงานจำนวนและอัตราผู้ป่วยในจากสาเหตุภายนอก

การศึกษานี้เป็นการนำข้อมูลจริงจากแฟ้มข้อมูลวินิจฉัยโรคของผู้ป่วยใน (DIAGNOSIS_IPD) และแฟ้มข้อมูลทั่วไปของประชาชนในเขตรับผิดชอบและผู้ที่มาใช้บริการ (PERSON) ตามโครงสร้างฐานข้อมูลด้านการแพทย์และสุขภาพในรูปแบบ 43 แฟ้มมาตรฐาน ซึ่งรวบรวมจากทั่วประเทศมาประมวลผลจำนวน 201,733,782 records เพื่อจัดทำเป็นรายงานจำนวนและอัตราผู้ป่วยในจากสาเหตุภายนอก ซึ่งประกอบด้วย 2 รายงาน ได้แก่ 1) อัตราการป่วยของผู้ป่วยในจากสาเหตุภายนอก กลุ่มอายุ 15 - 59 ปี 2) อัตราการป่วยของผู้ป่วยในจากสาเหตุภายนอก กลุ่มอายุ 60 ปีขึ้นไป ทั้งนี้เพื่อนำ Hadoop Map Reduce มาพัฒนาเป็นแนวทางใหม่ในการวิเคราะห์ข้อมูลรายงานการป่วยดังกล่าว

จากตารางที่ 3 พบว่าการใช้ Hadoop Map Reduce ประมวลผลรายงานจำนวนและอัตราผู้ป่วยในจากสาเหตุภายนอก ผลที่ได้คือ รายงานอัตราการป่วยของผู้ป่วยในจากสาเหตุภายนอก (Report14) ใช้เวลาประมวลผลรวมทั้งสิ้น 6,610,529 มิลลิวินาที หรือประมาณ 1.84 ชั่วโมง และรายงานอัตราการป่วยของผู้ป่วยในจากสาเหตุภายนอก กลุ่มอายุ 60 ปีขึ้นไป ใช้เวลาประมวลผลรวมทั้งสิ้น 6,616,688 มิลลิวินาที หรือประมาณ 1.84 ชั่วโมง

ดังนั้นจากการศึกษาครั้งนี้พบว่าสามารถนำ Map Reduce มาใช้ประมวลผลแฟ้มข้อมูลวินิจฉัยโรคของผู้ป่วยใน (DIAGNOSIS_IPD) และแฟ้มข้อมูลทั่วไปของประชาชนในเขตรับผิดชอบและผู้ที่มาใช้บริการ (PERSON) ตามโครงสร้างฐานข้อมูลด้านการแพทย์และสุขภาพในรูปแบบ 43 แฟ้มมาตรฐานที่ถูกรวบรวมมาจากทั่วประเทศซึ่งเป็นข้อมูลขนาดใหญ่จำนวน 201,733,782 records เพื่อจัดทำรายงานจำนวนและอัตราการป่วยของผู้ป่วยในจากสาเหตุภายนอก ทั้ง 2 รายงานดังกล่าวได้

ตารางที่ 3 ผลลัพธ์การประมวลผลรายงานจำนวนและอัตราผู้ป่วยในจากสาเหตุภายนอก โดยวิธี Map Reduce

Report	Job1				Job2				Job3				Total time spent (ms)
	Map Input Records	*Reduce Output Records*	*CPU time spent (ms)*	*Time spent (ms)*	*Map Input Records*	*Reduce Output Records*	*CPU time spent (ms)*	*Time spent (ms)*	*Map Input Records*	*Reduce Output Records*	*CPU time spent (ms)*	*Time spent (ms)*	
Report 14	201,733,782	487,741	2,763,850	6,597,226	489,781	1,273	5,280	9,239	1,273	10	1,080	4,064	6,610,529
Report 15	201,733,782	487,741	2,763,850	6,597,226	489,781	1,273	5,210	15,595	1,273	10	1,110	3,867	6,616,688

สรุป

การศึกษาครั้งนี้ได้นำเสนอแนวทางใหม่ในการวิเคราะห์ข้อมูลรายงานการป่วย โดยวิธี Map Reduce โดยนำข้อมูลจำนวน 3 แฟ้ม จากข้อมูลแฟ้มตามโครงสร้างฐานข้อมูลด้านการแพทย์และสุขภาพในรูปแบบ 43 แฟ้มมาตรฐานมาจัดทำเป็นรายงานการป่วยทั้งหมด 3 กลุ่มรายงาน ได้แก่ 1 รายงานการป่วยผู้ป่วยนอก 2 รายงานการป่วยผู้ป่วยใน และ 3 รายงานจำนวนและอัตราการป่วยของผู้ป่วยในจากสาเหตุภายนอก ผลชี้ให้เห็นว่าสามารถนำ Hadoop Map Reduce ไปปรับใช้เป็นแนวทางในการวิเคราะห์ข้อมูลเพื่อจัดทำเป็นรายงานการป่วยของกระทรวงสาธารณสุขได้

เอกสารอ้างอิง

1. สำนักนโยบายและยุทธศาสตร์ สำนักงานปลัดกระทรวงสาธารณสุข กระทรวงสาธารณสุข. คู่มือการปฏิบัติงานการจัดเก็บและจัดส่งข้อมูลตามโครงสร้างมาตรฐานข้อมูลด้านสุขภาพ กระทรวงสาธารณสุข Version 2.1. มกราคม 2559; หน้า ก - ข.
2. สำนักนโยบายและยุทธศาสตร์ สำนักงานปลัดกระทรวงสาธารณสุข กระทรวงสาธารณสุข, สรุปรายงานการป่วย พ.ศ. 2557; 2558, หน้า 1.
3. Patil P, Phursule R. Survey paper on big data processing and Hadoop components. International Journal of Science and Research, 2014; 3 (10): 585-590.
4. Dean J, Ghemawat S. MapReduce: simplied data processing on large clusters. Commun ACM, 2008; 51, 107-113.
5. Gunarathne T. Hadoop MapReduce v2, 2nd ed, 2015, 65 - 67.
6. Rattanapoka C. The design and implementation of computer traffic log searcher system using Hadoop Map/Reduce framework. The Journal of Industrial Technology, 2012; 8 (3): 61-71.
7. White T. Hadoop: The Definitive Guide. 4th ed.; 2015, 12 - 14.
8. Ghemawat S, Gobioff H, Leung ST. The Google file system. Proceedings of the nineteenth ACM symposium on Operating systems principles (SOSP '03). ACM, New York, NY, USA, 2003, 29-43.
9. Borthakur D. The Hadoop Distributed File System: Architecture and Design, The Apache Software Foundation, 2005, 55-60.
10. Blanas S, Patel JM, Ercegovac V, Rao J, Shekita EJ, Tian Y. A comparison of join algorithms for log processing in MapReduce. Proceedings of the 2010 ACM SIGMOD International Conference on Management of data; 2010, 89-93.